aws

Pola

AWS Prescriptive Guidance

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Prescriptive Guidance: Pola

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

AWS Pola Bimbingan Preskriptif 1
Fondasi awan 3
Mengotomatiskan pembuatan akun menggunakan Landing Zone Accelerator 4
Ringkasan 4
Prasyarat dan batasan 4
Arsitektur 5
Alat 6
Praktik terbaik 9
Epik 9
Sumber daya terkait 34
Informasi tambahan 35
AWS Sumber daya inventaris secara otomatis 37
Ringkasan 37
Prasyarat dan batasan 38
Arsitektur 39
Alat 39
Praktik terbaik 41
Epik 41
Pemecahan masalah 48
Sumber daya terkait 48
Bangun AWS landing zone dengan MongoDB Atlas 50
Ringkasan 50
Prasyarat dan batasan 51
Arsitektur 51
Alat 53
Epik 53
Sumber daya terkait 63
Konfigurasikan Log Aliran VPC di seluruh akun 65
Ringkasan 65
Prasyarat dan batasan 65
Arsitektur 66
Alat 66
Praktik terbaik 67
Epik 69

Sumber daya terkait 70
Informasi tambahan 70
Kelola set AWS izin menggunakan Terraform 73
Ringkasan 73
Prasyarat dan batasan 74
Arsitektur 74
Alat 76
Praktik terbaik 77
Epik 78
Pemecahan masalah 89
Sumber daya terkait 90
Informasi tambahan 90
Tandai lampiran Transit Gateway secara otomatis 92
Ringkasan 92
Prasyarat dan batasan 92
Arsitektur 92
Alat 94
Epik 95
Sumber daya terkait 102
Lebih banyak pola 103
AI & pembelajaran mesin 104
Kaitkan AWS CodeCommit repositori dengan Amazon SageMaker AI Studio Classic di seluruh akun 106
Ringkasan 106
Prasyarat dan batasan 106
Arsitektur 106
Alat 107
Epik 108
Informasi tambahan 113
Secara otomatis mengekstrak konten dari file PDF 116
Ringkasan 116
Prasyarat dan batasan 116
Arsitektur 117
Alat 118
Epik 119
Sumber daya terkait 123

Lampiran 124
Bangun model peramalan awal dingin menggunakan SageMaker AI DeepAR 125
Ringkasan 125
Prasyarat dan batasan 125
Arsitektur 126
Alat 127
Praktik terbaik 127
Epik 128
Sumber daya terkait 130
Membangun MLOps alur kerja menggunakan SageMaker AI dan Azure DevOps 131
Ringkasan 131
Prasyarat dan batasan 131
Arsitektur 132
Alat 134
Praktik terbaik 135
Epik 136
Pemecahan masalah 145
Sumber daya terkait 146
Konfigurasikan logging di Amazon Bedrock menggunakan CloudFormation 148
Ringkasan 148
Prasyarat dan batasan 148
Arsitektur 149
Alat 150
Epik 151
Sumber daya terkait 153
Buat wadah Docker SageMaker untuk pelatihan model di Step Functions 155
Ringkasan 155
Prasyarat dan batasan 155
Arsitektur 156
Alat 156
Epik 157
Sumber daya terkait 169
Buat kontrol akses di Amazon EKS dengan agen Amazon Bedrock 170
Ringkasan 170
Prasyarat dan batasan 170
Arsitektur 171

Alat 172
Praktik terbaik 173
Epik 173
Pemecahan masalah 192
Sumber daya terkait 193
Terapkan kasus penggunaan RAG pada AWS 194
Ringkasan 194
Prasyarat dan batasan 194
Arsitektur 195
Alat 197
Praktik terbaik 198
Epik 199
Sumber daya terkait 202
Informasi tambahan 202
Menerapkan beberapa objek model pipeline dalam satu titik akhir SageMaker 204
Ringkasan 204
Prasyarat dan batasan 204
Arsitektur 205
Alat 205
Epik 206
Sumber daya terkait 216
Terapkan validasi keamanan pengkodean waktu nyata dengan Kiro 217
Ringkasan 217
Prasyarat dan batasan 218
Arsitektur 219
Alat 220
Praktik terbaik 222
Epik 223
Pemecahan masalah 231
Sumber daya terkait 232
Informasi tambahan 233
Kembangkan asisten obrolan AI dengan menggunakan RAG dan prompt ReAct 236
Ringkasan 236
Prasyarat dan batasan 237
Arsitektur 238
Alat 239

Praktik terbaik 241
Epik 241
Pemecahan masalah 249
Sumber daya terkait 249
Informasi tambahan 250
Kembangkan asisten berbasis obrolan menggunakan Amazon Bedrock 251
Ringkasan 251
Prasyarat dan batasan 252
Arsitektur 252
Alat 253
Praktik terbaik 255
Epik 256
Sumber daya terkait 260
Informasi tambahan 260
Dokumentasikan pengetahuan kelembagaan dari input suara 263
Ringkasan 263
Prasyarat dan batasan 263
Arsitektur 264
Alat 265
Praktik terbaik 267
Epik 267
Sumber daya terkait 273
Hasilkan rekomendasi yang dipersonalisasi menggunakan Amazon Personalize 275
Ringkasan 275
Prasyarat dan batasan 275
Arsitektur 275
Alat 277
Epik 278
Sumber daya terkait 280
Informasi tambahan 281
Merampingkan alur kerja ML menggunakan SageMaker AI dan Hydra 285
Ringkasan 285
Prasyarat dan batasan 285
Arsitektur 286
Alat 287
Praktik terbaik 288

Epik 288
Pemecahan masalah 295
Sumber daya terkait 295
Informasi tambahan 295
..... 297
Ringkasan 297
Prasyarat dan batasan 298
Arsitektur 301
Alat 303
Praktik terbaik 304
Epik 304
Sumber daya terkait 311
Informasi tambahan 312
Gunakan Amazon Q Developer sebagai asisten pengkodean 317
Ringkasan 317
Prasyarat dan batasan 317
Alat 318
Praktik terbaik 318
Epik 319
Pemecahan masalah 328
Sumber daya terkait 328
Gunakan SageMaker Processing untuk rekayasa fitur terdistribusi dari kumpulan data ML skala terabyte 329
Ringkasan 329
Prasyarat dan batasan 329
Arsitektur 330
Alat 333
Epik 333
Sumber daya terkait 346
Lampiran 347
Visualisasikan hasil AI/ML model menggunakan Flask dan Elastic Beanstalk 348
Ringkasan 348
Prasyarat dan batasan 348
Arsitektur 349
Alat 351
Epik 351

Sumber daya terkait 361
Informasi tambahan 361
Lebih banyak pola 366
Analitik 367
Menganalisis data Amazon Redshift di Layanan Analisis Microsoft SQL Server 369
Ringkasan 369
Prasyarat dan batasan 369
Arsitektur 369
Alat 370
Epik 370
Sumber daya terkait 372
..... 373
Ringkasan 373
Prasyarat dan batasan 373
Arsitektur 374
Alat 374
Epik 375
Sumber daya terkait 380
Otomatiskan konsumsi data dari AWS Data Exchange Amazon S3 381
Ringkasan 381
Prasyarat dan batasan 381
Arsitektur 381
Alat 382
Epik 383
Sumber daya terkait 385
Lampiran 385
Otomatiskan penegakan enkripsi di AWS Glue 386
Ringkasan 386
Prasyarat dan batasan 386
Arsitektur 386
Alat 387
Praktik terbaik 388
Epik 389
Sumber daya terkait 391
Membuat pipeline data untuk memproses data Google Analytics menggunakan AWS DataOps Development Kit 392

Ringkasan 392
Prasyarat dan batasan 392
Arsitektur 392
Alat 393
Epik 394
Pemecahan masalah 396
Sumber daya terkait 397
Informasi tambahan 397
Membangun pipa pemrosesan video 400
Ringkasan 400
Prasyarat dan batasan 400
Arsitektur 401
Alat 401
Epik 402
Sumber daya terkait 409
Informasi tambahan 410
Lampiran 410
Buat pipeline ETL dari Amazon S3 ke Amazon Redshift 411
Ringkasan 411
Prasyarat dan batasan 411
Arsitektur 412
Alat 412
Epik 413
Sumber daya terkait 420
Informasi tambahan 421
Membangun mesh data perusahaan dengan Amazon DataZone 422
Ringkasan 422
Prasyarat dan batasan 422
Arsitektur 423
Alat 424
Epik 425
Sumber daya terkait 437
Informasi tambahan 437
Hitung value at risk (VaR) dengan menggunakan layanan AWS 440
Ringkasan 440
Prasyarat dan batasan 441

Arsitektur 441
Alat 442
Praktik terbaik 443
Epik 444
Sumber daya terkait 447
Konfigurasikan akses lintas akun ke Katalog Data AWS Glue bersama 448
Ringkasan 448
Prasyarat dan batasan 448
Arsitektur 448
Alat 450
Epik 450
Sumber daya terkait 463
Informasi tambahan 463
Konversi NORMALIZE ke Amazon Redshift SQL 464
Ringkasan 464
Prasyarat dan batasan 464
Arsitektur 464
Alat 465
Epik 470
Sumber daya terkait 470
Konversi RESET WHEN ke Amazon Redshift SQL 472
Ringkasan 472
Prasyarat dan batasan 472
Arsitektur 472
Alat 473
Epik 477
Sumber daya terkait 477
Menerapkan dan mengelola data lake tanpa server di AWS 479
Ringkasan 479
Prasyarat dan batasan 479
Arsitektur 480
Alat 481
Epik 482
Sumber daya terkait 484
..... 486
Ringkasan 486

Prasyarat dan batasan 486
Arsitektur 487
Alat 487
Epik 488
Sumber daya terkait 491
Lampiran 491
Pastikan Amazon EMR masuk ke Amazon S3 492
Ringkasan 492
Prasyarat dan batasan 493
Arsitektur 493
Alat 494
Epik 494
Sumber daya terkait 497
Lampiran 497
Hasilkan data uji menggunakan AWS Glue 498
Ringkasan 498
Prasyarat dan batasan 498
Arsitektur 499
Alat 499
Praktik terbaik 500
Epik 500
Sumber daya terkait 511
Informasi tambahan 511
Menyerap data IoT langsung ke Amazon S3 516
Ringkasan 516
Prasyarat dan batasan 516
Arsitektur 517
Alat 518
Praktik terbaik 518
Epik 519
Pemecahan masalah 526
Sumber daya terkait 527
Informasi tambahan 528
Luncurkan pekerjaan Spark di Amazon EMR menggunakan fungsi Lambda 532
Ringkasan 532
Prasyarat dan batasan 532

Arsitektur 533
Alat 533
Epik 534
Sumber daya terkait 537
Informasi tambahan 537
Lampiran 540
Migrasikan beban kerja Apache Cassandra ke Amazon Keyspaces 541
Ringkasan 541
Prasyarat dan batasan 541
Arsitektur 541
Alat 542
Praktik terbaik 543
Epik 543
Pemecahan masalah 556
Sumber daya terkait 556
Informasi tambahan 556
Migrasi data Hadoop ke Amazon S3 dengan menggunakan WANdisco 558
Ringkasan 558
Prasyarat dan batasan 558
Arsitektur 559
Epik 560
Sumber daya terkait 565
Informasi tambahan 565
Migrasi Oracle Business Intelligence 12C ke AWS Cloud 567
Ringkasan 567
Prasyarat dan batasan 567
Arsitektur 568
Alat 569
Epik 570
Sumber daya terkait 584
Informasi tambahan 584
Migrasikan Tumpukan ELK ke AWS Cloud 590
Ringkasan 590
Prasyarat dan batasan 590
Arsitektur 592
Alat 594

Epik 595
Sumber daya terkait 602
Informasi tambahan 604
Migrasi data untuk AWS menggunakan Starburst 605
Ringkasan 605
Prasyarat dan batasan 605
Arsitektur 605
Alat 607
Epik 608
Sumber daya terkait 611
Optimalkan konsumsi ETL ukuran file input 612
Ringkasan 612
Prasyarat dan batasan 612
Arsitektur 612
Alat 613
Epik 613
Sumber daya terkait 616
Informasi tambahan 617
Mengatur pipeline ETL dengan Step Functions 618
Ringkasan 618
Prasyarat dan batasan 618
Arsitektur 619
Alat 620
Epik 621
Pemecahan masalah 628
Sumber daya terkait 629
Informasi tambahan 629
Lakukan analitik ML menggunakan Amazon Redshift ML 630
Ringkasan 630
Prasyarat dan batasan 630
Arsitektur 631
Alat 632
Epik 633
Sumber daya terkait 636
Kueri Amazon DynamoDB dengan menggunakan Amazon Athena 638
Ringkasan 638

Prasyarat dan batasan 638
Arsitektur 639
Alat 639
Epik 640
Pemecahan masalah 642
Sumber daya terkait 643
Kueri tabel DynamoDB menggunakan Athena 644
Ringkasan 644
Prasyarat dan batasan 644
Arsitektur 645
Alat 645
Epik 646
Sumber daya terkait 654
Informasi tambahan 654
Siapkan ruang data minimum yang layak 656
Ringkasan 656
Prasyarat dan batasan 657
Arsitektur 658
Alat 659
Praktik terbaik 660
Epik 660
Pemecahan masalah 716
Sumber daya terkait 716
Informasi tambahan 716
Siapkan penyortiran khusus bahasa untuk hasil kueri Amazon Redshift 721
Ringkasan 721
Prasyarat dan batasan 721
Alat 721
Epik 722
Sumber daya terkait 727
Informasi tambahan 727
Berlangganan fungsi Lambda ke notifikasi acara dari bucket S3 lintas wilayah 731
Ringkasan 731
Prasyarat dan batasan 731
Arsitektur 731
Alat 732

Epik 733
Sumber daya terkait 736
Tiga jenis pekerjaan AWS Glue untuk mengonversi data 738
Ringkasan 738
Prasyarat dan batasan 738
Arsitektur 738
Alat 739
Epik 740
Sumber daya terkait 742
Informasi tambahan 743
Lampiran 749
Visualisasikan log audit Amazon Redshift menggunakan Athena dan QuickSight 750
Ringkasan 750
Prasyarat dan batasan 750
Arsitektur 750
Alat 751
Epik 751
Sumber daya terkait 756
Lampiran 756
Visualisasikan laporan kredensi IAM menggunakan Amazon Quick Sight 757
Ringkasan 757
Prasyarat dan batasan 757
Arsitektur 758
Alat 759
Epik 759
Informasi tambahan 765
Lebih banyak pola 767
Hitung 769
Kontainer & layanan mikro 770
Akses Amazon Neptunus dari Amazon EKS 772
Akses aplikasi kontainer di Amazon ECS 784
Akses aplikasi kontainer di Amazon ECS dengan tipe peluncuran AWS Fargate 801
Akses aplikasi kontainer secara pribadi di Amazon EKS 815
Mengotomatiskan backup untuk Amazon RDS untuk instans DB Amazon RDS for PostgreSQL 823
Mengotomatiskan penerapan Node Termination Handler 836

Secara otomatis membangun dan menyebarkan aplikasi Java ke Amazon EKS 851
Salin gambar kontainer Amazon ECR di seluruh akun dan Wilayah 875
Membuat definisi tugas Amazon ECS pada EC2 instans menggunakan Amazon EFS 900
Menyebarkan fungsi Lambda dengan gambar kontainer 906
Menerapkan layanan mikro Java di Amazon ECS menggunakan AWS Fargate 915
Menerapkan paket Kubernetes menggunakan Amazon EKS dan Helm 921
Menerapkan cluster CockroachDB di Amazon EKS menggunakan Terraform 932
Menerapkan layanan mikro Java di Amazon EKS dan mengeksposnya dengan Application Load Balancer 943
Menerapkan aplikasi yang dikelompokkan ke Amazon ECS dengan menggunakan AWS Copilot 957
Menerapkan aplikasi berbasis GRPC di Amazon EKS 968
Menyebarkan aplikasi kontainer pada AWS IoT Greengrass V2 980
Menyebarkan wadah dengan menggunakan Elastic Beanstalk 1004
Hasilkan alamat IP keluar statis menggunakan Lambda dan Amazon VPC 1010
Identifikasi gambar kontainer duplikat secara otomatis 1024
Instal Agen SSM di node pekerja Amazon EKS 1053
Instal Agen dan CloudWatch agen SSM di node pekerja Amazon EKS menggunakan preBootstrapCommands 1059
Migrasikan Pengontrol Ingress NGINX untuk Mode Otomatis EKS 1066
Migrasikan beban kerja kontainer dari ARO ke ROSA 1087
Jalankan tugas Amazon ECS di Amazon WorkSpaces 1103
Jalankan wadah API Docker web ASP.NET di AWS 1113
Jalankan beban kerja stateful dengan penyimpanan data persisten 1125
Mengatur penskalaan otomatis menggunakan Amazon EKS Pod Identity dan KEDA 1150
..... 1173
Gunakan TLS timbal balik di Amazon ECS dengan ALB 1196
Lebih banyak pola 1226
Nirserver 1228
Buat aplikasi React Native menggunakan AWS Amplify 1229
Kelola penyewa secara terpusat di beberapa produk SaaS 1248
Konsolidasikan pembuatan URL dan unduhan objek yang telah ditetapkan sebelumnya Amazon S3 1260
Buat koneksi Amazon EventBridge lintas akun 1276
Mengirimkan catatan DynamoDB ke Amazon S3 1290
Menerapkan versi API berbasis jalur dengan menggunakan API Gateway 1297

Impor pustaka psycopg2 ke AWS Lambda 1309
Integrasikan API Gateway dengan Amazon SQS 1318
Proses secara APIs asinkron dengan AWS Lambda 1332
Proses secara APIs asinkron dengan Amazon DynamoDB Streams 1341
Proses secara APIs asinkron dengan Amazon SQS 1351
Jalankan tugas Systems Manager Automation secara sinkron dari Step Functions 1360
Jalankan pembacaan paralel objek S3 dengan AWS Lambda 1374
Kirim data telemetri dari Lambda ke OpenSearch 1386
Siapkan router sel tanpa server 1401
Siapkan akses pribadi ke bucket Amazon S3 1420
Memecahkan masalah status Step Functions 1427
Lebih banyak pola 1437
Jaringan 1440
Mengotomatiskan peering untuk AWS Transit Gateway 1441
Memusatkan konektivitas jaringan menggunakan AWS Transit Gateway 1448
Konfigurasikan enkripsi HTTPS untuk Oracle JD Edwards EnterpriseOne dengan menggunakan Application Load Balancer 1455
Connect ke data Application Migration Service dan kontrol pesawat melalui jaringan pribadi 1466
Buat objek Infoblox menggunakan sumber daya khusus AWS CloudFormation 1480
Buat arsitektur IPAM Multi-wilayah menggunakan Terraform 1494
Kustomisasi CloudWatch peringatan untuk Network Firewall 1510
Menerapkan sumber daya di Wavelength Zone menggunakan Terraform 1529
Migrasi data DNS secara massal ke zona host pribadi Route 53 1537
Ubah header HTTP saat Anda bermigrasi dari F5 ke Application Load Balancer di AWS ... 1547
Laporkan temuan Network Access Analyzer dalam beberapa Akun AWS 1553
Mengatur resolusi DNS untuk jaringan hybrid di lingkungan multi-akun AWS 1583
..... 1598
Lihat log dan metrik AWS Network Firewall menggunakan Splunk 1604
Lebih banyak pola 1614
Pengiriman konten 1615
Kirim AWS WAF log ke Splunk menggunakan Firehose 1616
Sajikan konten statis dalam bucket S3 melalui VPC dengan menggunakan CloudFront 1624
Lebih banyak pola 1633
Database & penyimpanan 1634
Basis Data 1635

Mengakses data SQL Server lokal menggunakan server tertaut 1638
Tambahkan HA ke Oracle PeopleSoft di AWS 1644
Menganalisis objek untuk migrasi sebagian dari Oracle ke PostgreSQL 1668
Menilai kinerja kueri untuk memigrasi SQL Server ke MongoDB 1684
Mengotomatiskan blue/green penyebaran database global Aurora dengan IAc 1694
Otomatiskan replikasi lintas akun instans Amazon RDS 1729
Mengotomatiskan tugas database dengan AWS Lambda dan Penjadwal Tugas 1743
Otomatiskan failover dan failback dengan DR Orchestrator Framework 1754
Secara otomatis mencadangkan database SAP HANA 1781
Secara otomatis menghasilkan model PynamoDB dan fungsi CRUD untuk DynamoDB 1789
Blokir akses publik ke Amazon RDS 1797
Konfigurasikan otentikasi Windows untuk Amazon RDS menggunakan Microsoft AD 1804
Konfigurasikan akses Amazon DynamoDB lintas akun 1820
Konfigurasikan perutean hanya-baca dalam grup ketersediaan Selalu Aktif 1833
Connect dengan menggunakan terowongan SSH di pgAdmin 1842
Mengkonversi kueri JSON Oracle ke database PostgreSQL SQL 1847
Salin tabel Amazon DynamoDB di seluruh akun 1878
Salin tabel Amazon DynamoDB di seluruh akun 1884
Buat laporan biaya dan penggunaan untuk Amazon RDS dan Amazon Aurora 1897
Menyebarkan SQL Server di FCIs Amazon dan Amazon EC2 FSx 1904
Meniru array PL/SQL asosiatif Oracle di Amazon Aurora dan Amazon RDS 1920
Aktifkan koneksi terenkripsi untuk instance PostgreSQL DB 1931
Enkripsi Amazon RDS yang ada untuk instans DB Amazon RDS for PostgreSQL 1940
Menerapkan penandaan otomatis database Amazon RDS saat diluncurkan 1948
Perkirakan biaya DynamoDB 1954
Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB 1966
Perkirakan ukuran mesin Amazon RDS untuk database Oracle menggunakan laporan AWR 1970
Ekspor Amazon RDS untuk tabel SQL Server ke bucket S3 2001
Menangani blok anonim dalam pernyataan SQL Dinamis 2012
Membantu menegakkan penandaan DynamoDB 2019
Melaksanakan DR Lintas wilayah 2025
Migrasikan 100+ argumen fungsi Oracle ke PostgreSQL 2042
Migrasikan beban kerja Redis ke Redis Enterprise Cloud di AWS 2048
Migrasi SAP HANA ke AWS menggunakan HSR 2075
Migrasi grup ketersediaan SQL Server Always On menggunakan MGN AWS 2088

Migrasikan SQL Server ke AWS menggunakan grup ketersediaan terdistribusi 2099
Migrasi database relasional ke MongoDB Atlas di AWS 2111
Migrasikan lingkungan MongoDB yang dihosting sendiri ke MongoDB Atlas 2121
Memigrasikan database Oracle ke Amazon DynamoDB 2133
Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Oracle dan AWS DMS SharePlex 2138
Bermigrasi dari MySQL lokal ke Amazon EC2 2147
Pantau Amazon Aurora untuk enkripsi 2156
Pantau GoldenGate log dengan menggunakan Amazon CloudWatch 2161
Replatform Oracle Database EE ke Amazon RDS for Oracle SE2 2174
Replikasi database mainframe ke AWS dengan menggunakan Exactly Connect 2186
Jadwalkan pekerjaan untuk Amazon RDS dan Aurora PostgreSQL 2205
Kirim pemberitahuan untuk RDS untuk SQL Server menggunakan server SMTP 2211
Siapkan DR untuk SAP di IBM Db2 di AWS 2226
Siapkan CI/CD pipeline untuk migrasi database dengan menggunakan Terraform 2247
Siapkan Microsoft SQL Server FCI di Amazon menggunakan EC2 FSx 2256
Menyiapkan HA/DR arsitektur untuk Oracle E-Business Suite di Amazon RDS Custom 2272
Streaming data dari database ke MongoDB Atlas di AWS 2281
Peran transisi untuk aplikasi Oracle PeopleSoft 2293
Bongkar data dari Amazon Redshift di seluruh akun 2329
Pola migrasi database berdasarkan beban kerja 2358
Lebih banyak pola 2369
Penyimpanan & cadangan 2374
Izinkan EC2 instance menulis akses ke bucket S3 di AMS 2375
Mengotomatiskan konsumsi aliran data ke dalam database Snowflake 2381
Secara otomatis mengenkripsi volume EBS 2392
Cadangkan server Sun SPARC di emulator Charon-SSP AWS 2403
Cadangkan dan arsipkan data ke Amazon S3 dengan Veeam 2426
Aktifkan pengarsipan DB2 log ke Amazon S3 2452
Migrasikan data Hadoop ke Amazon S3 menggunakan DistCp dan AWS untuk Amazon S3 PrivateLink 2470
Lebih banyak pola 2486
Alat developer 2488
DevOps 2489
Percepat MLOps dengan Backstage dan template Amazon SageMaker AI swalayan 2493
Mengotomatiskan AWS operasi infrastruktur 2513

Mengotomatiskan CloudFront pembaruan untuk perubahan titik akhir penyeimbang beban 2522
Mengotomatiskan CodeGuru ulasan untuk aplikasi AWS CDK Python 2530
Mengotomatiskan Rantai Pasokan AWS penyebaran data lake 2542
Mengotomatiskan penilaian sumber daya AWS 2578
Mengotomatiskan instalasi sistem SAP 2592
Mengotomatiskan portofolio Service Catalog dan penyebaran produk menggunakan AWS CDK 2605
Mengotomatiskan pipeline hotfix dinamis 2624
Mengotomatiskan penghapusan tumpukan CloudFormation 2652
Mengotomatiskan konsumsi metrik kustom Amazon MWAA 2664
Otomatiskan penerapan set tumpukan dengan menggunakan AWS dan AWS CodePipeline CodeBuild 2682
Secara otomatis melampirkan kebijakan terkelola untuk Systems Manager ke profil EC2 instance 2712
Secara otomatis membangun CI/CD jaringan pipa dan kluster Amazon ECS untuk layanan mikro 2728
Buat dan dorong gambar Docker ke Amazon ECR 2740
Membangun dan menguji aplikasi iOS dengan layanan AWS 2746
Konfigurasikan TLS timbal balik untuk aplikasi di Amazon EKS 2752
Buat sumber daya WorkSpaces Aplikasi Amazon menggunakan CloudFormation 2763
Buat parser log khusus untuk Amazon ECS menggunakan Firelens 2770
Buat kerangka orkestrasi sumber daya 2779
Buat permintaan tarik otomatis untuk AWS infrastruktur 2793
Buat pipeline CI dinamis untuk proyek Java dan Python 2805
Menyebarkan kenari CloudWatch Synthetics 2822
Terapkan ChatOps solusi untuk mengelola hasil pemindaian SAST 2829
Menerapkan sistem agen di Amazon Bedrock dengan CreWai 2847
..... 2864
Menerapkan kode di beberapa Wilayah AWS 2868
Terapkan beban kerja dari DevOps pipeline Azure ke kluster Amazon EKS pribadi 2881
Jalankan kueri Amazon Redshift SQL 2915
Ekspor laporan AWS Backup sebagai file CSV 2928
Ekspor tag EC2 instans Amazon ke file CSV 2938
Buat CloudFormation template AWS yang berisi aturan terkelola AWS Config 2945
Berikan akses lintas akun instance SageMaker notebook ke repositori CodeCommit 2952
Menerapkan strategi percabangan GitHub Flow 2964

Menerapkan strategi percabangan Gitflow 2973
Menerapkan strategi percabangan Trunk 2985
Menerapkan pemindaian Checkov kustom terpusat 2991
Menerapkan diagnostik dengan K8sgpt dan Amazon Bedrock 3002
Memulai CI/CD jaringan pipa yang berbeda setelah mendeteksi perubahan dalam monorepo 3016
Integrasikan repositori Bitbucket dengan AWS Amplify 3033
Luncurkan CodeBuild proyek di seluruh akun AWS menggunakan Lambda 3042
Kelola failover Multi-AZ untuk cluster EMR dengan menggunakan Application Recovery Controller 3055
Mengelola blue/green penyebaran layanan mikro ke beberapa akun dan Wilayah 3070
Pantau repositori Amazon ECR untuk izin wildcard 3103
Optimalkan penerapan tanpa server multi-akun 3108
AWS Service Catalog Produk penyediaan menggunakan GitHub Tindakan 3129
Menyediakan peran IAM dengan hak istimewa paling sedikit 3140
Publikasikan CloudWatch metrik Amazon ke file CSV 3154
Hapus EC2 entri Amazon Akun AWS 3160
Hapus EC2 entri Amazon dengan cara yang sama Akun AWS 3177
Jalankan pengujian unit untuk pekerjaan ETL Python di AWS Glue 3187
Siapkan CI/CD pipa dengan CodePipeline 3198
Siapkan logging terpusat pada skala menggunakan Terraform 3214
Siapkan end-to-end enkripsi untuk aplikasi di Amazon EKS 3239
Sederhanakan penerapan aplikasi multi-tenant Amazon EKS 3252
Merampingkan alur kerja Amazon Lex 3271
Berlangganan beberapa titik akhir email ke topik SNS 3289
Gunakan konstruksi AWS Fargate WaitCondition kait 3294
Gunakan repo Git pihak ketiga di AWS CodePipeline 3307
Validasi konfigurasi Terraform dengan menggunakan AWS CodePipeline 3317
Lebih banyak pola 3333
Infrastruktur 3337
Akses host bastion menggunakan Session Manager dan Instance Connect 3339
Memusatkan resolusi DNS dengan menggunakan AWS Managed Microsoft AD 3355
Memusatkan pemantauan dengan menggunakan Observability Access Manager 3367
Periksa EC2 instance untuk tag wajib saat peluncuran 3380
Bersihkan sumber daya AFT dengan aman setelah kehilangan file status 3386
Buat pipeline di Wilayah AWS yang tidak mendukung AWS CodePipeline 3399

Sesuaikan nama peran default dengan menggunakan AWS CDK aspek dan lubang keluar ... 3407
Menerapkan cluster Cassandra di Amazon EC2 dengan statis pribadi IPs ... 3415
Memperluas VRFs ke AWS menggunakan Transit Gateway Connect ... 3422
Dapatkan notifikasi Amazon SNS untuk perubahan status pada kunci AWS KMS ... 3438
Pertahankan ruang IP yang dapat dirutekan dalam desain VPC multi-akun untuk subnet non-beban kerja ... 3446
Menyediakan produk Terraform di Service Catalog dari repo ... 3452
Daftarkan beberapa Akun AWS dengan satu alamat email ... 3470
Menyiapkan resolusi DNS untuk jaringan hybrid di lingkungan AWS akun tunggal ... 3487
Siapkan bot UiPath RPA secara otomatis di Amazon EC2 ... 3492
Siapkan PeopleSoft arsitektur yang sangat tersedia di AWS ... 3508
Menyiapkan pemulihan bencana untuk Oracle JD Edwards EnterpriseOne ... 3536
Mengatur deteksi dan pelaporan drift ... 3563
Berhasil mengimpor bucket S3 sebagai tumpukan CloudFormation ... 3569
Sinkronkan sistem file Amazon EFS di berbagai Wilayah ... 3580
Uji AWS infrastruktur menggunakan LocalStack dan Tes Terraform ... 3588
Tingkatkan cluster SAP Pacemaker dari ke ENSA1 ENSA2 ... 3596
Gunakan Availability Zone yang konsisten VPCs di berbagai akun ... 3618
Gunakan pengguna IDs dalam kebijakan IAM ... 3624
Validasi Account Factory untuk kode Terraform secara lokal ... 3635
Lebih banyak pola ... 3655
Aplikasi web & seluler ... 3659
Mengautentikasi pengguna aplikasi web React ... 3660
Buat aplikasi React menggunakan AWS Amplify dan Amazon Cognito ... 3670
Buat portal untuk micro-frontend ... 3686
Menerapkan SPA berbasis React ke Amazon S3 dan CloudFront ... 3717
Menerapkan API Amazon API Gateway menggunakan titik akhir pribadi ... 3724
Sematkan komponen visual Amazon Quick Sight ke dalam aplikasi web ... 3731
Jelajahi pengembangan aplikasi web dengan Green Boost ... 3753
Jalankan pengujian unit dengan menggunakan AWS CodeBuild ... 3781
Struktur proyek Python dalam arsitektur heksagonal ... 3789
Lebih banyak pola ... 3815
IoT ... 3817
Ekstrak dan kueri atribut AWS IoT SiteWise metadata ... 3818
Ringkasan ... 3818

Prasyarat dan batasan 3818
Arsitektur 3819
Alat 3819
Epik 3820
Sumber daya terkait 3823
Informasi tambahan 3823
..... 3826
Ringkasan 3826
Prasyarat dan batasan 3827
Arsitektur 3827
Alat 3828
Praktik terbaik 3828
Epik 3829
Pemecahan masalah 3844
Sumber daya terkait 3846
Informasi tambahan 3846
Lebih banyak pola 3848
Migrasi & modernisasi 3849
Migrasi 3850
Buat CloudFormation templat AWS untuk AWS DMS 3851
Memulai dengan penemuan portofolio otomatis 3855
Memigrasikan beban kerja Cloudera lokal ke AWS 3863
Mengatasi kesalahan koneksi setelah memigrasi SQL Server ke AWS 3878
Mulai ulang AWS Replication Agent secara otomatis tanpa menonaktifkan SELinux 3881
Arsitek ulang 3888
Rehost 4257
Pindah 4492
Platform Ulang 4535
Pola migrasi berdasarkan beban kerja 4961
Lebih banyak pola 4970
Modernisasi 4972
Menganalisis dan memvisualisasikan arsitektur perangkat lunak dalam CAST Imaging 4973
Menilai kesiapan aplikasi sebelum bermigrasi ke AWS dengan menggunakan CAST Highlight 4982
Secara otomatis mengarsipkan data DynamoDB yang kedaluwarsa ke Amazon S3 5006
Membangun arsitektur tanpa server multi-tenant di Amazon Service OpenSearch 5023

Menyebarkan aplikasi multi-stack 5072
Menerapkan aplikasi bersarang menggunakan AWS SAM 5081
Menerapkan isolasi penyewa SaaS untuk Amazon S3 dengan menggunakan TVM AWS Lambda 5090
Menerapkan pola saga tanpa server menggunakan Step Functions 5114
Mengelola aplikasi kontainer lokal dengan Amazon ECS Anywhere 5129
Memodernisasi aplikasi Formulir Web ASP.NET di AWS 5139
Jalankan beban kerja berbasis peristiwa dengan AWS Fargate 5155
Orientasi penyewa dalam arsitektur SaaS 5165
Gunakan CQRS dan sumber acara 5190
Lebih banyak pola 5215
Mainframe 5218
Akses Layanan AWS dari IBM z/OS dengan menginstal AWS CLI 5220
Cadangkan dan arsipkan data mainframe ke Amazon S3 5239
Membangun program COBOL Db2 dengan menggunakan dan AWS Mainframe Modernization AWS CodeBuild 5267
Membangun PAC Server Perusahaan Fokus Mikro 5291
Membangun penampil file mainframe di AWS Cloud 5311
Kontainerisasi aplikasi Blu Age yang dimodernisasi 5327
Konversi data EBCDIC ke ASCII di AWS 5337
Konversi file EBCDIC mainframe ke file ASCII menggunakan AWS Lambda 5356
Konversi file data mainframe dengan tata letak rekaman yang kompleks 5379
Menerapkan lingkungan untuk aplikasi kontainer 5397
Hasilkan wawasan data Db2 z/OS 5407
Hasilkan wawasan menggunakan AWS Mainframe Modernization dan Amazon Q 5458
Menerapkan otentikasi Microsoft Entra ID untuk AWS Blu Age 5477
Integrasikan Stonebranch Universal Controller dengan AWS 5496
Memigrasi dan mereplikasi file VSAM ke AWS Cloud menggunakan Tepat 5529
Modernisasi aplikasi mainframe menggunakan AWS Transform 5547
Modernisasi aplikasi mainframe menggunakan AWS Transform 5566
Memodernisasi manajemen output mainframe pada AWS 5601
Modernisasi beban kerja pencetakan batch mainframe Anda pada AWS 5647
Modernisasi lingkungan mainframe Anda dengan Rocket Software Enterprise Suite 5676
Modernisasi beban kerja pencetakan online mainframe Anda di AWS 5693
Memindahkan file mainframe ke Amazon S3 menggunakan Transfer Family 5728
Optimalkan kinerja aplikasi modern AWS Blu Age Anda 5741

Mengamankan akses pengguna dalam database federasi Db2 5765
Transfer z/OS data Db2 ke AWS 5775
Lebih banyak pola 5803
Manajemen 5804
Manajemen biaya 5805
Membuat laporan biaya dan penggunaan terperinci untuk pekerjaan AWS Glue 5806
Buat laporan biaya dan penggunaan terperinci untuk klaster EMR Amazon 5811
Lebih banyak pola 5815
Komputasi pengguna akhir 5816
Menerapkan otentikasi SAMP 2.0 untuk Amazon WorkSpaces 5817
Lebih banyak pola 5827
Komputasi performa tinggi 5828
Menerapkan sistem file Lustre dengan menggunakan Terraform dan DRA 5829
Siapkan dasbor pemantauan Grafana untuk AWS ParallelCluster 5837
Lebih banyak pola 5850
Awan hibrida 5851
Siapkan pipeline untuk beban kerja hybrid di Amazon ECS Anywhere 5852
Lebih banyak pola 5872
Manajemen & tata kelola 5873
Peringatan saat sumber daya Amazon Data Firehose tidak dienkripsi 5875
Otomatiskan alokasi Amazon VPC IPv4 IPAM CIDR 5880
Otomatis menambahkan atau memperbarui entri registri Windows 5911
Secara otomatis membuat RFC menggunakan Python 5916
Secara otomatis berhenti dan memulai instans Amazon RDS DB 5923
Memusatkan distribusi paket perangkat lunak di AWS Organizations dengan menggunakan Terraform 5938
Konfigurasikan logging untuk aplikasi.NET di CloudWatch Log 5951
Salin produk AWS Service Catalog di seluruh akun dan Wilayah AWS 5960
Buat matriks RACI untuk operasi cloud 5970
Buat alarm untuk metrik kustom menggunakan CloudWatch 5976
Buat AWS Cloud9 IDE dengan volume EBS terenkripsi default 5983
Buat CloudWatch dasbor berbasis tag secara otomatis 5989
Dokumentasikan desain landing zone Anda 5998
Aktifkan Amazon DevOps Guru di seluruh organisasi dengan AWS CDK 6002
Mengatur set izin untuk beberapa akun menggunakan AFT 6030
Menerapkan AFT dengan menggunakan pipeline bootstrap 6042

Kelola produk AWS Service Catalog di beberapa akun dan Wilayah AWS 6066
Pantau kluster Alat Pacu Jantung SAP RHEL 6074
Pantau aktivitas aplikasi menggunakan Wawasan CloudWatch Log 6093
Pantau penggunaan AMI di seluruh Akun AWS 6104
Lihat detail snapshot EBS untuk akun atau organisasi AWS 6120
Lebih banyak pola 6127
Pesan & komunikasi 6130
Otomatiskan konfigurasi RabbitMQ di Amazon MQ 6131
Tingkatkan kualitas panggilan di workstation agen di Amazon Connect 6139
Lebih banyak pola 6155
Strategi multi-akun 6156
Migrasi Akun AWS dari ke AWS Organizations AWS Control Tower 6157
Menyiapkan peringatan untuk penutupan akun terprogram di AWS Organizations 6167
Lebih banyak pola 6177
Keamanan, identitas & kepatuhan 6178
Otomatiskan respons insiden dan forensik 6180
Ringkasan 6180
Prasyarat dan batasan 6180
Arsitektur 6181
Alat 6183
Epik 6184
Sumber daya terkait 6188
Informasi tambahan 6188
Lampiran 6189
Secara otomatis mengaudit akses dari alamat IP publik 6190
Ringkasan 6190
Prasyarat dan batasan 6190
Arsitektur 6191
Alat 6193
Praktik terbaik 6194
Epik 6194
Pemecahan masalah 6198
Sumber daya terkait 6199
Secara otomatis memulihkan instans dan klaster Amazon RDS DB yang tidak terenkripsi 6200
Ringkasan 6200
Prasyarat dan batasan 6200

Arsitektur 6202
Alat 6202
Praktik terbaik 6203
Epik 6204
Sumber daya terkait 6210
Informasi tambahan 6210
Secara otomatis memvalidasi dan menerapkan kebijakan dan peran IAM 6212
Ringkasan 6212
Prasyarat dan batasan 6213
Arsitektur 6214
Alat 6214
Epik 6215
Sumber daya terkait 6219
Mengintegrasikan Security Hub CSPM dan Jira secara dua arah 6220
Ringkasan 6220
Prasyarat dan batasan 6221
Arsitektur 6221
Alat 6223
Epik 6224
Sumber daya terkait 6233
Informasi tambahan 6234
Bangun pipa untuk gambar kontainer yang diperkeras 6236
Ringkasan 6236
Prasyarat dan batasan 6236
Arsitektur 6237
Alat 6240
Epik 6240
Pemecahan masalah 6248
Sumber daya terkait 6249
Memusatkan manajemen kunci akses IAM di AWS Organizations dengan menggunakan Terraform 6250
Ringkasan 6250
Prasyarat dan batasan 6251
Arsitektur 6251
Alat 6253
Praktik terbaik 6253

Epik 6254
Pemecahan masalah 6263
Sumber daya terkait 6264
Periksa CloudFront distribusi Amazon untuk log akses, HTTPS, dan versi TLS 6265
Ringkasan 6265
Prasyarat dan batasan 6265
Arsitektur 6266
Alat 6266
Epik 6267
Sumber daya terkait 6270
Lampiran 6270
Pilih alur autentikasi Amazon Cognito 6271
Ringkasan 6271
Prasyarat dan batasan 6271
Arsitektur 6272
Alat 6276
Epik 6276
Sumber daya terkait 6280
Informasi tambahan 6281
Buat aturan AWS Config khusus menggunakan Guard 6282
Ringkasan 6282
Prasyarat dan batasan 6283
Arsitektur 6283
Alat 6288
Epik 6288
Pemecahan masalah 6290
Sumber daya terkait 6291
Buat laporan temuan Prowler dari beberapa Akun AWS 6293
Ringkasan 6293
Prasyarat dan batasan 6294
Arsitektur 6295
Alat 6295
Epik 6297
Pemecahan masalah 6319
Sumber daya terkait 6320
Informasi tambahan 6320

Menerapkan AWS Control Tower kontrol menggunakan AWS CDK 6322
Ringkasan 6322
Prasyarat dan batasan 6323
Arsitektur 6324
Alat 6325
Praktik terbaik 6326
Epik 6326
Sumber daya terkait 6333
Informasi tambahan 6334
Terapkan AWS Control Tower kontrol menggunakan Terraform 6337
Ringkasan 6337
Prasyarat dan batasan 6338
Arsitektur 6339
Alat 6339
Praktik terbaik 6340
Epik 6340
Pemecahan masalah 6345
Sumber daya terkait 6347
Informasi tambahan 6347
Terapkan Otomasi Keamanan untuk AWS WAF menggunakan Terraform 6350
Ringkasan 6350
Prasyarat dan batasan 6350
Arsitektur 6351
Alat 6351
Praktik terbaik 6352
Epik 6352
Pemecahan masalah 6355
Sumber daya terkait 6356
Menerapkan pipeline yang mendeteksi masalah keamanan dalam kode 6357
Ringkasan 6357
Prasyarat dan batasan 6357
Arsitektur 6357
Alat 6358
Epik 6359
Pemecahan masalah 6362
Sumber daya terkait 6362

Informasi tambahan 6362
Menyebarkan kontrol detektif untuk subnet publik 6365
Ringkasan 6365
Prasyarat dan batasan 6366
Arsitektur 6366
Alat 6367
Praktik terbaik 6368
Epik 6368
Sumber daya terkait 6376
Informasi tambahan 6377
Menyebarkan kontrol pencegahan untuk subnet publik 6380
Ringkasan 6380
Prasyarat dan batasan 6380
Arsitektur 6381
Alat 6382
Epik 6383
Sumber daya terkait 6389
Informasi tambahan 6389
Mendeteksi instans Amazon RDS dengan sertifikat CA yang kedaluwarsa 6392
Ringkasan 6392
Prasyarat dan batasan 6392
Arsitektur 6393
Alat 6395
Praktik terbaik 6396
Epik 6396
Pemecahan masalah 6400
Sumber daya terkait 6401
Buat kebijakan IAM secara dinamis dengan IAM Access Analyzer 6402
Ringkasan 6402
Prasyarat dan batasan 6402
Arsitektur 6403
Alat 6404
Epik 6406
Sumber daya terkait 6412
Aktifkan GuardDuty menggunakan CloudFormation templat 6413
Ringkasan 6413

Prasyarat dan batasan 6413
Arsitektur 6414
Alat 6414
Epik 6415
Sumber daya terkait 6419
Informasi tambahan 6419
Aktifkan enkripsi data transparan di Amazon RDS for SQL Server 6421
Ringkasan 6421
Prasyarat dan batasan 6421
Arsitektur 6421
Alat 6422
Epik 6422
Sumber daya terkait 6425
Membantu mencegah penghapusan kunci KMS terjadwal 6426
Ringkasan 6426
Prasyarat dan batasan 6426
Arsitektur 6427
Alat 6428
Epik 6429
Sumber daya terkait 6432
Informasi tambahan 6433
Lampiran 6433
Identifikasi ember Amazon S3 publik di AWS Organizations 6434
Ringkasan 6434
Prasyarat dan batasan 6434
Arsitektur 6435
Alat 6436
Epik 6437
Pemecahan masalah 6441
Sumber daya terkait 6441
Informasi tambahan 6442
Menyerap log keamanan AWS ke Microsoft Sentinel 6443
Ringkasan 6443
Prasyarat dan batasan 6443
Arsitektur 6445
Alat 6446

Praktik terbaik 6447
Epik 6447
Sumber daya terkait 6457
Mengelola AWS Organizations kebijakan sebagai kode 6459
Ringkasan 6459
Prasyarat dan batasan 6460
Arsitektur 6460
Alat 6462
Praktik terbaik 6463
Epik 6463
Pemecahan masalah 6481
Sumber daya terkait 6483
Informasi tambahan 6484
Kelola set izin Pusat Identitas IAM menggunakan CodePipeline 6486
Ringkasan 6486
Prasyarat dan batasan 6486
Arsitektur 6488
Alat 6489
Praktik terbaik 6490
Epik 6490
Pemecahan masalah 6499
Sumber daya terkait 6500
Kelola kredensil dengan AWS Secrets Manager 6501
Ringkasan 6501
Prasyarat dan batasan 6501
Arsitektur 6501
Alat 6502
Epik 6502
Sumber daya terkait 6504
Informasi tambahan 6504
Pantau ElastiCache kluster Amazon untuk enkripsi saat istirahat 6507
Ringkasan 6507
Prasyarat dan batasan 6508
Arsitektur 6508
Alat 6509
Praktik terbaik 6510

Epik 6511
Pemecahan masalah 6514
Sumber daya terkait 6516
Pantau aktivitas pengguna root IAM 6517
Ringkasan 6517
Prasyarat dan batasan 6517
Arsitektur 6518
Alat 6518
Epik 6520
Sumber daya terkait 6525
Informasi tambahan 6525
Beri tahu saat pengguna IAM dibuat 6526
Ringkasan 6526
Prasyarat dan batasan 6526
Arsitektur 6526
Alat 6527
Epik 6528
Sumber daya terkait 6530
Lampiran 6531
Mencegah akses internet dengan menggunakan SCP 6532
Ringkasan 6532
Prasyarat dan batasan 6532
Alat 6533
Praktik terbaik 6533
Epik 6533
Sumber daya terkait 6536
Laporkan identitas IAM Identity Center dan tugasnya 6537
Ringkasan 6537
Prasyarat dan batasan 6537
Arsitektur 6539
Alat 6539
Epik 6539
Pemecahan masalah 6541
Sumber daya terkait 6543
Informasi tambahan 6543
Membatasi akses berdasarkan alamat IP atau geolokasi 6546

Ringkasan 6546
Prasyarat dan batasan 6546
Alat 6547
Epik 6548
Sumber daya terkait 6553
Pindai repositori Git untuk informasi sensitif 6554
Ringkasan 6554
Prasyarat dan batasan 6554
Arsitektur 6554
Alat 6554
Praktik terbaik 6555
Epik 6555
Sumber daya terkait 6560
Transfer file aman menggunakan Transfer Family dan GuardDuty 6561
Ringkasan 6561
Prasyarat dan batasan 6561
Arsitektur 6562
Alat 6564
Praktik terbaik 6565
Epik 6565
Pemecahan masalah 6571
Sumber daya terkait 6574
Informasi tambahan 6574
Mengamankan data sensitif di CloudWatch Log menggunakan Macie 6576
Ringkasan 6576
Prasyarat dan batasan 6576
Arsitektur 6577
Alat 6578
Praktik terbaik 6579
Epik 6579
Pemecahan masalah 6600
Sumber daya terkait 6600
Kirim peringatan dari AWS Network Firewall ke saluran Slack 6601
Ringkasan 6601
Prasyarat dan batasan 6601
Arsitektur 6602

Alat 6603
Epik 6604
Sumber daya terkait 6610
Informasi tambahan 6610
Kirim atribut khusus ke token Amazon Cognito 6615
Ringkasan 6615
Prasyarat dan batasan 6615
Arsitektur 6616
Alat 6617
Praktik terbaik 6617
Epik 6619
Pemecahan masalah 6628
Sumber daya terkait 6628
Informasi tambahan 6628
Sederhanakan manajemen sertifikat pribadi dengan menggunakan AWS Private CA dan AWS RAM 6632
Ringkasan 6632
Prasyarat dan batasan 6632
Arsitektur 6633
Alat 6634
Epik 6635
Sumber daya terkait 6642
Informasi tambahan 6643
Merampingkan manajemen EC2 kepatuhan Amazon dengan agen Amazon Bedrock dan AWS Config 6644
Ringkasan 6644
Prasyarat dan batasan 6644
Arsitektur 6645
Alat 6646
Praktik terbaik 6647
Epik 6647
Pemecahan masalah 6650
Sumber daya terkait 6650
Perbarui kredensyal dari IAM Identity Center menggunakan PowerShell 6651
Ringkasan 6651
Prasyarat dan batasan 6651

Arsitektur 6652
Alat 6652
Praktik terbaik 6653
Epik 6653
Pemecahan masalah 6655
Sumber daya terkait 6656
Informasi tambahan 6656
Gunakan Network Firewall untuk menangkap nama DNS dari lalu lintas keluar 6659
Ringkasan 6659
Prasyarat dan batasan 6659
Arsitektur 6660
Alat 6660
Epik 6661
Gunakan Terraform untuk mengaktifkan secara otomatis GuardDuty 6676
Ringkasan 6676
Prasyarat dan batasan 6677
Arsitektur 6679
Alat 6680
Epik 6681
Sumber daya terkait 6690
Informasi tambahan 6691
Verifikasi praktik terbaik untuk PCI DSS 4.0 6692
Ringkasan 6692
Prasyarat dan batasan 6693
Alat 6694
Epik 6694
Sumber daya terkait 6697
Informasi tambahan 6697
Lampiran 6698
Lebih banyak pola 6699
..... 6702

AWS Pola Panduan Preskriptif

Amazon Web Services (AWS) Pola Panduan Preskriptif memberikan step-by-step instruksi, arsitektur, alat, dan kode untuk menerapkan skenario migrasi, modernisasi, dan penerapan cloud tertentu. Pola-pola ini, yang diperiksa oleh para ahli materi pelajaran di AWS, dimaksudkan untuk pembangun dan pengguna langsung yang berencana, atau sedang dalam proses, bermigrasi ke. AWS Mereka juga mendukung pengguna yang sudah aktif AWS dan mencari cara untuk mengoptimalkan atau memodernisasi operasi cloud mereka.

Anda dapat menggunakan pola-pola ini untuk memindahkan beban kerja lokal atau cloud dengan berbagai kompleksitas ke AWS dan untuk mempercepat upaya adopsi, pengoptimalan, dan modernisasi cloud Anda, terlepas dari apakah Anda berada dalam tahap pembuktian konsep, perencanaan, atau implementasi proyek Anda. Misalnya, untuk proyek migrasi cloud:

- Pada tahap perencanaan, Anda dapat mengevaluasi berbagai opsi yang tersedia untuk dimigrasi. AWS Anda dapat memilih pola yang tepat yang sesuai dengan kebutuhan Anda, tergantung pada apakah Anda ingin pindah, rehost, replatform, atau rearsitek. Anda juga dapat memahami berbagai alat yang tersedia untuk migrasi, dan mulai merencanakan untuk mendapatkan lisensi atau memulai percakapan awal dengan vendor.
- Dalam pembuktian konsep dan fase implementasi, Anda dapat mengikuti step-by-step instruksi yang diberikan dalam pola untuk memigrasikan beban kerja Anda. AWS Setiap pola mencakup detail seperti prasyarat, arsitektur referensi target, alat, step-by-step tugas, praktik terbaik, pemecahan masalah, dan kode.
- Jika Anda sudah menggunakan AWS Cloud, Anda dapat menemukan pola yang akan membantu Anda memodernisasi, mengoptimalkan, menskalakan, dan mengamankan penggunaan sumber daya cloud Anda.

Untuk melihat daftar pola berdasarkan domain teknis, gunakan tautan berikut atau opsi pemfilteran dan pencarian di beranda Panduan AWS Preskriptif.

- Yayasan awan
- AI & pembelajaran mesin
- Analitik
- Hitung
- Database & penyimpanan

- Alat pengembang
- IoT
- Migrasi & modernisasi
- Manajemen
- Keamanan, identitas & kepatuhan

Untuk melihat semua publikasi, termasuk panduan, strategi, dan pola, lihat halaman beranda Panduan AWS Preskriptif.

Fondasi awan

Topik

- Mengotomatiskan pembuatan akun dengan menggunakan Akselerator Zona Landing di AWS
- Inventarisasi AWS sumber daya secara otomatis di beberapa akun dan Wilayah
- Bangun AWS landing zone yang mencakup MongoDB Atlas
- Konfigurasikan Log Aliran VPC untuk sentralisasi di seluruh Akun AWS
- Kelola set AWS izin secara dinamis dengan menggunakan Terraform
- Tandai lampiran Transit Gateway secara otomatis menggunakan AWS Organizations
- Lebih banyak pola

Mengotomatiskan pembuatan akun dengan menggunakan Akselerator Zona Landing di AWS

Justin Kuskowski, Joe Behrens, dan Nathan Scott, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan Landing Zone Accelerator pada AWS solusi untuk secara otomatis menerapkan baru Akun AWS ketika pengguna yang berwenang mengirimkan permintaan. Ini digunakan AWS Step Functions untuk mengatur sejumlah fungsi. AWS Lambda Fungsi Lambda menambahkan informasi akun ke repositori Git, memulai AWS CodePipeline pipeline, dan memvalidasi bahwa sumber daya yang diperlukan disediakan. AWS Ketika proses selesai, pengguna menerima pemberitahuan bahwa akun telah dibuat.

Secara opsional, Anda dapat mengintegrasikan grup ID Microsoft Entra dan menetapkan set AWS IAM Identity Center izin selama proses pembuatan akun. Jika organisasi Anda menggunakan ID Microsoft Entra sebagai sumber identitas, fitur opsional ini membantu Anda mengelola dan mengonfigurasi akses ke akun baru secara otomatis.

Prasyarat dan batasan

Prasyarat

- Akses ke akun manajemen di AWS Organizations
- AWS Cloud Development Kit (AWS CDK) versi 2.118.0 atau yang lebih baru, diinstal dan dikonfigurasi
- Python versi 3.9 atau yang lebih baru, diinstal
- AWS Command Line Interface (AWS CLI) versi 2.13.19 atau yang lebih baru, diinstal
- Docker versi 24.0.6 atau yang lebih baru, diinstal
- Akselerator Zona Pendaratan pada AWS solusi, digunakan di akun manajemen
- (Opsional) Microsoft Entra ID dan IAM Identity Center, terintegrasi

Batasan

Alur kerja pembuatan akun mendukung eksekusi berurutan untuk menerapkan satu. Akun AWS Batasan ini memastikan bahwa alur kerja pembuatan akun telah berhasil diselesaikan tanpa harus bersaing untuk sumber daya selama proses paralel.

Arsitektur

Arsitektur target

Gambar berikut menunjukkan arsitektur tingkat tinggi untuk mengotomatisasi pembuatan baru Akun AWS dengan menggunakan Akselerator Zona Pendaratan aktif. AWS AWS Step Functions mengatur otomatisasi. Setiap tugas dalam alur kerja Step Functions dilakukan oleh satu atau beberapa AWS Lambda fungsi.

Diagram menunjukkan alur kerja berikut:

1. Pengguna meminta akun dengan menjalankan skrip Python atau dengan menggunakan Amazon API Gateway.
2. Alur kerja Orkestrator Pembuatan Akun dimulai pada. AWS Step Functions
3. Alur kerja memperbarui `account-config.yaml` file di repositori kode sumber. Ini juga memulai Akselerator Zona Pendaratan pada AWS pipa dan memeriksa status pipa. Pipeline ini membuat dan menyiapkan akun baru. Untuk informasi selengkapnya tentang cara kerjanya, lihat ikhtisar Arsitektur untuk Akselerator Zona Pendaratan. AWS
4. (Opsional) Saat pipeline selesai, alur kerja memeriksa apakah grup ada di Microsoft Entra ID. Jika grup tidak ada di Microsoft Entra ID, alur kerja akan menambahkan grup ke Microsoft Entra ID.
5. Alur kerja melakukan langkah-langkah tambahan yang tidak dapat dilakukan oleh Landing Zone Accelerator pada AWS solusi. Langkah-langkah default meliputi:
   - Membuat alias akun di AWS Identity and Access Management (IAM)
   - Melampirkan tag ke akun di AWS Organizations
   - Membuat parameter di AWS Systems Manager Parameter Store berdasarkan tag yang ditetapkan ke akun
6. (Opsional) Alur kerja menetapkan satu atau beberapa set izin ke grup ID Microsoft Entra yang sebelumnya Anda tentukan. Set izin memungkinkan pengguna dalam grup untuk mengakses akun baru dan memungkinkan mereka untuk melakukan tindakan yang Anda konfigurasikan.
7. Sebuah AWS Lambda fungsi menjalankan tes QA dan validasi. Ini memvalidasi pembuatan sumber daya, memeriksa apakah tag telah dibuat, dan memvalidasi bahwa sumber daya keamanan telah digunakan.
8. Alur kerja merilis akun dan menggunakan Amazon Simple Email Service (Amazon SES) untuk memberi tahu pengguna bahwa proses telah berhasil diselesaikan.

Untuk informasi selengkapnya tentang alur kerja Step Functions, lihat diagram alur kerja Step Functions di bagian Informasi tambahan dari pola ini.

Aplikasi Microsoft Entra ID

Jika Anda memilih untuk mengintegrasikan dengan Microsoft Entra ID, Anda membuat dua aplikasi berikut ketika Anda menerapkan pola ini:

- Aplikasi yang ditautkan ke IAM Identity Center dan memastikan bahwa grup ID Microsoft Entra tersedia di IAM Identity Center. Dalam contoh ini, aplikasi Microsoft Entra ID ini diberi nama`LZA2`.
- Aplikasi yang memungkinkan fungsi Lambda untuk berkomunikasi dengan Microsoft Entra ID dan memanggil Microsoft Graph. APIs Dalam pola ini, aplikasi ini diberi nama`create_aws_account`.

Aplikasi ini mengumpulkan data yang digunakan untuk menyinkronkan grup ID Microsoft Entra dan menetapkan set izin.

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun. Dalam pola ini, Anda menggunakan API Gateway untuk memeriksa ketersediaan Akun AWS nama, memulai AWS Step Functions alur kerja, dan memeriksa status eksekusi Step Functions.
- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS Solusi ini menggunakan EventBridge aturan yang memulai fungsi Lambda jika status alur kerja Step Functions berubah menjadi`Failed`,, `Timed-out` atau. `Aborted`
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua aplikasi Anda Akun AWS dan cloud.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda. Dalam pola ini, AWS KMS kunci digunakan untuk mengenkripsi data, seperti data yang disimpan di Amazon Simple Storage Service (Amazon S3), variabel lingkungan Lambda, dan data di Step Functions.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon Simple Email Service (Amazon SES) membantu Anda mengirim dan menerima email dengan menggunakan alamat email dan domain Anda sendiri. Setelah akun baru berhasil dibuat, Anda menerima pemberitahuan melalui Amazon SES.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email. Jika terjadi kesalahan selama proses pembuatan akun, Amazon SNS mengirimkan pemberitahuan ke alamat email yang Anda konfigurasikan.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.
- AWS Systems Manager Parameter Store menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.

Alat-alat lainnya

- awscurl mengotomatiskan proses penandatanganan permintaan AWS API dan membantu Anda membuat permintaan sebagai perintah curl standar.
- Microsoft Entra ID, sebelumnya dikenal sebagai Azure Active Directory, adalah layanan manajemen identitas dan akses berbasis cloud.
- Microsoft Graph APIs membantu Anda mengakses data dan intelijen di layanan cloud Microsoft, seperti Microsoft Entra dan Microsoft 365.

Repositori kode

Kode untuk pola ini tersedia di GitHub lza-account-creation-workflowrepositori.

Direktori lambda_layer berisi lapisan berikut, yang direferensikan dalam lebih dari satu fungsi Lambda:

- account_creation_helper - Lapisan ini mencakup modul untuk mengasumsikan peran dan memeriksa kemajuan dalam. AWS Service Catalog
- boto3 - Lapisan ini mencakup AWS SDK for Python (Boto3)modul untuk memastikan bahwa AWS Lambda memiliki versi terbaru.
- identity_center_helper - Lapisan ini mendukung panggilan ke IAM Identity Center.

Direktori lambda_src berisi fungsi Lambda berikut:

- AccountTagToSsmParameter— Fungsi ini menggunakan tag yang dilampirkan ke akun untuk membuat parameter di Parameter Store. AWS Organizations Setiap parameter dimulai dengan `/account/tags/` awalan.
- AttachPermissionSet— Fungsi ini menambahkan izin yang disetel ke grup Pusat Identitas IAM.
- Azure ADGroup Sync - Fungsi ini menyinkronkan target grup Microsoft Entra ID ke IAM Identity Center.
- CheckForRunningProcesses— Fungsi ini memeriksa apakah `AWSAccelerator-Pipeline` pipeline sedang berjalan. Jika pipeline sedang berjalan, fungsi menunda AWS Step Functions alur kerja.
- CreateAccount— Fungsi ini menggunakan AWS Service Catalog dan AWS Control Tower membuat yang baru Akun AWS.
- CreateAdditionalResources— Fungsi ini menciptakan AWS sumber daya yang tidak dikelola oleh Landing Zone Accelerator atau AWS CloudFormation, seperti alias akun dan AWS Service Catalog tag.
- GetAccountStatusFungsi ini memindai produk yang disediakan AWS Service Catalog untuk menentukan apakah proses pembuatan akun telah selesai.
- GetExecutionStatus— Fungsi ini mengambil status AWS Step Functions eksekusi yang sedang berjalan atau selesai.
- NameAvailability— Fungsi ini memeriksa apakah Akun AWS nama sudah ada di AWS Organizations.
- ReturnResponse— Jika pembuatan akun berhasil, fungsi ini mengembalikan ID akun baru. Jika pembuatan akun tidak berhasil, ia mengembalikan pesan kesalahan.

- RunStepFunction— Fungsi ini menjalankan AWS Step Functions alur kerja yang membuat akun.
- SendEmailWithSES — Fungsi ini mengirimkan email ke pengguna yang menunggu pembuatan akun selesai.
- Validasi ADGroup SyncTo SSO - Fungsi ini memeriksa apakah grup ID Microsoft Entra yang ditentukan disinkronkan dengan IAM Identity Center.
- ValidateResources— Fungsi ini memvalidasi bahwa semua AWS Control Tower penyesuaian telah berjalan dengan sukses.

Praktik terbaik

Kami merekomendasikan konvensi penamaan berikut untuk: AWS CDK

- Mulai semua parameter dengan `p` awalan.
- Mulai semua kondisi dengan `c` awalan.
- Mulai semua sumber daya dengan `r` awalan.
- Mulai semua output dengan `o` awalan.

Epik

Menyebarkan peran IAM untuk validasi dan penandaan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Akselerator Zona Pendaratan AWS untuk penyesuaian.	1. Di Landing Zone Accelerator pada repositori AWS kode, buat file bernama. `customizations-config.yaml` Anda menggunakan file ini untuk menentukan kustomisasi untuk solusi inti. Untuk informasi selengkapnya, lihat Menyesuaikan solusinya.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Dalam `customizations-config.yaml` file, buat bagian yang disebut `cloudFormationStacks` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersiaplah untuk menyebarkan `lza-account-creation-validation` peran.	Sekarang, Anda menyesuaikan solusi untuk menerapkan peran `lza-account-creation-validation` IAM di semua akun selain akun manajemen. Peran ini menyediakan fungsi `ValidateResources` Lambda dengan akses hanya-baca ke akun baru. 1. Unduh account-creation-validation-rolefile.yaml dari file. GitHub 2. Simpan file ke lokasi yang ditunjukkan di bagian templat `customizations-config.yaml` file. 3. Buka file `customizations-config.yaml` . 4. Di `cloudFormationStacks` bagian ini, tambahkan kode berikut. Perbarui target Wilayah AWS sesuai kebutuhan untuk landing zone Anda: `- deploymentTargets: organizationalUnits: - Root excludedAccounts:`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>-
	Management
	
	descripti
	on: IAM Role to allow
	Account Validation
	name: lza-
	account-creation-
	validation
	regions:
	- us-
	east-1
	template:
	cloudformation/acc
	ount-creation-vali
	dation-role.yaml
	runOrder: 1
	terminati
	onProtection: true
	parameters:
	- name:
	pManagementAccount
	Id
	value:
	"{{ account Managemen
	t }}"</pre> 5. Simpan dan tutup file `customizations-config.yaml`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersiaplah untuk menyebarkan `account-tagging-to-ssm-parameter-role` peran.	Sekarang, Anda menyesuaikan solusi untuk menerapkan peran `account-tagging-to-ssm-parameter-role` IAM di semua akun selain akun manajemen. Peran ini digunakan untuk membuat parameter di AWS Systems Manager Parameter Store. 1. Unduh file account-tagging-to-ssm-parameter-role.yaml dari. GitHub 2. Simpan file ke lokasi yang ditunjukkan di bagian templat `customizations-config.yaml` file. 3. Buka file `customizations-config.yaml` . 4. Di `cloudFormationStacks` bagian ini, tambahkan kode berikut. Perbarui target Wilayah AWS sesuai kebutuhan untuk landing zone Anda: `- deploymentTargets: organizationalUnits: - Root excludedAccounts: - Management`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> descripti
	on: IAM Role to
	 create SSM Parameter
	s based on Account
	 Tagging
	          name: lza-
	account-tagging-to-
	ssm-parameter
	          regions:
	            - us-
	east-1
	          template:
	 cloudformation/acc
	ount-tagging-to-ss
	m-parameter-role.y
	aml
	          runOrder: 1
	          terminati
	onProtection: true
	
	          parameters:
	            - name:
	 pManagementAccount
	Id
	              value:
	 "{{ account Managemen
	t }}"</pre> 5. Simpan dan tutup file `customizations-config.yaml`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersiaplah untuk menyebarkan `config-log-validation-role` peran.	Sekarang, Anda menyesuaikan solusi untuk menerapkan peran `config-log-validation-role` IAM di akun arsip log. Peran ini memungkinkan fungsi `ValidateResources` Lambda mengakses bucket Amazon S3 untuk aturan logging dan akses. AWS Config 1. Unduh config-log-validation-rolefile.yaml dari file. GitHub 2. Simpan file ke lokasi yang ditunjukkan di bagian templat `customizations-config.yaml` file. 3. Buka file `customizations-config.yaml` . 4. Di `cloudFormationStacks` bagian ini, tambahkan kode berikut. Perbarui target Wilayah AWS sesuai kebutuhan untuk landing zone Anda: `- deploymentTargets: accounts: - LogArchive description: IAM Role to`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	validate Config and Logs name: lza-config-log-validation-role regions: - us-east-1 template: cloudformation/config-log-validation-role.yaml runOrder: 1 terminationProtection: true parameters: - name: pManagementAccountId value: "{{ account Management }}" 5. Simpan, tutup, dan komit perubahan yang dibuat pada `customizations-config.yaml` file.	

(Opsional) Dapatkan data dari Microsoft Entra ID

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aplikasi yang memungkinkan fungsi Lambda untuk berkomunikasi dengan Microsoft Entra ID.	1. Di pusat admin Microsoft Entra ID, daftarkan `create_aws_account` aplikasi. Untuk petunjuk,	ID Microsoft Entra

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	lihat Mendaftarkan aplikasi di dokumentasi Microsoft.
	2. Ikuti petunjuk di Perbarui izin yang diminta aplikasi dalam dokumentasi Microsoft untuk mengonfig urasi izin Microsoft Graph berikut untuk aplikasi: `create_aws_account`
	• Aplikasi.Baca.Semua
	  • Jenis: Aplikasi
	  • Diperlukan persetujuan admin: Ya
	• AppRoleAssignment. ReadWrite.Semua
	  • Jenis: Aplikasi
	  • Diperlukan persetujuan admin: Ya
	• Kelompok. ReadWrite .Semua
	  • Jenis: Aplikasi
	  • Diperlukan persetujuan admin: Ya
	• ServicePrincipalEn dpoint.Baca.Semua
	  • Jenis: Aplikasi
	  • Diperlukan persetujuan admin: Ya
	• Sinkronisasi. ReadWrite .Semua
	  • Jenis: Aplikasi	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Diperlukan persetujuan admin: Ya
	• Pengguna.Baca
	  • Jenis: Delegasikan
	  • Diperlukan persetujuan admin: Tidak	

Ambil nilai untuk `create_aws_account` aplikasi.	Sekarang, Anda mengambil nilai-nilai yang Anda butuhkan untuk `create_aws_account` aplikasi.
	1. Di pusat admin Microsoft Entra ID, navigasikan ke Pendaftaran aplikasi, lalu pilih. `create_aws_account`
	2. Di panel kiri, pilih Ikhtisar.
	3. Pada halaman Ikhtisar, catat nilai-nilai berikut:
	  • ID Aplikasi (klien)
	  • ID Direktori (penyewa)
	4. Di panel kiri, di bawah Kelola, pilih Sertifikat & rahasia.
	5. Pada halaman Sertifikat & Rahasia, pilih tab Rahasia klien, lalu catat nilai-nilai berikut:
	  • Nilai rahasia klien
	  • ID rahasia klien
		ID Microsoft Entra

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aplikasi yang menginteg rasikan Microsoft Entra ID dengan IAM Identity Center.	Di pusat admin Microsoft Entra ID, daftarkan `LZA2` aplikasi. Untuk petunjuk, lihat Mendaftarkan aplikasi di dokumentasi Microsoft.	ID Microsoft Entra
Ambil nilai untuk `LZA2` aplikasi.	Sekarang, Anda mengambil nilai-nilai yang Anda butuhkan untuk `LZA2` aplikasi. 1. Di pusat admin Microsoft Entra ID, navigasikan ke aplikasi Enterprise, lalu pilih`LZA2`. 2. Di panel kiri, pilih Ikhtisar. 3. Pada halaman Ikhtisar, catat nilai-nilai berikut: • Nama • ID Objek 4. Di panel kiri, di bawah Kelola, pilih Manifest. 5. Di file JSON, di `appRoles` bagian ini, cari peran aplikasi bernama`User`. 6. Catat `id` nilai untuk peran aplikasi ini.	ID Microsoft Entra

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat rahasia.	1. Dalam AWS CLI, masukkan perintah berikut untuk membuat variabel. Gunakan nilai yang Anda ambil untuk LZA2 aplikasi `create_aws_account` dan:
	`# Variables for create_aws_account app TENANT_ID='<Directory ID>' CLIENT_ID='<Application ID>' SECRET_ID='<Client secret ID>' SECRET_VALUE='<Client secret value>'`
	`# Variables for LZA2 app OBJECT_ID='<Object ID>' APP_ROLE_ID='<App role ID>' ENTERPRISE_APP_NAME='<Name>'`
	2. Masukkan perintah berikut untuk membuat rahasia bernama `GraphApiSecret` AWS Secrets Manager:
	`aws secretsmanager create-secret \ --name GraphApiSecret \`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--secret-string
	"{\"client_id\":
	\"${CLIENT_ID}\",
	\"tenant_id\":
	\"${TENANT_ID}\",
	\"object_id\":
	\"${OBJECT_ID}\",
	\"app_role_id\":
	\"${APP_ROLE_ID}\"
	, \"secret_value\":
	\"${SECRET_VALUE}\
	", \"secret_id\":
	\"${SECRET_ID}\"}"</pre> Jika Anda perlu memperbarui rahasia di masa depan, Anda dapat memperbarui variabel dan menjalankan perintah berikut: <pre>aws secretsmanager
	update-secret \
	--secret-id
	GraphApiSecret \
	--secret-string
	"{\"client_id\":
	\"${CLIENT_ID}\",
	\"tenant_id\":
	\"${TENANT_ID}\",
	\"object_id\":
	\"${OBJECT_ID}\",
	\"app_role_id\":
	\"${APP_ROLE_ID}\"
	, \"secret_value\":
	\"${SECRET_VALUE}\
	", \"secret_id\":
	\"${SECRET_ID}\"}"</pre>	

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning kode sumber.	1. Masukkan perintah berikut untuk mengkloning lza-account-creation-workflowrepositori:
	`git clone https://github.com/aws-samples/lza-account-creation-workflow`
	2. Masukkan perintah berikut untuk menavigasi ke `config` direktori repositori kloning:
	`cd lza-account-creation-workflow/config`
		DevOps insinyur

Perbarui `deploy-config.yaml` file.	1. Buka file `deploy-config.yaml` . 2. Tinjau template dan perbarui nilai yang diperlukan untuk penerapan di AWS lingkungan Anda. Misalnya, perbarui nilai untuk yang berikut ini: • `accountCreationFailure` • `accountCompletionFromEmail` • `ssoLoginUrl` • `rootEmailPrefix` • `rootEmailDomain`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Jika Anda berintegrasi dengan Microsoft Entra ID, lakukan hal berikut: • Atur `enableAzureADIntegration` ke `true`. • Untuk `graphApiSecretName` nilai, masukkan rahasia yang Anda buat sebelumnya (`GraphApiSecret`). 4. Simpan dan tutup file `deploy-config.yaml`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan solusi di AWS lingkungan Anda.	1. Masukkan perintah berikut:
	`cdk bootstrap <account-number>/<Region>`
	Untuk informasi selengkapnya, lihat Bootstrapping di dokumentasi. AWS CDK
	2. Masukkan perintah berikut untuk mensintesis CloudFormation template:
	`cdk synth`
	Untuk informasi selengkapnya, lihat Mengkonfigurasi dan melakukan sintesis AWS CDK tumpukan dalam AWS CDK dokumentasi.
	3. Masukkan perintah berikut untuk menyebarkan solusi.
	`cdk deploy`
	Untuk informasi selengkapnya, lihat Menerapkan AWS CDK aplikasi dalam AWS CDK dokumentasi.
	Note
	Solusi ini menggunakan bucket Amazon
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	S3 untuk menyimpan kode sumber untuk solusi ini. Anda dapat menggunakan skrip upload_to_source_bucket.py untuk membuat arsip kode sumber dan mengunggah versi yang diperbarui.	

Opsi 1 - Buat akun menggunakan Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi argumen mana yang akan digunakan.	Pilih argumen mana yang akan digunakan saat Anda menjalankan skrip Python yang memulai alur kerja Step Functions. Untuk daftar lengkap argumen, lihat bagian Informasi tambahan dari pola ini.	AWS DevOps, Python
Mulai skrip Python.	1. Masukkan perintah berikut untuk menavigasi ke repositori kloning: `cd lza-account-creation-workflow` 2. Jalankan skrip Python yang memulai alur kerja Step Functions. Berikut	DevOps insinyur, Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ini adalah contoh perintah yang mencakup contoh argumen dan nilai-nilai: `python ./run-stepfunction.py \ --account-name "lza-test-01" \ --support-dl "johnsmith@example.com" \ --managed-org-unit "Workloads/Workload-1" \ --purpose "Testing new micro service" \ --force-update true \ --ad-integration "{\"CustomerAccountAdmin\": \"platform-admin\", \"CustomerAccountDev\": \"workload1-app1\"}" \ --bypass-creation true \ --tags APPLICATION=TestingMicroService` 3. Tunggu hingga Anda menerima pemberitahuan bahwa akun telah berhasil dibuat. **Note** Jika proses pembuatan akun	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	gagal, Amazon SNS mengirimk an pemberitahuan ke alamat email yang Anda tentukan `accountCr eationFai lure` dalam file. `deploy-co nfig.yaml` Pemohon akun tidak diberitahu.	

Opsi 2 - Buat akun menggunakan API Gateway dan awscurl

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan variabel untuk awscurl.	1. Masukkan perintah berikut untuk menavigasi ke direktori kode sumber: `cd lza-account-creati on-workflow` 2. Masukkan perintah berikut untuk mengatur variabel kunci AWS akses. Anda dapat menyalin variabel dari portal AWS akses dan kemudian menempelkannya ke dalam shell. Berikut ini adalah contohnya:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` export AWS_ACCES S_KEY_ID="<id>" export AWS_SECRE T_ACCESS_KEY="<key >" export AWS_SESSI ON_TOKEN="<token>" ``` 3. Masukkan perintah berikut untuk mengatur panggilan API Wilayah AWS untuk: ``` export AWS_REGION= $(aws configure get region) ``` 4. Masukkan perintah berikut untuk mengambil titik akhir API Gateway dari output: `lza-account-creati on-workflow-applic ation` CloudFormation ``` export API_GATEW AY_ENDPOINT=$(aws cloudformation describe-stacks -- stack-name "lza-acco unt-creation-workf low-application" --query 'Stacks[*].Outputs[?OutputK ey==`oApiGatewayCr eateAccountEndpoin t`].OutputValue' -- output text) ```	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa ketersediaan nama.	Masukkan perintah berikut untuk memverifikasi bahwa nama tersebut tersedia untuk Akun AWS. Ganti <AWS_ACCOUNT_NAME> dengan nama akun target: `awscurl --service execute-api \ --region ${AWS_REGION} \ --access_key ${AWS_ACCESS_KEY_ID} \ --secret_key ${AWS_SECRET_ACCESS_KEY} \ --security_token ${AWS_SESSION_TOKEN} \ -X POST ${API_GATEWAY_ENDPOINT}check_name?account_name=<AWS_ACCOUNT_NAME>`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan alur kerja pembuatan akun.	1. Di folder root repositori kloning, buka file. `api_example.json` 2. Perbarui parameter dengan nilai konfigurasi Anda: `{ "account_name": "lza-test-01", "account_email": "johnsmith+lzatest01@example.com", "support_dl": "johnsmith@example.com", "managed_org_unit": "Workloads/Workload-1", "ad_integration": [{ "PermissionSetName": "CustomerAccountAdmin", "ActiveDirectoryGroupName": "platform-admin" }, { "PermissionSetName": "CustomerAccountDev", "ActiveDirectoryGroupName": "workload1-app1" }],`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> "force_update":
	 "true",
	    "bypass_c
	reation": "false",
	    "account_tags": [
	        {
	            "Key":
	 "Environment",
	            "Value":
	 "Dev"
	        }, {
	            "Key":
	 "DeploymentMethod",
	            "Value":
	 "ApiGateway"
	        }
	    ]
	}</pre>	

	3. Simpan dan tutup file `api_example.json` .	

	4. Masukkan perintah berikut untuk memulai alur kerja Step Functions:
	<pre>awscurl --service
	 execute-api \
	    --data @api-exam
	ple.json \
	    --region ${AWS_REG
	ION} \
	    --access_key
	 ${AWS_ACCESS_KEY_I
	D} \
	    --secret_key
	 ${AWS_SECRET_ACCES
	S_KEY} \
	    --security_token
	 ${AWS_SESSION_TOKE
	N} \</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>-X POST ${API_GATEWAY_ENDPOINT}execute</pre>
	5. Pada output dari perintah sebelumnya, catat eksekusi Step Functions Amazon Resource Name (ARN).
	6. Jika Anda ingin memeriksa status alur kerja Step Functions, masukkan perintah berikut. Ganti `<STEP_FUNCTION_EXECUTION_NAME>` dengan eksekusi Step Functions ARN atau nama:
	<pre>awscurl --service execute-api \
	--region ${AWS_REGION} \
	--access_key ${AWS_ACCESS_KEY_ID} \
	--secret_key ${AWS_SECRET_ACCESS_KEY} \
	--security_token ${AWS_SESSION_TOKEN} \
	-X GET ${API_GATEWAY_ENDPOINT}get_execution_status?execution=<STEP_FUNCTION_EXECUTION_NAME></pre>
	7. Tunggu hingga Anda menerima pemberitahuan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bahwa akun telah berhasil dibuat. **Note** Jika proses pembuatan akun gagal, Amazon SNS mengirimkan pemberitahuan ke alamat email yang Anda tentukan `accountCreationFailure` dalam file. `deploy-config.yaml` Pemohon akun tidak diberitahu.	

(Opsional) Bersihkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus objek dari ember Amazon S3.	Hapus objek apa pun di bucket Amazon S3 berikut: • `lza-account-creation-work-<CDK_UNIQUE_ID>` • `lza-account-creation-workflow-src-<AWS_REGION>-<AWS_ACCOUNT>`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `lza-account-creation-workflow-<AWS_REGION>-<AWS_ACCOUNT>`	

Hapus CloudFormation tumpukan.	Masukkan perintah berikut untuk menghapus CloudFormation tumpukan:
	`aws cloudformation delete-stack \ --stack-name lza-account-creation-workflow-application`
	`aws cloudformation wait stack-delete-complete \ --stack-name lza-account-creation-workflow-application`
		AWS DevOps

Hapus pipa.	Masukkan perintah berikut untuk menghapus `lza-account-creation-workflow-pipeline` pipeline: `cdk destroy lza-account-creation-workflow-pipeline --force`	AWS DevOps

Sumber daya terkait

- Akselerator Zona Pendaratan AWS aktif (Perpustakaan AWS Solusi)
- Memecahkan AWS CDK masalah umum (dokumentasi)AWS CDK

Informasi tambahan

Diagram alur kerja Step Functions

Gambar berikut menunjukkan status dalam alur kerja Step Functions.

Argumen

Berikut ini adalah argumen yang dapat Anda gunakan ketika Anda menjalankan script Python yang memulai alur kerja Step Functions.

Argumen berikut diperlukan:

- `account-name (-a)`(string) — Nama yang baru Akun AWS.
- `support-dl (-s)`(string) — Alamat email yang menerima pemberitahuan saat proses pembuatan akun selesai.
- `managed-org-unit (-m)`(string) — Unit organisasi terkelola (OU) yang akan berisi akun baru.

Argumen berikut adalah opsional:

- `ad-integration (-ad)`(kamus string) - Grup ID Microsoft Entra dan set izin yang ditetapkan. Berikut ini adalah contoh bagaimana menggunakan argumen ini:

```
--ad-integration "{\"<PermissionSetName>\": \"<EntraIdGroupName>\"}"
```

- `account-email (-e)`(string) — Alamat email untuk pengguna root yang baru Akun AWS.

> **Note**
>
> Jika argumen ini tidak digunakan, alamat email akan dihasilkan dengan menggunakan nilai `rootEmailPrefix` dan `rootEmailDomain` dari `configs/deploy-config.yaml` file. Jika alamat email tidak diberikan, alamat email dibuat dengan menggunakan format berikut: `rootEmailPrefix+accountName@rootEmailDomain`

- `region (-r)`(string) - Wilayah AWS Tempat alur kerja Step Functions digunakan. Nilai default-nya adalah `us-east-1`.
- `force-update (-f)`(string Boolean) - Masukkan `true` AWS Service Catalog untuk memaksa memperbarui produk yang disediakan.

- `bypass-creation (-b)`(string Boolean) - Masukkan `true` untuk memotong menambahkan akun ke `accounts-config.yaml` file dan melewati menjalankan pipeline. `AWSAccelerator-Pipeline` Argumen ini biasanya digunakan untuk menguji proses alur kerja pembuatan akun atau untuk menjalankan langkah-langkah Step Functions lainnya jika terjadi kesalahan dalam `Landing Zone Accelerator` pipeline.
- `tags (-t)`(string) — Tag tambahan yang ingin Anda tambahkan ke Akun AWS. Secara default, tag berikut ditambahkan:`account-name`,`support-dl`, dan`purpose`. Berikut ini adalah contoh bagaimana menggunakan argumen ini:

```
--tags TEST1=VALUE1 TEST2=VALUE2
```

Inventarisasi AWS sumber daya secara otomatis di beberapa akun dan Wilayah

Matej Macek, Amazon Web Services

Ringkasan

Pola ini menguraikan pendekatan otomatis untuk mempertahankan inventaris AWS sumber daya yang komprehensif di beberapa akun dan Wilayah AWS. Ini dirancang untuk membantu insinyur infrastruktur dan keamanan meningkatkan praktik manajemen sumber daya mereka. Ini digunakan AWS Config untuk melacak perubahan sumber daya, Amazon Athena untuk kueri, dan Amazon Quick Sight untuk dasbor interaktif. Anda menerapkan solusi ini dengan menerapkan AWS CloudFormation tumpukan.

Solusi ini mirip dengan yang disajikan dalam Visualisasi AWS Config data menggunakan Amazon Athena dan Amazon Quick Sight AWS (posting blog). Pola ini memperluas solusi itu untuk mengatasi persyaratan umum berikut dan memberikan manfaat utama berikut:

- Berfokus pada kepatuhan - Pendekatan ini dapat membantu Anda memenuhi persyaratan peraturan seperti PCI DSS, NIST SP 800-53, ISO/IEC 27001, HIPAA, GDPR, dan lainnya yang mengamanatkan inventaris aset yang akurat.
- Kerangka kustomisasi - Ini memberikan dasar untuk membuat dasbor Quick Sight untuk berbagai AWS sumber daya, sehingga Anda dapat menyesuaikan solusi dengan kebutuhan spesifik Anda.
- Peningkatan yang digerakkan oleh pengguna — Pendekatan ini menggabungkan umpan balik dari kasus penggunaan dunia nyata dan menangani permintaan untuk solusi yang lebih komprehensif.

Tim infrastruktur, keamanan, dan keuangan sering menghadapi tantangan visibilitas dan kolaborasi di lingkungan dinamis, multi-akun, atau Multi-wilayah. Solusi ini dirancang untuk mengatasi tantangan tersebut dan secara signifikan mengurangi waktu dan upaya yang diperlukan untuk membuat dan memelihara inventaris sumber daya. Hasilnya adalah pandangan terpusat sumber daya yang membantu Anda meningkatkan keputusan alokasi sumber daya, mengidentifikasi dan mengurangi risiko, mengoptimalkan biaya, dan meningkatkan visibilitas dan kolaborasi secara keseluruhan. Pendekatan ini menjembatani kesenjangan antara solusi konseptual dan kebutuhan implementasi dunia nyata untuk tujuan keamanan, kepatuhan, dan operasional.

Prasyarat dan batasan

Prasyarat

- Berikut ini aktif Akun AWS:
  - Akun manajemen - Akun terpusat untuk penagihan, membuat akun, dan mengendalikan akses di seluruh organisasi
  - Akun audit — Hub terpusat untuk pemantauan keamanan, pemeriksaan kepatuhan, dan pemberitahuan drift
  - Akun arsip log — Akun terpusat untuk menyimpan dan menganalisis data yang dikumpulkan
- Di akun audit, AWS Config agregator yang mengumpulkan dan mengumpulkan data konfigurasi dari akun target dan Wilayah Anda
- Di akun arsip log, siapkan yang berikut ini:
  - Bucket Amazon Simple Storage Service (Amazon S3) tempat Anda menyimpan data dari agregator AWS Config
  - Langganan Amazon Quick Suite
  - Koneksi resmi antara Quick Sight dan Amazon Athena
  - Izin untuk mengakses bucket Amazon S3 melalui kueri Athena
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Izin untuk menerapkan CloudFormation tumpukan yang menyediakan sumber daya berikut:
  - Sebuah AWS Lambda fungsi
  - Konfigurasi notifikasi Amazon S3
  - Database Athena, tabel, dan tampilan
  - Kumpulan data Quick Sight dan sumber data
- Izin untuk menjalankan otomatisasi di AWS Systems Manager
- Izin untuk mengakses Quick Suite

Batasan

- Solusinya bergantung pada AWS Config. AWS Config biasanya mencatat perubahan konfigurasi ke sumber daya Anda tepat setelah perubahan terdeteksi, atau pada frekuensi yang Anda tentukan. Namun, ini adalah upaya terbaik dan kadang-kadang bisa memakan waktu lebih lama.
- Solusi ini hanya melacak jenis sumber daya yang AWS Config mendukung.

- Solusinya tidak melacak inventaris sumber daya di penyedia cloud lain atau lingkungan lokal.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat halaman titik akhir dan kuota Layanan di AWS dokumentasi, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan proses yang efisien untuk mengumpulkan, mengatur, menganalisis, dan memvisualisasikan konfigurasi dan data kepatuhan di beberapa akun dalam suatu AWS organisasi.

Diagram menunjukkan alur kerja berikut:

1. Pada jadwal berkala, AWS Config agregator mengumpulkan data konfigurasi dan kepatuhan tentang sumber daya di akun target dan Wilayah dan kemudian mengirimkan data ke bucket Amazon S3 di akun arsip log.
2. Menambahkan AWS Config data baru ke bucket Amazon S3 akan memanggil fungsi. AWS Lambda
3. Fungsi Lambda mempartisi data dengan mengonfigurasi kunci dengan nilai yang sesuai dengan Wilayah dan tanggal setiap file snapshot. Ini membantu AWS Glue secara efisien menanyakan dan memproses konfigurasi dan data kepatuhan.
4. Amazon Athena menggunakan AWS Glue skema untuk menjalankan kueri SQL terhadap data yang disimpan di bucket Amazon S3. Ini menggunakan metadata skema dari AWS Glue untuk memahami struktur data.
5. Tampilan di Athena menentukan dan mengekstrak kumpulan data target.
6. Dasbor di Quick Sight membantu Anda memvisualisasikan dan menganalisis kumpulan data.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS

- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu. AWS Config Agregator mengumpulkan data AWS Config konfigurasi dan kepatuhan dari beberapa Akun AWS dan Wilayah.
- AWS Glueadalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data. Pola ini menggunakan AWS Glue Data Catalog dan Schema registry.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon Quick Sight adalah layanan intelijen bisnis (BI) yang membantu Anda mengubah data mentah menjadi wawasan yang bermakna melalui visualisasi, dasbor, dan laporan interaktif. Quick Sight adalah komponen inti dari Amazon Quick Suite.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar. AWS Systems Manager Otomasi menyederhanakan tugas pemeliharaan, penyebaran, dan remediasi umum bagi banyak orang. Layanan AWS

Repositori kode

AWS CloudFormation Template untuk pola ini tersedia di GitHub repositori AWS Config visualisasi. CloudFormation Template ini menyebarkan runbook AWS Systems Manager otomatisasi yang disiapkan AWS Config untuk digunakan dengan Amazon Athena. Otomatisasi ini bersiap AWS Glue untuk terhubung dengan bucket Amazon S3 yang ditunjuk, membuat tampilan di Amazon Athena, dan mengonfigurasi Quick Sight untuk visualisasi dasbor.

Praktik terbaik

- Kami menyarankan Anda mengikuti praktik terbaik dalam Mengatur dan mengatur AWS lingkungan multi-akun yang aman dengan Panduan AWS Control Tower AWS Preskriptif.
- Kami menyarankan Anda membuat AWS Config agregator yang mengumpulkan data konfigurasi dan kepatuhan untuk seluruh AWS organisasi. Untuk informasi selengkapnya, lihat Agregasi Data Multi-Wilayah Multi-Akun dalam dokumentasi. AWS Config
- Sebelum menerapkan solusi ini, kami sarankan Anda meninjau informasi harga saat ini untuk Amazon S3 AWS Config, Athena, dan Quick Suite.

Epik

Menyebarkan tumpukan CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh CloudFormation template.	Unduh template Config-QuickSight CloudFormation - Visualization-SSM-Automation.yaml.	Administrator AWS, administrator Cloud, DevOps insinyur
Ubah CloudFormation template.	Selesaikan langkah ini hanya jika Anda menggunakan AWS Control Towerdan AWS Config dikelola oleh AWS Control Tower. Anda perlu memodifikasi CloudFormation template. 1. Masuk ke akun manajemen. 2. Buka konsol AWS Organizations. 3. Arahkan ke halaman Pengaturan. Halaman ini menampilkan detail tentang organisasi, termasuk ID organisasi.	DevOps insinyur, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Salin ID organisasi.
	5. Di editor teks pilihan Anda, buka file Config- QuickSight -Visualization-SSM-Automation.yaml.
	6. Temukan baris berikut:
	`return re.match( '^AWSLogs/(\d+)/Config/([\w-]+)/(\d+)/(\d+)/(\d+)/(\d+)/ConfigSnapshot/[^\\]+$', object_key)`
	7. Ganti baris ini dengan yang berikut, di `<ORGANIZATION_ID>` mana ID yang sebelumnya Anda salin:
	`return re.match( '^<ORGANIZATION_ID>/AWSLogs/(\d+)/Config/([\w-]+)/(\d+)/(\d+)/(\d+)/(\d+)/ConfigSnapshot/[^\\]+$', object_key)`
	8. Simpan dan tutup file Config- QuickSight -Visualization-SSM-Automation.yaml.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan.	Ikuti petunjuk di Buat tumpukan dari CloudFormation konsol. Perhatikan hal-hal berikut: 1. Pilih Unggah file templat, lalu pilih file YAMG yang Anda unduh. 2. Untuk Nama tumpukan, masukkan `Config-QuickSight-Visualization-SSM-Automation` . 3. Pilih Kirim.	Administrator AWS, administrator Cloud, DevOps insinyur

Jalankan otomatisasi di Systems Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Temukan nama pengguna Quick Suite Anda.	1. Buka konsol Quick Suite. 2. Buka menu profil. 3. Catat nama pengguna. Anda membutuhkan nilai ini nanti.	Administrator AWS, administrator Cloud, DevOps insinyur
Temukan nama saluran pengiriman dan nama bucket Amazon S3.	1. Di AWS CLI, masukkan perintah berikut: `aws configservice describe-delivery-channels` 2. Catat nama bucket Amazon S3 dan nama saluran AWS Config pengiriman Anda.	Administrator AWS, administrator Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda membutuhkan nilai-nilai ini nanti.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan otomatisasi di Systems Manager.	1. Buka konsol AWS Systems Manager. 2. Di panel navigasi, pilih Dokumen. 3. Pilih yang dimiliki oleh saya. 4. Pilih Config- QuickSight - Visualization. 5. Pilih Eksekusi otomatisasi. 6. Di bagian Parameter input, masukkan nilai Anda untuk parameter berikut: • `ConfigDeliveryChannelName` — Masukkan nama saluran AWS Config pengiriman Anda. Parameter ini diperlukan. • `ConfigS3BucketLocation` — Masukkan nama bucket Amazon S3 tempat Anda menyimpan data AWS Config konfigurasi. Parameter ini diperlukan. • `QuickSightUserName` — Masukkan nama pengguna yang memiliki akses administratif ke Quick Suite. Parameter ini diperlukan. • `AutomationAssumeRole` — Nama Sumber Daya Amazon (ARN) dari peran AWS	Administrator AWS, administrator Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Identity and Access Management (IAM) yang memungkinkan Otomasi Systems Manager untuk melakukan tindakan atas nama Anda. Parameter ini bersifat opsional. Biarkan parameter ini kosong. • `DeleteConfigVisualization` — Pilih `false`. 7. Pilih Eksekusi.	

Visualisasikan data dalam Quick Sight

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Segarkan data.	Untuk menjadwalkan penyegaran kumpulan data sesuai dengan kebutuhan spesifik Anda, ikuti petunjuk di Menyegarkan data SPICE.	Administrator AWS, DevOps insinyur, administrator Cloud
Buat analisis.	Untuk membuat dasbor di Quick Sight yang membantu Anda memvisualisasikan sumber daya, ikuti petunjuk di Memulai analisis di Quick Sight.	Administrator Suite Cepat
Buat dasbor.	1. Setelah Anda selesai memodifikasi analisis Quick Sight, ikuti petunjuk di dasbor Penerbitan untuk	Administrator Suite Cepat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membuat dasbor. Dasbor adalah analisis yang dapat Anda bagikan dengan pengguna Quick Suite lainnya. 2. Ikuti petunjuk dalam Memberikan akses ke dasbor untuk berbagi dasbor dengan pengguna Quick Suite target Anda.	

(Opsional) Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya yang dibuat oleh otomatisasi Systems Manager.	1. Buka konsol AWS Systems Manager. 2. Di panel navigasi, pilih Dokumen. 3. Pilih yang dimiliki oleh saya. 4. Pilih Config- QuickSight - Visualization. 5. Pilih Eksekusi otomatisasi. 6. Di bagian Parameter input, untuk `DeleteConfigVisualization` parameter, masukkan `true`. 7. Pilih Eksekusi.	Administrator AWS, administrator Cloud, DevOps insinyur
Hapus CloudFormation tumpukan.	Untuk menghapus sumber daya di `Config-QuickSight-Visualization-SSM-Automatio`	Administrator AWS, administrator Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n tumpukan, ikuti petunjuk di Hapus tumpukan dari CloudFormation konsol.	

Pemecahan masalah

Isu	Solusi

Amazon Quick Suite mencoba untuk terhubung ke `us-east-1` Wilayah AWS, tetapi pembuatan sumber daya di Wilayah tersebut tidak diizinkan.	Kebijakan kontrol layanan membatasi langganan Anda ke Amazon Quick Suite di Wilayah ini. Dalam kebijakan kontrol layanan, tentukan target secara manual Wilayah AWS. Ganti `<REGION_ID>` dengan pengenal Wilayah yang sesuai:
	`https://<REGION_ID>.quicksight.aws.amazon.com/sn/start/dashboards`
	Berikut ini adalah contohnya:
	`https://eu-central-1.quicksight.aws.amazon.com/sn/start/dashboards`

Di Amazon Athena, Anda menemukan pesan berikut: `Before you run your first query, you need to set up a query result location in Amazon S3.`	Pastikan Anda telah menyiapkan ember Amazon S3 tempat Anda akan menyimpan hasil kueri dari Amazon Athena. Kemudian ikuti petunjuk di Tentukan lokasi hasil kueri menggunakan konsol Amazon Athena.

Sumber daya terkait

AWS dokumentasi

- AWS Config dokumentasi
- Dokumentasi Amazon Quick Suite

AWS posting blog

- Mengotomatiskan visualisasi AWS Config data dengan AWS Systems Manager
- Cara merekam perubahan konfigurasi sumber daya secara berkala dengan AWS Config

Sumber daya lainnya

- Pusat Pembelajaran Komunitas Amazon Quick Suite
- Galeri Komunitas Amazon Quick Suite

Bangun AWS landing zone yang mencakup MongoDB Atlas

Igor Alekseev, Amazon Web Services

Anuj Panchal, MongoDB

Ringkasan

Pola ini menjelaskan cara membangun AWS landing zone yang terintegrasi dengan cluster MongoDB Atlas. Infrastruktur secara otomatis digunakan dengan menggunakan skrip Terraform.

AWS Lingkungan multi-akun yang terstruktur dengan baik, yang disebut landing zone, menawarkan skalabilitas dan keamanan, terutama untuk perusahaan. Ini berfungsi sebagai fondasi untuk penyebaran cepat beban kerja dan aplikasi, dan membantu memastikan kepercayaan dalam keamanan dan infrastruktur. Membangun landing zone memerlukan pertimbangan yang cermat terhadap faktor teknis dan bisnis, termasuk struktur akun, jaringan, keamanan, dan manajemen akses. Pertimbangan ini harus selaras dengan pertumbuhan masa depan dan tujuan bisnis organisasi Anda.

Kasus penggunaan untuk pola ini meliputi yang berikut ini.

- Platform SaaS dan PaaS Perusahaan: Perangkat lunak multitenant as a service (SaaS) aplikasi dan platform sebagai layanan (PaaS) platform yang berjalan AWS dapat menggunakan pengaturan ini untuk membantu menyediakan akses pribadi yang aman ke MongoDB Atlas tanpa mengekspos data melalui internet publik.
- Industri yang sangat diatur: Perbankan, jasa keuangan, perawatan kesehatan, dan beban kerja pemerintah yang memerlukan kepatuhan ketat terhadap standar seperti Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), System and Organization Controls 2 (), dan General Data Protection Regulation (GDPRSOC2) mendapat manfaat dari:
  - Konektivitas pribadi terenkripsi melalui AWS PrivateLink
  - Multi-AZ ketersediaan tinggi set replika MongoDB
- AI/ML Beban kerja yang aman: Saluran pipa pelatihan atau inferensi di Amazon Bedrock, SageMaker Amazon AI, atau model AI khusus dapat mengambil dan menyimpan data dengan aman di MongoDB Atlas. PrivateLink
- Pemulihan bencana dan kelangsungan bisnis: Desain multi-AZ memastikan bahwa tidak ada satu pun kegagalan Availability Zone yang mengganggu beban kerja. Replika Atlas yang disetel di

seluruh Availability Zones memastikan failover otomatis. Ini sangat penting untuk layanan yang selalu aktif seperti aplikasi teknologi keuangan (fintech), perbankan digital, atau pemantauan kesehatan.

Prasyarat dan batasan

Prasyarat

- Pemilik organisasi mengakses MongoDB Atlas sehingga Anda dapat membuat kunci Atlas API. Untuk informasi tentang persyaratan ini, lihat Mengelola Akses Organisasi dalam dokumentasi MongoDB.
- Aktif Akun AWS.
- Terraform, diinstal dan dikonfigurasi.
- Cluster MongoDB Atlas, dibuat dengan MongoDB versi 6.0 atau yang lebih baru.
- Keakraban dengan MongoDB dan MongoDB Atlas. Untuk informasi selengkapnya, lihat dokumentasi MongoDB Atlas.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram arsitektur referensi berikut menggambarkan pengaturan penyebaran untuk AWS landing zone yang terintegrasi dengan titik akhir pribadi MongoDB Atlas. Arsitektur referensi ini menunjukkan bagaimana membangun AWS landing zone yang aman, terukur, dan sangat tersedia yang terintegrasi dengan MongoDB Atlas. Dengan menggabungkan praktik AWS terbaik seperti penyebaran multi-AZ, kontrol keamanan dengan hak istimewa, dan konektivitas pribadi, desain ini memungkinkan organisasi untuk menyediakan lingkungan yang kuat untuk aplikasi modern.

Arsitektur ini terdiri dari yang berikut:

VPC

- Satu virtual private cloud (VPC) mencakup tiga Availability Zone.
- VPC dibagi lagi menjadi subnet yang selaras dengan setiap Availability Zone. Subnet ini mendistribusikan beban kerja untuk ketersediaan tinggi.

Akses internet

- Gateway internet menyediakan konektivitas internet outbound untuk sumber daya yang membutuhkannya, seperti host aplikasi atau bastion.
- Subnet publik dapat menampung gateway NAT, yang memungkinkan beban kerja subnet pribadi untuk mengunduh pembaruan, tambalan, dan paket lain yang diperlukan tanpa memaparkannya langsung ke internet publik.

Subnet pribadi dan tabel rute

- Komponen aplikasi, layanan mikro, atau sumber daya sensitif lainnya biasanya berada di subnet pribadi.
- Tabel rute khusus mengontrol arus lalu lintas. Rutekan lalu lintas keluar langsung dari subnet pribadi ke gateway NAT untuk akses internet yang aman dan hanya jalan keluar.
- Permintaan masuk dari aliran internet melalui penyeimbang beban elastis atau host benteng (jika digunakan) di subnet publik, dan kemudian rute dengan tepat ke sumber daya subnet pribadi.

Konektivitas MongoDB Atlas melalui PrivateLink

- Arsitektur menggunakan PrivateLink (melalui titik akhir VPC) untuk terhubung dengan aman ke MongoDB Atlas tanpa mengekspos data Anda ke internet publik.
- Permintaan tetap ada di jaringan AWS backbone. Data dalam perjalanan mendapat manfaat dari PrivateLink enkripsi dan tidak pernah dialihkan melalui internet publik.
- VPC khusus MongoDB Atlas menghosting node primer dan sekunder Anda, dan menyediakan lingkungan yang aman dan terisolasi untuk cluster database terkelola Anda.

Deployment Multi-AZ

- Komponen infrastruktur penting (seperti gateway NAT dan subnet aplikasi) didistribusikan di setidaknya tiga Availability Zone. Jika Availability Zone mengalami pemadaman, arsitektur ini memastikan bahwa beban kerja di Availability Zone yang tersisa tetap beroperasi.

- MongoDB Atlas, secara default, menawarkan ketersediaan tinggi melalui set replika dan memastikan bahwa lapisan database Anda tetap toleran terhadap kesalahan. Infrastruktur kritis tersebar di setidaknya tiga Availability Zone untuk ketahanan.

Alat

Layanan AWS

- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API untuk mengambil rahasia secara terprogram.

Produk dan alat lainnya

- MongoDB Atlas adalah database yang dikelola sepenuhnya sebagai layanan (dBaaS) untuk menyebarkan dan mengelola database MongoDB di cloud.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal. Dalam pola ini, Anda menggunakan Terraform untuk menjalankan skrip untuk memfasilitasi penerapan sumber daya yang diperlukan di dan AWS MongoDB Atlas.

Repositori kode

Kode untuk pola ini tersedia di repositori AWS dan MongoDB Atlas Landing Zone. GitHub

Epik

Penemuan dan penilaian lengkap

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi pemangku kepentingan utama.	Identifikasi semua pemangku kepentingan utama dan anggota tim yang terlibat dalam proyek landing zone Anda. Ini dapat mencakup peran seperti:	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Administrator basis data () DBAs • DevOps insinyur • Pengembang aplikasi • Arsitek aplikasi	
Buat cetak biru struktural.	Buat cetak biru yang menguraikan struktur yang diinginkan dari landing zone berkemampuan Atlas Anda dan AWS MongoDB.	Pimpin migrasi
Buat rencana arsitektur.	Bekerja dengan arsitek aplikasi Anda untuk menganalisis persyaratan dan merancang arsitektur yang tahan terhadap kesalahan dan tangguh. Pola ini menyediakan template arsitektur starter untuk referensi Anda. Anda dapat menyesuaikan template ini untuk memenuhi kebutuhan keamanan dan infrastruktur organisasi Anda.	Arsitek awan
Rencanakan penyiapan dan penerapan.	Tentukan, dengan semua pemangku kepentingan, bagaimana arsitektur akan diterapkan, bagaimana langkah-langkah keamanan akan diterapkan, dan aspek lainnya untuk memastikan keselarasan dengan kepentingan organisasi dan tim yang meminta.	Pimpinan migrasi, DevOps insinyur, DBA

Siapkan lingkungan MongoDB Atlas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Kloning kode dari GitHub repositori dengan menjalankan perintah: `git clone https://github.com/mongodb-partners/AWS-MongoDB-Atlas-Landing-Zone`	Pengembang aplikasi, DevOps insinyur
Dapatkan ID organisasi Atlas Anda.	1. Jika Anda tidak memiliki akun MongoDB Atlas, daftar untuk satu. 2. Ikuti langkah-langkah dalam dokumentasi MongoDB untuk membuat organisasi. 3. Salin ID organisasi.	DBA
Hasilkan kunci API tingkat organisasi Atlas.	Untuk membuat kunci API tingkat organisasi Anda di Atlas, ikuti petunjuk dalam dokumentasi MongoDB.	DBA
Buat rahasia di AWS Secrets Manager.	Simpan kunci MongoDB Atlas API yang dihasilkan pada langkah sebelumnya sebagai rahasia nilai kunci di Secrets Manager. Untuk petunjuk, lihat dokumentasi Secrets Manager.	DevOps insinyur
Pilih tingkat cluster Atlas.	Untuk memilih tingkat cluster Atlas yang benar, ikuti	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	petunjuk dalam dokumentasi MongoDB.	

Mengatur AWS lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah skrip Terraform.	Di salinan GitHub repositor i lokal Anda, perbarui nama rahasia di modules/mongodb-atlas/mainfile.tf (baris 12), sehingga Terraform dapat mengambil kredensil dari Secrets Manager selama penerapan.	DevOps insinyur
Buat ID kunci AWS akses dan kunci rahasia.	Untuk membuat ID kunci AWS akses dan kunci rahasia, ikuti petunjuk di artikel AWS re:Post Bagaimana cara membuat kunci AWS akses? Ini adalah praktik terbaik untuk menetapkan kebijakan dengan hak istimewa paling sedikit yang diperlukan, tetapi untuk kasus ini, pilih kebijakan. `AdministratorAccess` Setelah Anda membuat kunci akses, tinjau praktik terbaik Keamanan di IAM untuk mempelajari praktik terbaik dalam mengelola kunci akses.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alokasikan alamat IP elastis.	Alokasikan setidaknya dua alamat IDs IP elastis. Untuk petunjuknya, lihat dokumentasi Amazon Virtual Private Cloud (Amazon VPC).	DevOps insinyur
Buat ember S3.	Buat bucket S3 untuk menyimpan status penerapan Terraform Anda dengan mengikuti petunjuk di dokumentasi Amazon Simple Storage Service (Amazon S3).	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui bucket S3 untuk penyimpanan.	Perbarui informasi bucket S3 di versi lokal environments/development/main.tf agar sesuai dengan nama dan Wilayah bucket yang Anda buat pada langkah sebelumnya, dan tentukan awalan key. Contoh: `terraform { ... backend "s3" { bucket = "startup-name-product-terraform" key = "network/dev" region = "ap-southeast-1" } }` Untuk contoh ini, Anda dapat mengonfigurasi Terraform untuk menggunakan `network/dev` awalan key untuk mengatur file status Terraform. Anda dapat mengubah nilai ke `prod` atau `staging` mencocokkan lingkungan yang ingin Anda buat. Untuk informasi tentang menggunakan beberapa lingkungan, lihat langkah terakhir di bagian ini.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkap nya tentang awalan kunci Amazon S3, lihat Mengatur objek menggunakan awalan dalam dokumentasi Amazon S3.	
Tetapkan variabel Terraform.	Sample landing zone mendefinisikan nilai variabel input dengan menggunakan file definisi variabel Terraform. File variabel terletak di environments/development/variables.tf. Anda dapat mengatur nilai variabel dalam environments/development/terraformfile.tfvars. Konfigura sikan variabel-variabel ini seperti yang dijelaskan dalam file Readme untuk GitHub repositori.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur variabel lingkungan.	Jika Anda berencana untuk menjalankan skrip Terraform di mesin lokal Anda, siapkan variabel lingkungan berikut: • `AWS_ACCESS_KEY_ID` : ID kunci AWS akses • `AWS_SECRET_ACCESS_KEY` : kunci akses AWS rahasia • `AWS_DEFAULT_REGION` : Wilayah AWS • `TF_LOG`: Tingkat log Terraform (`DEBUG`atau) `INFO` Untuk informasi selengkapnya tentang pengaturan variabel lingkungan, lihat dokumentasi AWS Command Line Interface (AWS CLI).	DevOps insinyur
Periksa konfigurasi VPC.	Untuk mengikuti praktik terbaik yang direkomen dasikan oleh AWS, konfigura sikan pengaturan untuk VPC dan subnet CIDRs, gateway NAT, rute, dan tabel rute dalam skrip Terraform untuk memenuhi kebutuhan organisasi Anda. Untuk spesifik, lihat file Readme untuk repositori. GitHub	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sumber daya tag.	Anda dapat menandai AWS sumber daya Anda untuk memantaunya saat digunakan oleh skrip Terraform. Sebagai contoh, lihat file Readme untuk GitHub repositori. Untuk informasi tentang pemantaua n sumber daya melalui tag untuk biaya, penggunaan, dan sebagainya, lihat Mengaktif kan tag alokasi biaya yang ditentukan pengguna dalam dokumentasi. AWS Billing	DevOps insinyur
Gunakan beberapa lingkunga n.	GitHub Repositori menyediak an folder `development` lingkungan. Anda juga dapat menambahkan lingkunga n Anda sendiri di folder lingkungan. Untuk menambahk an lingkungan, salin `development` folder ke folder baru (misalnya , `prod` atau`staging`) di bawah`environments` . Anda kemudian dapat memperbar ui `terraform.tfvars` file dengan nilai baru.	DevOps insinyur

Menyebarkan landing zone

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi direktori kerja Terraform.	Untuk menginisialisasi direktori kerja dan mengunduh paket yang diperlukan, jalankan perintah: `terraform init`	DevOps insinyur
Buat rencana eksekusi.	Untuk membuat rencana eksekusi dan memvisual isasikan perubahan yang akan dilakukan Terraform pada infrastruktur Anda, jalankan perintah: `terraform plan`	DevOps insinyur
Menyebarkan perubahan.	Untuk mengimplementasikan perubahan pada infrastruktur Anda seperti yang dijelaskan dalam kode, jalankan perintah: `terraform apply`	DevOps insinyur
Validasi penerapan.	Validasi komponen yang dibuat atau dimodifikasi Terraform di infrastruktur Anda. Untuk menguji penyiapan , sediakan sumber daya komputasi (misalnya, EC2 instans atau AWS Lambda	DevOps insinyur, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	fungsi Amazon) di dalam atau dilampirkan ke VPC.	

Hapus sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan.	Setelah Anda selesai menguji, jalankan perintah berikut untuk menghancurkan sumber daya yang diterapkan Terraform di infrastruktur Anda: `terraform destroy`	DevOps insinyur

Sumber daya terkait

Penemuan dan penilaian

- Kiat administratif untuk pengaturan landing zone (AWS Control Tower dokumentasi)
- Harapan untuk konfigurasi landing zone (AWS Control Tower dokumentasi)
- Praktik terbaik untuk pembaruan landing zone (AWS Control Tower dokumentasi)

Menyiapkan MongoDB Atlas dan lingkungan AWS

- Mendapatkan MongoDB Atlas ()AWS Marketplace
- Memori (dokumentasi MongoDB Atlas)
- Contoh ukuran dengan kumpulan data sampel Atlas (dokumentasi MongoDB Atlas)
- Contoh ukuran untuk aplikasi seluler (dokumentasi MongoDB Atlas)
- Lalu lintas jaringan (dokumentasi MongoDB Atlas)
- Cluster auto-scaling (dokumentasi MongoDB Atlas)
- Template ukuran Atlas (dokumentasi MongoDB Atlas)

- Siapkan koneksi peering jaringan (dokumentasi MongoDB Atlas)
- Titik akhir pribadi di Atlas (dokumentasi MongoDB Atlas)
- Enkripsi tingkat bidang sisi klien (dokumentasi basis data MongoDB)
- Enkripsi otomatis (dokumentasi database MongoDB)
- Memilih tingkat cluster (dokumentasi MongoDB Atlas)

Menyebarkan landing zone

- Terraform aktif AWS(CI/CD untuk Jaringan 5G di whitepaper) AWS
- MongoDB Atlas dengan Terraform (dokumentasi MongoDB)

Konfigurasikan Log Aliran VPC untuk sentralisasi di seluruh Akun AWS

Benjamin Morris dan Aman Kaur Gandhi, Amazon Web Services

Ringkasan

Dalam AWS virtual private cloud (VPC), fitur VPC Flow Logs dapat menyediakan data yang berguna untuk pemecahan masalah operasional dan keamanan. Namun, ada batasan dalam menggunakan VPC Flow Logs di lingkungan multi-akun. Secara khusus, log aliran lintas akun dari Amazon CloudWatch Logs tidak didukung. Sebagai gantinya, Anda dapat memusatkan log dengan mengonfigurasi bucket Amazon Simple Storage Service (Amazon S3) dengan kebijakan bucket yang sesuai.

> **Note**
>
> Pola ini membahas persyaratan untuk mengirim log aliran ke lokasi terpusat. Namun, jika Anda juga ingin log tersedia secara lokal di akun anggota, Anda dapat membuat beberapa log aliran untuk setiap VPC. Pengguna yang tidak memiliki akses ke akun Arsip Log dapat melihat log lalu lintas untuk pemecahan masalah. Atau, Anda dapat mengonfigurasi log aliran tunggal untuk setiap VPC yang mengirimkan log ke CloudWatch Log. Anda kemudian dapat menggunakan filter langganan Amazon Data Firehose untuk meneruskan log ke bucket S3. Untuk informasi selengkapnya, lihat bagian Sumber daya terkait.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Organizations Organisasi dengan akun yang digunakan untuk memusatkan log (misalnya, Arsip Log)

Keterbatasan

Jika Anda menggunakan kunci terkelola AWS Key Management Service (AWS KMS) `aws/s3` untuk mengenkripsi bucket pusat, bucket pusat tidak akan menerima log dari akun lain. Sebagai gantinya, Anda akan melihat kode `Unsuccessful` kesalahan 400 dengan pesan seperti `"LogDestination:`

`<bucketName> is undeliverable"` untuk yang Anda berikan`ResourceId`. Ini karena kunci AWS terkelola akun tidak dapat dibagikan di seluruh akun. Solusinya adalah dengan menggunakan enkripsi terkelola Amazon S3 (SSE-S3) atau kunci yang dikelola AWS KMS pelanggan yang dapat Anda bagikan dengan akun anggota.

Arsitektur

Arsitektur target

Dalam diagram berikut, dua log aliran digunakan untuk setiap VPC. Satu mengirim log ke grup CloudWatch Log lokal. Yang lain mengirim log ke bucket S3 di akun logging terpusat. Kebijakan bucket mengizinkan layanan pengiriman log untuk menulis log ke bucket.

> **Note**
>
> Mulai November 2023, AWS sekarang mendukung aws: SourceOrg ID condition key. Kondisi ini memungkinkan Anda untuk menolak menulis ke bucket terpusat untuk akun di luar AWS Organizations organisasi Anda.

Otomatisasi dan skala

Setiap VPC dikonfigurasi untuk mengirim log ke bucket S3 di akun logging pusat. Gunakan salah satu solusi otomatisasi berikut untuk membantu memastikan bahwa log aliran dikonfigurasi dengan tepat:

- CloudFormation StackSets
- AWS Control Tower Account Factory untuk Terraform (AFT)
- AWS Config Aturan dengan remediasi

Alat

Alat

- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS Pola ini menggunakan fitur VPC Flow Logs untuk menangkap informasi tentang lalu lintas IP yang menuju dan dari antarmuka jaringan di VPC Anda.

Praktik terbaik

Menggunakan infrastruktur sebagai kode (IAc) dapat sangat menyederhanakan proses penyebaran VPC Flow Logs. Mengabstraksi definisi penerapan VPC Anda untuk menyertakan konstruksi sumber daya log aliran akan menerapkan VPCs log aliran Anda secara otomatis. Ini ditunjukkan di bagian selanjutnya.

Log aliran terpusat

Contoh sintaks untuk menambahkan log aliran terpusat ke modul VPC di HashiCorp Terraform: Kode ini membuat log aliran yang mengirim log dari VPC ke bucket S3 terpusat. Perhatikan bahwa pola ini tidak mencakup pembuatan bucket S3. Untuk pernyataan kebijakan bucket yang direkomendasikan, lihat bagian Informasi tambahan.

```
variable "vpc_id" { type = string }
locals { custom_log_format_v5 = "$${version} $${account-id} $${interface-id} $
${srcaddr} $${dstaddr} $${srcport} $${dstport} $${protocol} $${packets} $${bytes}
 $${start} $${end} $${action} $${log-status} $${vpc-id} $${subnet-id} $${instance-
id} $${tcp-flags} $${type} $${pkt-srcaddr} $${pkt-dstaddr} $${region} $${az-id} $
${sublocation-type} $${sublocation-id} $${pkt-src-aws-service} $${pkt-dst-aws-service}
 $${flow-direction} $${traffic-path}" }
resource "aws_flow_log" "centralized_flow_log" {
  log_destination      = "arn:aws:s3:::centralized-vpc-flow-logs-
<log_archive_account_id>" # Optionally, a prefix can be added after the ARN.
  log_destination_type = "s3"
  traffic_type         = "ALL"
  vpc_id               = var.vpc_id
  log_format           = local.custom_log_format_v5 # If you want fields from VPC Flow
 Logs v3+, you will need to create a custom log format.
}
```

Untuk informasi selengkapnya tentang format log kustom, lihat dokumentasi Amazon VPC.

Log aliran lokal

Contoh sintaks untuk menambahkan log aliran lokal ke modul VPC di Terraform dengan izin yang diperlukan: Kode ini membuat log aliran yang mengirimkan log dari VPC ke grup Log lokal. CloudWatch

```
data "aws_region" "current" {}
variable "vpc_id" { type = string }
resource "aws_iam_role" "local_flow_log_role" {
  name = "flow-logs-policy-${var.vpc_id}"
  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Principal": {"Service": "vpc-flow-logs.amazonaws.com"},
      "Action": "sts:AssumeRole"
  }]
}
EOF
}
resource "aws_iam_role_policy" "logs_permissions" {
  name = "flow-logs-policy-${var.vpc_id}"
  role = aws_iam_role.local_flow_log_role.id
  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [{
      "Action": ["logs:CreateLog*", "logs:PutLogEvents", "logs:DescribeLog*",
 "logs:DeleteLogDelivery"],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:${data.aws_region.current.name}:*:log-group:vpc-flow-
logs*"
  }]
}
EOF
}
resource "aws_cloudwatch_log_group" "local_flow_logs" {
  name              = "vpc-flow-logs/${var.vpc_id}"
  retention_in_days = 30
}
resource "aws_flow_log" "local_flow_log" {
  iam_role_arn    = aws_iam_role.local_flow_log_role.arn
  log_destination = aws_cloudwatch_log_group.local_flow_logs.arn
  traffic_type    = "ALL"
```

```
  vpc_id          = var.vpc_id
}
```

Epik

Menyebarkan infrastruktur VPC Flow Logs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan strategi enkripsi dan buat kebijakan untuk bucket S3 pusat.	Bucket pusat tidak mendukung AWS KMS `aws/s3` kunci, jadi Anda harus menggunakan SSE-S3 atau kunci yang dikelola AWS KMS pelanggan. Jika Anda menggunakan AWS KMS kunci, kebijakan kunci harus mengizinkan akun anggota untuk menggunakan kunci tersebut.	Kepatuhan
Buat bucket log aliran pusat.	Buat bucket pusat tempat flow log akan dikirim, dan terapkan strategi enkripsi yang Anda pilih pada langkah sebelumnya. Ini harus dalam Arsip Log atau akun yang bertujuan serupa. Dapatkan kebijakan bucket dari bagian Informasi tambahan, dan terapkan ke bucket pusat setelah memperbarui placeholder dengan nilai spesifik lingkungan Anda.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Log Aliran VPC untuk mengirim log ke bucket log aliran pusat.	Tambahkan log aliran ke setiap VPC yang ingin Anda kumpulkan datanya. Cara yang paling skalabel untuk melakukan ini adalah dengan menggunakan alat IAc seperti AFT atau. AWS Cloud Development Kit (AWS CDK) Misalnya, Anda dapat membuat modul Terraform yang menerapkan VPC di samping log aliran. Jika perlu, Anda menambahkan log aliran secara manual.	Administrator jaringan
Konfigurasikan Log Aliran VPC untuk dikirim ke Log lokal CloudWatch .	(Opsional) Jika Anda ingin log aliran terlihat di akun tempat log dibuat, buat log aliran lain untuk mengirim data ke CloudWatch Log di akun lokal. Atau, Anda dapat mengirim data ke bucket S3 khusus akun di akun lokal.	AWS Umum

Sumber daya terkait

- Cara Memfasilitasi Analisis Data dan Memenuhi Persyaratan Keamanan dengan Menggunakan Data Log Aliran Terpusat (posting AWS blog)
- Cara mengaktifkan VPC Flow Logs secara otomatis menggunakan AWS Config aturan (AWS posting blog)

Informasi tambahan

Kebijakan bucket

Contoh kebijakan bucket ini dapat diterapkan ke bucket S3 pusat untuk log aliran, setelah Anda menambahkan nilai untuk nama placeholder.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<BUCKET_NAME>/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceOrgID": "<ORG_ID>"
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::<BUCKET_NAME>",
            "Condition": {
                "StringEquals": {
                    "aws:SourceOrgID": "<ORG_ID>"
                }
            }
        },
        {
            "Sid": "DenyUnencryptedTraffic",
            "Effect": "Deny",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:*",
            "Resource": [
```

```
                "arn:aws:s3:::<BUCKET_NAME>/*",
                "arn:aws:s3:::<BUCKET_NAME>"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

Kelola set AWS izin secara dinamis dengan menggunakan Terraform

Vinicius Elias dan Marcos Vinicius Pinto Jordao, Amazon Web Services

Ringkasan

AWS IAM Identity Center meningkatkan AWS Identity and Access Management (IAM) dengan menyediakan hub terpusat untuk mengelola akses masuk tunggal ke dan aplikasi cloud. Akun AWS Namun, manajemen manual set izin Pusat Identitas IAM dapat menjadi semakin kompleks dan rawan kesalahan seiring pertumbuhan organisasi Anda. Kompleksitas ini dapat menyebabkan potensi kesenjangan keamanan dan overhead administratif.

Solusi ini memungkinkan Anda untuk mengelola set izin melalui infrastruktur sebagai kode (IAc) menggunakan pipeline continuous integration and continuous delivery (CI/CD) yang dibangun dengan native. Layanan AWS Ini memungkinkan integrasi tanpa batas dari mekanisme penetapan set izin dengan peristiwa AWS Control Tower siklus hidup atau lingkungan Account Factory for Terraform (AFT). Pendekatan ini menyediakan konfigurasi identitas dinamis untuk yang baru dan yang sudah ada Akun AWS.

EventBridge Aturan Amazon memantau Akun AWS pembuatan dan pembaruan, yang membantu konfigurasi identitas Anda tetap disinkronkan dengan struktur organisasi Anda. Setelah membuat atau memperbarui akun di AWS Control Tower atau AFT, pipeline dipicu. Ini mengevaluasi satu set file JSON dengan definisi set izin dan aturan penetapan. Kemudian pipeline berlaku dan menyinkronkan pengaturan di semua akun.

Pendekatan ini memberikan manfaat sebagai berikut:

- Konsistensi - Menghilangkan penyimpangan konfigurasi manual di seluruh organisasi Anda AWS
- Auditabilitas - Mempertahankan riwayat lengkap dari semua perubahan manajemen identitas
- Skalabilitas - Secara otomatis menerapkan konfigurasi saat lingkungan Anda tumbuh AWS
- Keamanan - Mengurangi kesalahan manusia dalam penugasan izin
- Kepatuhan — Memfasilitasi memenuhi persyaratan peraturan melalui perubahan yang terdokumentasi dan aturan penugasan

Prasyarat dan batasan

- Lingkungan multi-akun dengan AWS Control Tower dan AWS Organizations pengaturan. Secara opsional, Anda dapat menggunakan AFT dengan AWS Control Tower.
- Administrator yang didelegasikan Pusat Identitas IAM Akun AWS untuk menerima solusinya. Untuk informasi selengkapnya, lihat Administrasi yang didelegasikan dalam dokumentasi Pusat Identitas IAM.
- Sebuah sistem kontrol versi (VCS) repositori untuk menangani kode utama. Untuk sampel, lihat GitHub repositori solusi.
- AWS Sumber daya yang diperlukan untuk manajemen backend Terraform, seperti bucket Amazon Simple Storage Service (Amazon S3) dan tabel Amazon DynamoDB.

Batasan

- Pipeline menggunakan sumber daya AWS asli dan Terraform open source. Pipa tidak siap untuk melakukan panggilan ke ekosistem pihak ketiga.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan komponen dan alur kerja untuk pola ini.

AWS Control Tower aliran peristiwa

Solusinya dimulai dengan integrasi peristiwa yang berasal dari salah satu AWS Control Tower atau AFT. Pilihan antara satu atau layanan lainnya dibuat pada waktu implementasi melalui definisi variabel. Terlepas dari metode yang digunakan, pipeline dipicu setiap kali akun dibuat atau diperbarui. Pipeline merekonsiliasi kebijakan yang disimpan dalam repositori manajemen set izin.

Berikut ini adalah peristiwa AWS Control Tower siklus hidup:

- `CreateManagedAccount`— Saat akun baru dibuat
- `UpdateManagedAccount`— Saat akun yang ada diperbarui

Perutean acara

EventBridge berfungsi sebagai layanan pemrosesan acara pusat, menangkap peristiwa yang dihasilkan di AWS Control Tower akun. Ketika peristiwa terjadi, EventBridge secara cerdas mengarahkannya ke bus acara terpusat di akun solusi. AWS Control Tower peristiwa siklus hidup mengikuti pola perutean yang berbeda. Jika AFT didefinisikan sebagai sumber peristiwa, akun manajemen AFT menangani peristiwa alih-alih AWS Control Tower akun. Arsitektur berbasis peristiwa ini memungkinkan respons otomatis terhadap perubahan organisasi tanpa intervensi manual.

Proses integrasi AFT

Ketika peristiwa AWS Control Tower siklus hidup mencapai akun manajemen AFT, peristiwa tersebut secara otomatis memicu beberapa proses hilir yang intrinsik ke AFT. Setelah alur kerja kustomisasi akun AFT selesai, ia menerbitkan pesan ke topik Simple Notification Service `aft-notifications` Amazon (Amazon SNS) khusus. Topik itu memicu `aft-new-account-forward-event` AWS Lambda fungsi yang diterapkan oleh solusi ini. Fungsi Lambda mengirimkan acara ke bus acara akun solusi, di mana ia digunakan untuk memulai pipeline.

Infrastruktur sebagai pipa kode

Pipa solusi beroperasi sebagai mekanisme penyebaran yang sepenuhnya otomatis. AWS CodePipeline Layanan terus memantau repositori untuk perubahan. Setelah mendeteksi komit baru, secara otomatis memulai alur kerja penerapan dan memulai proses sekuensial yang mencakup fase validasi dan eksekusi. Sistem menjalankan `plan` operasi Terraform untuk mengidentifikasi perubahan yang diusulkan, diikuti oleh `apply` perintah Terraform untuk mengimplementasikan perubahan tersebut di lingkungan. AWS Khususnya, pipa berjalan tanpa gerbang persetujuan manual. Pendekatan ini memungkinkan penyebaran cepat perubahan infrastruktur sambil mempertahankan auditabilitas melalui log pipa dan file status Terraform.

Pipeline memanfaatkan AWS CodeBuild untuk menjalankan operasi Terraform di lingkungan yang terkendali dengan izin yang sesuai. Melalui pendekatan IAc ini, pipeline dapat melakukan operasi manajemen izin yang komprehensif termasuk:

- Buat set izin baru.
- Perbarui set izin yang ada.
- Hapus set izin yang tidak perlu.
- Kelola penetapan izin ini di seluruh akun dan grup dalam organisasi. AWS

Untuk menjaga konsistensi infrastruktur dan mencegah perubahan yang bertentangan, solusinya mengimplementasikan sistem manajemen status backend Terraform menggunakan bucket Amazon S3 dan tabel Amazon DynamoDB khusus. Pendekatan ini menyediakan lokasi penyimpanan persisten untuk file status Terraform dan mekanisme penguncian status untuk mencegah modifikasi bersamaan pada sumber daya yang sama.

Kode Terraform utama menggunakan modul AWS `permission-sets` Terraform resmi. Modul ini dapat secara dinamis mengelola set izin di IAM Identity Center, berdasarkan template set izin.

Manajemen kontrol sumber

Templat set izin (file JSON) berada di sistem kontrol versi eksternal, seperti GitHub, yang menyediakan repositori terpusat untuk konfigurasi manajemen identitas. Pendekatan ini menetapkan satu sumber kebenaran untuk definisi set izin, sambil memungkinkan pengembangan kolaboratif melalui praktik tinjauan kode standar. Pengguna yang berwenang dapat melakukan perubahan pada templat ini mengikuti proses manajemen perubahan organisasi. Komit ini berfungsi sebagai pemicu utama untuk pipeline penerapan otomatis, memulai proses pembaruan infrastruktur.

Untuk contoh cara mengonfigurasi set izin menggunakan file JSON di repositori, lihat Informasi tambahan.

Alat

Layanan AWS

- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeConnectionsmemungkinkan AWS sumber daya dan layanan, seperti CodePipeline, untuk terhubung ke repositori kode eksternal, seperti. GitHub
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.

- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua aplikasi Anda Akun AWS dan cloud.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email. Ini memungkinkan pemberitahuan push untuk acara manajemen akun, memastikan bahwa pihak terkait diberitahu tentang perubahan atau tindakan penting dalam sistem.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di organisasi AWS Sampel di dalam GitHub repositori sample-terraform-aws-permission-sets-pipeline.

Praktik terbaik

- Selalu sematkan versi modul dan penyedia Terraform yang digunakan untuk menjalankan kode dalam produksi.

- Gunakan alat analisis kode statis, seperti Checkov, untuk memindai kode Anda dan kemudian memecahkan masalah keamanan.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Buat prasyarat (opsional)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sumber daya backend Terraform.	Jika Anda belum membuat AWS sumber daya backend Terraform, gunakan langkah-langkah berikut untuk membuat bucket () Amazon S3 `s3-tf-backend-{ACCOUNT_ID}` dan tabel DynamoDB (). `ddb-tf-backend` 1. Masuk ke Akun AWS tempat Anda akan menggunakan solusi, dan buka AWS CloudShelldi AWS Control Tower rumah Wilayah AWS. 2. Jalankan perintah berikut, ganti placeholder `{ACCOUNT_ID}` dengan ID Anda Akun AWS : `aws s3api create-bucket --bucket s3-tf-backend-{ACCOUNT_ID}`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws s3api put-bucket-versioning --bucket s3-tf-backend-{ACCOUNT_ID} --versioning-configuration Status=Enabled` `aws dynamodb create-table --table-name ddb-tf-backend --attribute-definitions AttributeName=LockID,AttributeType=S --key-schema AttributeName=LockID,KeyType=HASH --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran lintas akun.	Anda harus memberikan peran IAM lintas akun dalam konfigurasi penyedia `event-source-account` Terraform AWS . Jika Anda belum membuat peran ini, gunakan langkah-langkah berikut untuk membuatnya: 1. Masuk ke Akun AWS yang akan menjadi sumber acara Anda (akun AWS Control Tower manajemen atau akun AFT), dan buka AWS CloudShell. 2. Jalankan perintah berikut, ganti placeholder `{ACCOUNT_ID}` dengan Akun AWS ID yang Anda gunakan untuk solusi ini, bukan ID Anda saat ini Akun AWS : `aws iam create-role \ --role-name CrossAccountRole \ --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": {`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"AWS": "arn:aws:iam::{ACCOUNT_ID}:root"
	},
	
	"Action": "sts:AssumeRole"
	}
	]
	}'</pre>
	1. Periksa output perintah dan salin peran Amazon Resource Name (ARN) (misalnya,`arn:aws:iam::111122223333:role/CrossAccountRole`).
	2. Untuk melampirkan kebijakan IAM ke peran, jalankan perintah berikut:
	
	<pre>aws iam attach-role-policy \
	--role-name CrossAccountRole \
	--policy-arn arn:aws:iam::aws:policy/AdministratorAccess</pre>
	Contoh ini menggunakan kebijakan AdministratorAccessIAM AWS terkelola. Jika mau, Anda dapat menggunak	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an kebijakan yang lebih spesifik.	

Siapkan repositori set izin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori khusus.	Tugas ini mengasumsikan bahwa Anda menggunak an GitHub. Buat repositor i khusus untuk menyimpan kode Terraform utama dan file JSON template set izin.	DevOps insinyur
Siapkan kode set izin.	Untuk informasi tentang bagaimana Anda dapat menyusun file-file berikut, lihat contoh kode pada repositori solusi: — utama.tf — keluaran.tf — penyedia.jinja — templat Salin konten, simpan `providers.jinja` nilainya, dan buat penyesuaian yang diperlukan untuk file lain. Misalnya, tambahkan izin set file template ke dalam `templates` atau pin versi `aws-ia/permission-`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`sets/aws` modul dalam `main.tf` file.	
Komit perubahan Anda.	Komit dan dorong perubahan ke repositori yang Anda buat sebelumnya. Simpan nama repositori dan GitHub organisasinya, misalnya,. `myorg/aws-ps-pipeline`	DevOps insinyur

Siapkan kode penerapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kontennya.	Unduh (kloning) konten dari repositori solusi.	DevOps insinyur
Memenuhi variabel.	Buat `terraform.tfvars` file dan tambahkan variabel yang diperlukan berikut: • `repository_name` — Nama izin set repositori yang Anda buat sebelumnya • `branch_name` — Nama cabang repositori set izin • `vcs_provider` — Penyedia VCS Anda • `account_lifecycle_events_source` — Sumber peristiwa untuk memicu pipa (AWS Control Tower atau AFT)	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`repository_name = "myorg/aws-ps-pipeline"`
	`branch_name = "main"`
	`vcs_provider = "github"`
	`account_lifecycle_events_source = "CT"`
	
	Untuk informasi tentang opsi variabel tambahan, lihat file variabel.tf di repositori pola ini. GitHub	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan konfigurasi backend Terraform.	Dalam `backend.tf` file, ganti placeholder dengan nilai Anda sendiri. Gunakan AWS Control Tower home Wilayah AWS, dan berikan nama-nama bucket Amazon S3 dan tabel DynamoDB yang dibuat sebelumnya. `terraform { required_version = ">=1.6" backend "s3" { region = "{region}" bucket = "{bucket_name}" key = "terraform.tfstate" dynamodb_table = "{table_name}" encrypt = "true" } }` Jika mau, Anda dapat menggunakan konfigurasi backend Terraform Anda sendiri.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Sesuaikan konfigurasi penyedia Terraform.	Dalam `providers.tf` file, ganti placeholder dengan informasi Anda sendiri. Gunakan Wilayah AWS Control Tower asal, dan berikan ARN dari peran IAM lintas akun yang dibuat sebelumnya untuk penyedia. `event-source-account`
	`provider "aws" {`
	`  region = "{region}"`
	`}`
	
	`provider "aws" {`
	`  alias  = "event-so`
	`urce-account"`
	`  region = "{region}"`
	`  assume_role {`
	`  role_arn = "{role_ar`
	`n}"`
	`  }`
	`}`
		DevOps insinyur

Terapkan solusi secara manual

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih Akun AWS.	Kami menyarankan Anda menerapkan solusi di akun administrator yang didelegas ikan IAM Identity Center. Namun, Anda juga dapat menerapkannya di akun AWS Organizations manajemen.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk masuk ke akun yang dipilih di Wilayah yang sama dengan instans Pusat Identitas IAM, gunakan. AWS CLI Pastikan bahwa peran IAM yang Anda gunakan memiliki izin untuk mengambil peran yang ditentukan untuk `event-source-account` penyedia di langkah sebelumnya. Selain itu, peran ini harus memiliki akses ke AWS sumber daya yang digunakan dalam konfigurasi backend Terraform.	
Jalankan Terraform secara manual.	Untuk menginisialisasi, merencanakan, dan menerapkan konfigurasi, jalankan perintah Terraform berikut dalam urutan yang ditunjukkan: 1. `terraform init` 2. `terraform plan` 3. `terraform apply`	DevOps insinyur
Periksa hasil penerapan.	Di akun administrator yang didelegasikan IAM Identity Center, periksa apakah `aws-ps-pipeline` pipeline telah dibuat. Periksa juga apakah ada AWS CodeConnections koneksi dengan status Tertunda.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Selesaikan CodeConnections konfigurasi.	Untuk menyelesaikan CodeConnections konfigura si, gunakan langkah-langkah berikut: 1. Buka CodeConnections, pilih koneksi yang dibuat, dan pilih Perbarui koneksi yang tertunda. 2. Berikan GitHub kredensil Anda, instal GitHub Aplikasi baru, atau pilih yang sudah ada yang memiliki akses ke repositori set izin, dan pilih Connect. Pipeline sekarang harus memiliki akses ke repositori set izin. Untuk petunjuk mendetail, lihat Memperbarui sambungan yang tertunda di dokumentasi konsol Alat Pengembang.	AWS DevOps

Pilih alur eksekusi pipeline untuk menguji solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan pipeline oleh AWS Control Tower atau pembaruan AFT.	Setelah akun dibuat atau diubah dengan menggunak an AWS Control Tower atau AFT (tergantung pada jenis	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	peristiwa siklus hidup yang Anda pilih), pipeline dimulai.	
Jalankan pipeline dengan mengubah kode.	Setelah Anda mengubah kode dan mengkomitmennya ke `main` cabang, pipeline dimulai.	AWS DevOps
Jalankan pipa secara manual.	Untuk memulai pipeline secara manual, gunakan fitur Rilis perubahan di AWS CodePipeline.	AWS DevOps

Pemecahan masalah

Isu	Solusi
Akses ditolak	Verifikasi bahwa Anda memiliki izin yang diperlukan untuk menerapkan solusi.
CodeConnections masalah	• Periksa apakah status koneksi Tersedia daripada Tertunda. • Verifikasi bahwa CodeConnections konfigurasi telah dilengkapi dengan GitHub kredensil dengan benar. Untuk informasi selengkapnya, lihat Memperbarui sambungan yang tertunda di dokumentasi konsol Alat Pengembang. • Pastikan GitHub aplikasi memiliki izin akses yang tepat ke repositori set izin.
Masalah eksekusi pipa	• Periksa CloudWatch log Amazon untuk mengetahui kesalahan eksekusi pipeline. • Pastikan delegasi IAM Identity Center diatur dengan benar. Jika tidak, itu dapat

Isu	Solusi
	menyebabkan kegagalan pipa saat membaca Akun AWS metadata.
Izin menetapkan masalah penerapan	• Validasi bahwa sintaks JSON dalam file template set izin sudah benar. • Verifikasi bahwa kebijakan terkelola IAM yang direferensikan dalam templat ada. • Periksa apakah grup dan pengguna yang ditentukan dalam tugas ada di Pusat Identitas IAM.

Sumber daya terkait

Layanan AWS dokumentasi

- AWS IAM Identity Center Panduan Pengguna
- Kelola Akun AWS dengan set izin (dokumentasi Pusat Identitas IAM)

Sumber daya lainnya

- AWS Modul Set Izin (Terraform)

Informasi tambahan

File JSON dengan set izin sampel

Contoh berikut menunjukkan cara mengkonfigurasi set izin dengan menggunakan file JSON di repositori:

```
{
  "Name": "ps-billing", // Permission set identifier
  "Comment": "Sample permission set for billing access", // Comment to document the
 purpose of the permission set
  "Description": "Billing access in AWS", // Detailed description
  "SessionDuration": "PT4H", // Session duration = 4 hours (ISO 8601 format)
  "ManagedPolicies": [ // List of AWS IAM managed policies
```

```
    "arn:aws:iam::aws:policy/job-function/Billing",
    "arn:aws:iam::aws:policy/job-function/SupportUser",
    "arn:aws:iam::aws:policy/AWSSupportAccess",
    "arn:aws:iam::aws:policy/job-function/ViewOnlyAccess"
  ],
  "CustomerPolicies": [], // References to IAM policies previously created
  "CustomPolicy": {}, // Inline IAM policy defined directly in the permission set
  "PermissionBoundary": {  // AWS or customer managed IAM policy to be used as boundary
    "ManagedPolicy": "",
    "CustomerPolicy": ""
  },
  "Assignments": [ // Define the assignment rules
    {
      "all_accounts": true, // Apply to ALL active AWS accounts in organization
      "principal": "G_BILLING_USERS", // Group/user name in Identity Center
      "type": "GROUP", // Can be "GROUP" or "USER"
      "account_id": [], // List of AWS account ID (empty since all_accounts=true)
      "account_ou": [], // List of AWS Organizational Unit IDs with target AWS accounts
      "account_tag": [] // List of tags (key:value) to match AWS Organization accounts
 tags
    }
  ]
}
```

Untuk informasi selengkapnya, lihat skema JSON dalam dokumentasi modul Set AWS Izin di situs web Terraform.

Kiat

- Anda dapat menggunakan blok impor Terraform untuk mengimpor izin yang ada yang disetel ke solusi.
- Anda dapat menggunakan AFT untuk mengimplementasikan pipeline set AWS izin di akun yang didelegasikan. Untuk informasi lebih lanjut, lihat AFT Blueprints.

Tandai lampiran Transit Gateway secara otomatis menggunakan AWS Organizations

Richard Milner-Watts, Haris Bin Ayub, dan John Capps, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS), Anda dapat menggunakannya AWS Resource Access Manageruntuk berbagi AWS Transit Gatewaylintas Akun AWS batas. Namun, saat Anda membuat lampiran Transit Gateway melintasi batas akun, lampiran dibuat tanpa tag Nama. Itu bisa membuat identifikasi lampiran memakan waktu.

Solusi ini menyediakan mekanisme otomatis untuk mengumpulkan informasi tentang setiap lampiran Transit Gateway untuk akun dalam organisasi yang dikelola oleh AWS Organizations. Prosesnya termasuk mencari rentang Classless Inter-Domain Routing (CIDR) dari tabel rute Transit Gateway. Solusinya kemudian menerapkan tag Nama dalam bentuk `<CIDR-range>-<AccountName>` lampiran di dalam akun yang memegang gateway transit.

Solusi ini dapat digunakan bersama solusi seperti Orkestrator Jaringan Transit Tanpa Server dari Perpustakaan Solusi. AWS Orkestrator Jaringan Transit Tanpa Server memungkinkan pembuatan otomatis lampiran Transit Gateway Gateway dalam skala besar.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Organizations Organisasi yang berisi semua akun terkait
- Akses ke akun manajemen organisasi, di bawah akar organisasi, untuk membuat peran yang diperlukan AWS Identity and Access Management (IAM)
- Akun anggota Jaringan Bersama yang berisi satu atau beberapa gateway transit yang dibagikan dengan organisasi dan memiliki lampiran

Arsitektur

Tangkapan layar berikut Konsol Manajemen AWS menunjukkan contoh lampiran Transit Gateway tanpa tag Nama terkait dan dua lampiran Transit Gateway dengan tag Nama yang dihasilkan oleh solusi ini. Struktur tag Nama yang dihasilkan adalah`<CIDR-range>-<AccountName>`.

Solusi ini digunakan AWS CloudFormationuntuk menyebarkan AWS Step Functionsalur kerja yang mengelola pembuatan tag Nama Transit Gateway di semua konfigurasi. Wilayah AWS Alur kerja memanggil AWS Lambdafungsi, yang melakukan tugas yang mendasarinya.

Setelah solusi memperoleh nama akun dari AWS Organizations, mesin status Step Functions mendapatkan semua lampiran Transit Gateway IDs. Ini diproses secara paralel oleh Wilayah. Pemrosesan ini termasuk mencari rentang CIDR untuk setiap lampiran. Rentang CIDR diperoleh dengan mencari tabel rute Transit Gateway di dalam Wilayah untuk ID lampiran Transit Gateway yang cocok. Jika semua informasi yang diperlukan tersedia, solusinya akan menerapkan tag Nama ke lampiran. Solusinya tidak akan menimpa tag Nama yang ada.

Solusinya berjalan pada jadwal yang dikendalikan oleh EventBridge acara Amazon. Acara ini memulai solusi setiap hari pada pukul 6:00 UTC.

Tumpukan teknologi target

- Amazon EventBridge
- AWS Lambda
- AWS Organizations
- AWS Transit Gateway
- Amazon Virtual Private Cloud (Amazon VPC)
- AWS X-Ray

Arsitektur target

Arsitektur solusi dan alur kerja ditunjukkan pada diagram berikut.

1. Acara yang dijadwalkan memulai aturan.
2. EventBridge Aturan memulai mesin status Step Functions.
3. Mesin status memanggil fungsi `tgw-tagger-organizations-account-query` Lambda.
4. Fungsi `tgw-tagger-organizations-account-query` Lambda mengasumsikan peran dalam akun manajemen organisasi.
5. Fungsi `tgw-tagger-organizations-account-query` Lambda memanggil Organizations API untuk mengembalikan Akun AWS metadata.

6. Mesin status memanggil fungsi `tgw-tagger-attachment-query` Lambda.
7. Untuk setiap Wilayah, secara paralel, mesin status memanggil fungsi `tgw-tagger-rtb-query` Lambda untuk membaca rentang CIDR untuk setiap lampiran.
8. Untuk setiap Wilayah, secara paralel, mesin status memanggil fungsi `tgw-tagger-attachment-tagger` Lambda.
9. Tag nama dibuat untuk lampiran Transit Gateway di akun Jaringan Bersama.

Otomatisasi dan skala

Solusi memproses setiap Wilayah secara paralel untuk mengurangi total durasi proses.

Alat

Layanan AWS

- AWS CloudFormationmenyediakan cara untuk memodelkan kumpulan sumber daya terkait AWS dan pihak ketiga, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidup mereka, dengan memperlakukan infrastruktur sebagai kode.
- Amazon CloudWatch membantu Anda memantau metrik sumber daya AWS dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge menerima peristiwa, indikator perubahan lingkungan, dan menerapkan aturan untuk merutekan acara ke target. Aturan mencocokkan peristiwa dengan target berdasarkan struktur peristiwa, yang disebut pola peristiwa, atau sesuai jadwal.
- AWS Lambdaadalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya bila diperlukan dan menskalakan secara otomatis, dari beberapa permintaan setiap hari hingga ribuan setiap detik. Anda hanya membayar untuk waktu komputasi yang Anda konsumsi. Tidak ada biaya ketika kode Anda tidak berjalan.
- AWS Organizationsmembantu Anda mengelola dan mengatur lingkungan Anda secara terpusat saat Anda tumbuh dan meningkatkan sumber daya Anda AWS . Dengan menggunakan Organizations, Anda dapat membuat sumber daya baru Akun AWS dan mengalokasikan sumber daya secara terprogram, mengelompokkan akun untuk mengatur alur kerja, menerapkan kebijakan ke akun atau grup untuk tata kelola, dan menyederhanakan penagihan dengan menggunakan satu metode pembayaran untuk semua akun Anda.

- AWS Step Functionsadalah layanan alur kerja visual kode rendah yang digunakan untuk mengatur, mengotomatiskan proses bisnis Layanan AWS, dan membangun aplikasi tanpa server. Alur kerja mengelola kegagalan, percobaan ulang, paralelisasi, integrasi layanan, dan observabilitas sehingga pengembang dapat fokus pada logika bisnis bernilai lebih tinggi.
- AWS Transit Gatewaymenghubungkan VPCs dan jaringan lokal melalui hub pusat. Ini menyederhanakan jaringan Anda dan mengakhiri hubungan peering yang kompleks. Ini bertindak sebagai router cloud, sehingga setiap koneksi baru dibuat hanya satu kali.
- Amazon Virtual Private Cloud (Amazon VPC) adalah layanan untuk meluncurkan AWS sumber daya dalam jaringan virtual yang terisolasi secara logis yang Anda tentukan.
- AWS X-Raymengumpulkan data tentang permintaan yang disajikan aplikasi Anda, dan menyediakan alat yang dapat Anda gunakan untuk melihat, memfilter, dan mendapatkan wawasan tentang data tersebut guna mengidentifikasi masalah dan peluang pengoptimalan.

Kode

Kode sumber untuk solusi ini tersedia di GitHub repositori Tagger Lampiran Transit Gateway. Repositori mencakup file-file berikut:

- `tgw-attachment-tagger-main-stack.yaml`membuat semua sumber daya untuk mendukung solusi ini dalam akun Shared Networking.
- `tgw-attachment-tagger-organizations-stack.yaml`menciptakan peran dalam akun manajemen organisasi.

Epik

Menyebarkan tumpukan solusi utama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan informasi prasyarat yang diperlukan.	Untuk mengonfigurasi akses lintas akun dari fungsi Lambda AWS Organizations ke API, Anda memerlukan ID akun untuk akun manajemen organisasi.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Urutan pembuatan kedua CloudFormation tumpukan itu penting. Anda harus menyebarkan sumber daya ke akun Shared Networking terlebih dahulu. Peran dalam akun Jaringan Bersama harus sudah ada sebelum menyebarkan sumber daya ke akun manajemen organisasi. Lihat informasi yang lebih lengkap dalam dokumentasi AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan CloudFormation template untuk tumpukan solusi utama.	Template untuk tumpukan solusi utama akan menerapkan peran IAM, alur kerja Step Functions, fungsi Lambda, dan acara Amazon. CloudWatch Buka akun Konsol Manajemen AWS untuk Shared Networking, lalu buka konsol: &CFN. Buat tumpukan dengan menggunakan `tgw-attachment-tagger-main-stack.yaml` template dan nilai-nilai berikut: • Nama tumpukan — tgw-attachment-tagger-main-stack • awsOrganizationsRootAccountId— ID Akun untuk akun manajemen organisasi • TGWRegionsparameter — Wilayah AWS untuk solusi, dimasukkan sebagai string yang dibatasi koma • TGWListparameter — Transit gateway IDs untuk dikecualikan dari solusi, dimasukkan dalam string yang dibatasi koma Untuk informasi selengkapnya tentang meluncurkan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CloudFormation tumpukan, lihat AWS dokumentasi.	
Verifikasi bahwa solusi telah berhasil diluncurkan.	Tunggu CloudFormation tumpukan mencapai status CREATE_COMPLETE. Ini akan memakan waktu kurang dari satu menit. Buka konsol Step Functions , dan verifikasi bahwa mesin status baru telah dibuat dengan nama tgw-attachment-tagger-state-machine.	DevOps insinyur

Menerapkan tumpukan AWS Organizations

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan informasi prasyarat yang diperlukan.	Untuk mengonfigurasi akses lintas akun dari fungsi Lambda ke AWS Organizations API, Anda memerlukan ID akun untuk akun Shared Networkin g.	DevOps insinyur
Luncurkan CloudFormation template untuk tumpukan Organizations	Template untuk tumpukan AWS Organizations akan menerapkan peran IAM di akun manajemen organisasi. Akses konsol AWS untuk akun manajemen organisasi. lalu buka CloudFormation konsol. Buat tumpukan dengan menggunakan `tgw-attac`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`hment-tagger-organizations-stack.yaml` template dan nilai-nilai berikut: • Nama tumpukan — tgw-attachment-tagger-organizations-stack • NetworkingAccountIdparameter — ID Akun untuk akun Jaringan Bersama Untuk opsi pembuatan tumpukan lainnya, gunakan default.	
Verifikasi bahwa solusi telah berhasil diluncurkan.	Tunggu CloudFormation tumpukan mencapai status CREATE_COMPLETE. Ini akan memakan waktu kurang dari satu menit. Buka konsol AWS Identity and Access Management (IAM), dan verifikasi bahwa peran baru telah dibuat dengan nama tgw-attachment-tagger-organization-query-role.	DevOps insinyur

Verifikasi solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan mesin negara.	Buka konsol Step Functions untuk akun Shared Networkin	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	g, dan pilih State machine di panel navigasi. Pilih state machine tgw-attachment-tagger-state-machine, dan pilih Start Execution. Karena input ke mesin status ini tidak digunakan oleh solusi, Anda dapat menggunakan nilai default. `{ "Comment": "Insert your JSON here" }` Pilih Mulai Eksekusi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perhatikan mesin negara sampai selesai.	Pada halaman baru yang terbuka, Anda dapat menonton mesin status berjalan. Durasi akan tergantung pada jumlah lampiran Transit Gateway yang akan diproses. Di halaman ini, Anda dapat memeriksa setiap langkah mesin negara. Anda dapat melihat berbagai tugas dalam mesin negara dan mengikuti tautan ke CloudWatch log untuk fungsi Lambda. Untuk tugas yang berjalan secara paralel dalam peta, Anda dapat menggunakan daftar dropdown Indeks untuk melihat implementasi spesifik untuk setiap Wilayah.	DevOps insinyur
Verifikasi tag lampiran Transit Gateway.	Buka konsol VPC untuk akun Shared Networking, dan pilih Lampiran Transit Gateway. Di konsol, tag Nama disediakan untuk lampiran yang memenuhi kriteria (lampiran disebarkan ke tabel rute Transit Gateway, dan pemilik sumber daya adalah anggota organisasi).	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi inisiasi CloudWatch acara.	Tunggu CloudWatch acara dimulai. Ini dijadwalkan untuk 06:00 UTC. Kemudian buka konsol Step Functions untuk akun Shared Networking, dan pilih State machines di panel navigasi. Pilih mesin negara tgw-attachment-tagger-state-mesin. Verifikasi bahwa solusi berjalan pada pukul 06:00 UTC.	DevOps insinyur

Sumber daya terkait

- AWS Organizations
- AWS Resource Access Manager
- Orkestra Jaringan Transit Tanpa Server
- Membuat peran IAM
- Membuat tumpukan di AWS CloudFormation konsol

Lebih banyak pola

- Otomatiskan alokasi Amazon VPC IPv4 IPAM CIDR untuk yang baru dengan menggunakan AFT Akun AWS
- Mengatur set izin untuk beberapa akun dengan menggunakan Account Factory untuk Terraform
- Menyediakan AWS Service Catalog produk berdasarkan AWS CloudFormation template dengan menggunakan GitHub Actions
- Menyediakan peran IAM dengan hak istimewa paling sedikit dengan menerapkan solusi mesin penjual otomatis peran

AI & pembelajaran mesin

Topik

- Kaitkan AWS CodeCommit repositori di satu dengan Akun AWS Amazon SageMaker AI Studio Classic di akun lain
- Secara otomatis mengekstrak konten dari file PDF menggunakan Amazon Ttract
- Buat model peramalan awal dingin dengan menggunakan DeepAR untuk deret waktu di SageMaker Amazon AI Studio Lab
- Membangun MLOps alur kerja dengan menggunakan Amazon SageMaker AI dan Azure DevOps
- Konfigurasikan pencatatan pemanggilan model di Amazon Bedrock dengan menggunakan AWS CloudFormation
- Buat image container Docker kustom untuk SageMaker dan gunakan untuk pelatihan model di AWS Step Functions
- Gunakan agen Amazon Bedrock untuk mengotomatiskan pembuatan kontrol entri akses di Amazon EKS melalui petunjuk berbasis teks
- Terapkan kasus penggunaan RAG AWS dengan menggunakan Terraform dan Amazon Bedrock
- Menerapkan logika preprocessing ke dalam model MLdalam satu titik akhir menggunakan pipeline inferensi di Amazon SageMaker
- Menyebarkan validasi keamanan pengkodean real-time dengan menggunakan server MCP dengan Kiro dan asisten pengkodean lainnya
- Kembangkan asisten berbasis obrolan AI generatif tingkat lanjut dengan menggunakan RAG dan prompt ReAct
- Kembangkan asisten berbasis obrolan otomatis sepenuhnya dengan menggunakan agen dan basis pengetahuan Amazon Bedrock
- Dokumentasikan pengetahuan kelembagaan dari input suara dengan menggunakan Amazon Bedrock dan Amazon Transcribe
- Buat rekomendasi yang dipersonalisasi dan diberi peringkat ulang menggunakan Amazon Personalize
- Merampingkan alur kerja pembelajaran mesin dari pengembangan lokal hingga eksperimen yang dapat diskalakan dengan menggunakan SageMaker AI dan Hydra
- Terjemahkan bahasa alami ke dalam kueri DSL untuk OpenSearch dan kueri Elasticsearch

- Gunakan Amazon Q Developer sebagai asisten pengkodean untuk meningkatkan produktivitas Anda
- Gunakan SageMaker Processing untuk rekayasa fitur terdistribusi dari kumpulan data ML skala terabyte
- Visualisasikan hasil AI/ML model menggunakan Flask dan AWS Elastic Beanstalk
- Lebih banyak pola

Kaitkan AWS CodeCommit repositori di satu dengan Akun AWS Amazon SageMaker AI Studio Classic di akun lain

Laurens van der Maas dan Aubrey Oosthuizen, Amazon Web Services

Ringkasan

Pola ini memberikan instruksi dan kode tentang cara mengaitkan AWS CodeCommit repositori di satu Akun AWS (Akun A) dengan Amazon SageMaker AI Studio Classic di tempat lain Akun AWS (Akun B). Untuk menyiapkan asosiasi, Anda harus membuat kebijakan dan peran AWS Identity and Access Management (IAM) di Akun A dan kebijakan sebaris IAM di Akun B. Kemudian, Anda menggunakan skrip shell untuk mengkloning CodeCommit repositori dari Akun A ke Amazon AI Classic di Akun B. SageMaker

Prasyarat dan batasan

Prasyarat

- Dua Akun AWS, satu berisi CodeCommit repositori dan yang lainnya berisi Domain SageMaker AI dengan pengguna
- Domain dan pengguna SageMaker AI yang disediakan, dengan akses internet atau akses ke CodeCommit dan AWS Security Token Service (AWS STS) melalui titik akhir jaringan pribadi virtual (VPC)
- Pemahaman dasar IAM
- Pemahaman dasar tentang SageMaker AI Studio Classic
- Pemahaman dasar tentang Git dan CodeCommit

Keterbatasan

Pola ini hanya berlaku untuk SageMaker AI Studio Classic, bukan RStudio di Amazon SageMaker AI.

Arsitektur

Tumpukan teknologi

- Amazon SageMaker AI
- Amazon SageMaker AI Studio Klasik

- AWS CodeCommit
- AWS Identity and Access Management (IAM)
- Git

Arsitektur target

Diagram berikut menunjukkan arsitektur yang mengaitkan CodeCommit repositori dari Akun A ke SageMaker AI Studio Classic di Akun B.

Diagram menunjukkan alur kerja berikut:

1. Pengguna mengasumsikan `MyCrossAccountRepositoryContributorRole` peran dalam Akun A melalui peran, saat menggunakan `sts:AssumeRole` peran eksekusi SageMaker AI di SageMaker AI Studio Classic di Akun B. Peran yang diasumsikan mencakup CodeCommit izin untuk mengkloning dan berinteraksi dengan repositori yang ditentukan.
2. Pengguna melakukan perintah Git dari terminal sistem di SageMaker AI Studio Classic.

Otomatisasi dan skala

Pola ini terdiri dari langkah-langkah manual yang dapat diotomatisasi dengan menggunakan AWS Cloud Development Kit (AWS CDK), AWS CloudFormation, atau Terraform.

Alat

Alat AWS

- Amazon SageMaker AI adalah layanan pembelajaran mesin terkelola (ML) yang membantu Anda membangun dan melatih model ML, lalu menerapkannya ke lingkungan host yang siap produksi.
- Amazon SageMaker AI Studio Classic adalah lingkungan pengembangan terintegrasi (IDE) berbasis web untuk pembelajaran mesin yang memungkinkan Anda membuat, melatih, men-debug, menerapkan, dan memantau model pembelajaran mesin Anda.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.

  Pemberitahuan: AWS CodeCommit tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS CodeCommit dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Alat lainnya

- Git adalah sistem kontrol versi terdistribusi untuk melacak perubahan kode sumber selama pengembangan perangkat lunak.

Epik

Buat kebijakan IAM dan peran IAM di Akun A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM untuk akses repositori di Akun A.	1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM. 2. Pada panel navigasi, pilih Kebijakan, lalu pilih Buat kebijakan. 3. Pilih tab JSON. 4. Salin pernyataan kebijakan dari Contoh kebijakan IAM di bagian Informasi tambahan dari pola ini, lalu tempelkan pernyataan tersebut ke editor JSON. Pastikan untuk mengganti semua nilai placeholder dalam kebijakan. 5. Pilih Berikutnya:Tag, lalu pilih Berikutnya:Ulasan. 6. Untuk Nama, masukkan nama untuk kebijakan.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Catatan: Dalam pola ini, kebijakan IAM disebut sebagai`CrossAccountAccessForMySharedDemoRepo` , tetapi Anda dapat memilih nama kebijakan apa pun yang Anda inginkan. 7. Pilih Buat kebijakan. **Tip** Ini adalah praktik terbaik untuk membatasi cakupan kebijakan IAM Anda ke izin minimum yang diperlukan untuk kasus penggunaan Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk akses repositori di Akun A.	1. Di panel navigasi konsol IAM, pilih Peran, lalu pilih Buat peran. 2. Untuk jenis entitas Tepercaya, pilih akun AWS. 3. Di bagian akun AWS, pilih Akun AWS lain. 4. Untuk ID Akun, masukkan ID akun untuk Akun B. 5. Pada halaman Tambah izin, cari dan pilih `CrossAccountAccessForMySharedDemoRepo` kebijakan yang Anda buat sebelumnya. 6. Pilih Berikutnya. 7. Untuk nama Peran, masukkan nama. Catatan: Dalam pola ini, nama peran IAM disebut sebagai`MyCrossAccountRepositoryContributorRole`, tetapi Anda dapat memilih nama peran apa pun yang Anda inginkan. 8. Pilih Buat peran, lalu salin Nama Sumber Daya Amazon (ARN) dari peran baru.	AWS DevOps

Buat kebijakan inline IAM di Akun B

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan kebijakan inline ke peran eksekusi yang dilampirkan ke pengguna SageMaker Domain Anda di Akun B.	1. Di panel navigasi konsol IAM, pilih Peran. 2. Cari dan pilih peran eksekusi yang dilampirkan ke pengguna Domain SageMaker AI Anda di Akun B. 3. Pilih Tambahkan izin, lalu pilih Buat kebijakan sebaris. 4. Pilih tab JSON. 5. Salin pernyataan kebijakan berikut, lalu tempelkan ke editor JSON. `{ "Version": "2012-10-17", "Statement": [ { "Sid": "VisualEditor0", "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<Account_A_ID>:role/<Account_A_Role_Name>" } ] }`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Ganti `<Account_A_ID>` dengan ID akun untuk Akun A. Ganti `<Account_A_Role_Name>` dengan nama peran IAM yang Anda buat sebelumnya. 7. Pilih Tinjau kebijakan. 8. Untuk Nama, masukkan nama untuk kebijakan inline Anda. 9. Pilih Buat kebijakan.	

Kloning repositori di SageMaker AI Studio Classic untuk Akun B

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skrip shell di SageMaker AI Studio Classic di Akun B.	1. Di panel navigasi SageMaker konsol, pilih Studio. 2. Pilih profil pengguna Anda dan kemudian pilih Open Studio. 3. Di bagian Beranda, pilih Buka Peluncur. 4. Di bagian Utilitas dan file, pilih File teks. 5. Salin skrip dari Contoh skrip SageMaker shell di bagian Informasi tambahan dari pola ini, lalu tempelkan pernyataan ke file baru. Pastikan untuk mengganti	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	semua nilai placeholder dalam skrip. 6. Klik kanan tab untitled.txt pada file baru Anda, lalu pilih Ganti Nama Teks. Untuk Nama Baru, masukkan cross_account_git_clone.sh, lalu pilih Ganti nama.	
Memanggil skrip shell dari terminal sistem.	1. Di bagian Beranda SageMaker konsol, pilih Buka Peluncur. 2. Di bagian Utilitas dan file, pilih Terminal sistem. 3. Di terminal, jalankan perintah berikut: `chmod u+x ./cross_account_git_clone.sh && ./cross_account_git_clone.sh` Anda telah mengkloning CodeCommit repositori Anda di akun lintas SageMaker AI Studio. Anda sekarang dapat melakukan semua perintah Git dari terminal sistem.	AWS DevOps

Informasi tambahan

Contoh kebijakan IAM

Jika Anda menggunakan kebijakan contoh ini, lakukan hal berikut:

- Ganti <CodeCommit_Repository_Region> dengan Wilayah AWS untuk repositori.
- Ganti <Account_A_ID> dengan ID akun untuk Akun A.
- Ganti <CodeCommit_Repository_Name> dengan nama CodeCommit repositori Anda di Akun A.

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "codecommit:BatchGet*",
            "codecommit:Create*",
            "codecommit:DeleteBranch",
            "codecommit:Get*",
            "codecommit:List*",
            "codecommit:Describe*",
            "codecommit:Put*",
            "codecommit:Post*",
            "codecommit:Merge*",
            "codecommit:Test*",
            "codecommit:Update*",
            "codecommit:GitPull",
            "codecommit:GitPush"
        ],
        "Resource": [

 "arn:aws:codecommit:<CodeCommit_Repository_Region>:<Account_A_ID>:<CodeCommit_Repository_Name>
        ]
    }
]
}
```

Contoh skrip shell SageMaker AI

Jika Anda menggunakan contoh skrip ini, lakukan hal berikut:

- Ganti <Account_A_ID> dengan ID akun untuk Akun A.
- Ganti <Account_A_Role_Name> dengan nama peran IAM yang Anda buat sebelumnya.

- Ganti <CodeCommit_Repository_Region> dengan Wilayah AWS untuk repositori.
- Ganti <CodeCommit_Repository_Name> dengan nama CodeCommit repositori Anda di Akun A.

```
#!/usr/bin/env bash
#Launch from system terminal
pip install --quiet git-remote-codecommit

mkdir -p ~/.aws
touch ~/.aws/config

echo "[profile CrossAccountAccessProfile]
region = <CodeCommit_Repository_Region>
credential_source=EcsContainer
role_arn = arn:aws:iam::<Account_A_ID>:role/<Account_A_Role_Name>
output = json" > ~/.aws/config

echo '[credential "https://git-
codecommit.<CodeCommit_Repository_Region>.amazonaws.com"]
        helper = !aws codecommit credential-helper $@ --profile
 CrossAccountAccessProfile
        UseHttpPath = true' > ~/.gitconfig

git clone codecommit::<CodeCommit_Repository_Region>://
CrossAccountAccessProfile@<CodeCommit_Repository_Name>
```

Secara otomatis mengekstrak konten dari file PDF menggunakan Amazon Ttract

Tianxia Jia, Amazon Web Services

Ringkasan

Banyak organisasi perlu mengekstrak informasi dari file PDF yang diunggah ke aplikasi bisnis mereka. Misalnya, organisasi mungkin perlu mengekstrak informasi secara akurat dari file PDF pajak atau medis untuk analisis pajak atau pemrosesan klaim medis.

Di Amazon Web Services (AWS) Cloud, Amazon Textract secara otomatis mengekstrak informasi (misalnya, teks cetak, formulir, dan tabel) dari file PDF dan menghasilkan file berformat JSON yang berisi informasi dari file PDF asli. Anda dapat menggunakan Amazon Ttract di AWS Management Console atau dengan menerapkan panggilan API. Kami menyarankan Anda menggunakan panggilan API terprogram untuk menskalakan dan secara otomatis memproses sejumlah besar file PDF.

Saat Amazon Textract memproses file, ia membuat daftar `Block` objek berikut: halaman, baris dan kata-kata teks, formulir (pasangan nilai kunci), tabel dan sel, dan elemen pemilihan. Informasi objek lainnya juga disertakan, misalnya, kotak pembatas, interval kepercayaan IDs, dan hubungan. Amazon Ttract mengekstrak informasi konten sebagai string. Nilai data yang diidentifikasi dan diubah dengan benar diperlukan karena dapat lebih mudah digunakan oleh aplikasi hilir Anda.

Pola ini menjelaskan step-by-step alur kerja untuk menggunakan Amazon Ttract untuk secara otomatis mengekstrak konten dari file PDF dan memprosesnya menjadi output yang bersih. Pola menggunakan teknik pencocokan template untuk mengidentifikasi dengan benar bidang yang diperlukan, nama kunci, dan tabel, dan kemudian menerapkan koreksi pasca-pemrosesan untuk setiap tipe data. Anda dapat menggunakan pola ini untuk memproses berbagai jenis file PDF dan Anda kemudian dapat menskalakan dan mengotomatiskan alur kerja ini untuk memproses file PDF yang memiliki format yang identik.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Bucket Amazon Simple Storage Service (Amazon S3) yang ada untuk menyimpan file PDF setelah dikonversi ke format JPEG untuk diproses oleh Amazon Textract. Untuk informasi selengkapnya tentang bucket S3, lihat ikhtisar Bucket di dokumentasi Amazon S3.

- Notebook `Textract_PostProcessing.ipynb` Jupyter (terlampir), diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang notebook Jupyter, lihat Membuat buku catatan Jupyter di dokumentasi Amazon. SageMaker
- File PDF yang ada yang memiliki format identik.
- Pemahaman tentang Python.

Batasan

- File PDF Anda harus berkualitas baik dan dapat dibaca dengan jelas. File PDF asli direkomendasikan, tetapi Anda dapat menggunakan dokumen yang dipindai yang dikonversi ke format PDF jika semua kata individual jelas. Untuk informasi selengkapnya tentang ini, lihat praprosesan dokumen PDF dengan Amazon Textract: Deteksi dan penghapusan visual di Blog AWS Machine Learning.
- Untuk file multipage, Anda dapat menggunakan operasi asinkron atau membagi file PDF menjadi satu halaman dan menggunakan operasi sinkron. Untuk informasi selengkapnya tentang dua opsi ini, lihat Mendeteksi dan menganalisis teks dalam dokumen multihalaman serta Mendeteksi dan menganalisis teks dalam dokumen satu halaman dalam dokumentasi Amazon Textract.

Arsitektur

Alur kerja pola ini pertama-tama menjalankan Amazon Ttract pada contoh file PDF (Jalankan pertama kali) dan kemudian menjalankannya pada file PDF yang memiliki format identik dengan PDF pertama (Ulangi jalankan). Diagram berikut menunjukkan gabungan alur kerja First-time run dan Repeat run yang secara otomatis dan berulang kali mengekstrak konten dari file PDF dengan format yang identik.

Diagram menunjukkan alur kerja berikut untuk pola ini:

1. Konversi file PDF ke dalam format JPEG dan simpan dalam ember S3.
2. Panggil Amazon Texttract API dan uraikan file JSON respons Amazon Textract.
3. Edit file JSON dengan menambahkan `KeyName:DataType` pasangan yang benar untuk setiap bidang yang diperlukan. Buat `TemplateJSON` file untuk tahap Repeat run.
4. Tentukan fungsi koreksi pasca-pemrosesan untuk setiap tipe data (misalnya, float, integer, dan tanggal).

5. Siapkan file PDF yang memiliki format yang identik dengan file PDF pertama Anda.
6. Hubungi Amazon Texttract API dan uraikan respons Amazon Textract JSON.
7. Cocokkan file JSON yang diurai dengan file. `TemplateJSON`
8. Menerapkan koreksi pasca-pemrosesan.

File keluaran JSON akhir memiliki yang benar `KeyName` dan `Value` untuk setiap bidang yang diperlukan.

Tumpukan teknologi target

- Amazon SageMaker
- Amazon S3
- Amazon Textract

Otomatisasi dan skala

Anda dapat mengotomatiskan alur kerja Repeat run dengan menggunakan fungsi AWS Lambda yang memulai Amazon Ttract saat file PDF baru ditambahkan ke Amazon S3. Amazon Textract kemudian menjalankan skrip pemrosesan dan output akhir dapat disimpan ke lokasi penyimpanan. Untuk informasi selengkapnya tentang hal ini, lihat Menggunakan pemicu Amazon S3 untuk menjalankan fungsi Lambda dalam dokumentasi Lambda.

Alat

- Amazon SageMaker adalah layanan ML yang dikelola sepenuhnya yang membantu Anda membuat dan melatih model ML dengan cepat dan mudah, lalu langsung menerapkannya ke lingkungan host yang siap produksi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Textract memudahkan untuk menambahkan deteksi dan analisis teks dokumen ke aplikasi Anda.

Epik

Lari pertama kali

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konversi file PDF.	Siapkan file PDF untuk pertama kali dijalankan dengan membaginya menjadi satu halaman dan mengubahnya menjadi format JPEG untuk operasi sinkron Amazon Textract (). `Syn API` **Note** Anda juga dapat menggunakan operasi asinkron Amazon Textract (`Asyn API`) untuk file PDF multihalaman.	Ilmuwan data, Pengembang
Mengurai respons Amazon Texttract JSON.	Buka `Textract_PostProcessing.ipynb` notebook Jupyter (terlampir) dan panggil Amazon Textract API dengan menggunakan kode berikut: `response = textract.analyze_document( Document={ 'S3Object': { 'Bucket': BUCKET, 'Name': '{}'.format(filename)`	Ilmuwan data, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> }
	                },
	        FeatureTy
	pes=["TABLES",
	 "FORMS"])</pre>
	Parse JSON respon ke dalam bentuk dan tabel dengan menggunakan kode berikut:
	<pre>parseformKV=form_k
	v_from_JSON(response)
	parseformTable
	s=get_tables_fromJ
	SON(response)</pre>	

Edit file TemplateJSON.	Edit JSON yang diurai untuk masing-masing `KeyName` dan yang sesuai `DataType` (misalnya, string, float, integer, atau tanggal), dan header tabel (misalnya, dan). `ColumnNames` `RowNames` Template ini digunakan untuk setiap jenis file PDF individu, yang berarti bahwa template dapat digunakan kembali untuk file PDF yang memiliki format yang identik.	Ilmuwan data, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan fungsi koreksi pasca-pemrosesan.	Nilai dalam respons Amazon Textract untuk `TemplateJSON` file tersebut adalah string. Tidak ada diferensiasi untuk tanggal, float, integer, atau mata uang. Nilai-nilai ini harus dikonversi ke tipe data yang benar untuk kasus penggunaan hilir Anda. Perbaiki setiap tipe data sesuai dengan `TemplateJSON` file dengan menggunakan kode berikut: `finalJSON=postprocessingCorrection(parsedJSON,templateJSON)`	Ilmuwan data, Pengembang

Ulangi lari

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan file PDF.	Siapkan file PDF dengan membaginya menjadi satu halaman dan mengubahnya menjadi format JPEG untuk operasi sinkron Amazon Textract (). `Syn API` **Note** Anda juga dapat menggunakan operasi	Ilmuwan data, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	asinkron Amazon Textract (Asyn API) untuk file PDF multihalaman.	
Hubungi Amazon Texttract API.	Hubungi Amazon Texttract API dengan menggunakan kode berikut: `response = textract.analyze_document( Document={ 'S3Object': { 'Bucket': BUCKET, 'Name': '{}'.format(filename) } }, FeatureTypes=["TABLES", "FORMS"])`	Ilmuwan data, Pengembang
Mengurai respons Amazon Texttract JSON.	Parse JSON respon ke dalam bentuk dan tabel dengan menggunakan kode berikut: `parseformKV=form_kv_from_JSON(response) parseformTables=get_tables_fromJSON(response)`	Ilmuwan data, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Muat file TemplateJSON dan cocokkan dengan JSON yang diurai.	Gunakan `TemplateJSON` file untuk mengekstrak pasangan kunci-nilai yang benar dan tabel dengan menggunakan perintah berikut: `form_kv_corrected= form_kv_correction (parseformKV,templ ateJSON) form_table_correct ed=form_Table_corr ection(parseformTa bles, templateJSON) form_kv_table_correcte d_final={**form_kv _corrected , **form_ta ble_corrected}`	Ilmuwan data, Pengembang
Koreksi pasca pemrosesan.	Gunakan `DataType` dalam `TemplateJSON` file dan fungsi pasca-pemrosesan untuk mengoreksi data dengan menggunakan kode berikut: `finalJSON=postproc essingCorrection(f orm_kv_table_corre cted_final,templat eJSON)`	Ilmuwan data, Pengembang

Sumber daya terkait

- Secara otomatis mengekstrak teks dan data terstruktur dari dokumen dengan Amazon Ttract
- Ekstrak teks dan data terstruktur dengan Amazon Textract

- Sumber daya Amazon Texttract

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat model peramalan awal dingin dengan menggunakan DeepAR untuk deret waktu di SageMaker Amazon AI Studio Lab

Ivan Cui dan Eyal Shacham, Amazon Web Services

Ringkasan

Apakah Anda mengalokasikan sumber daya secara lebih efisien untuk lalu lintas web, memperkirakan permintaan pasien untuk kebutuhan staf, atau mengantisipasi penjualan produk perusahaan, peramalan adalah alat yang penting. Peramalan awal dingin membangun perkiraan untuk deret waktu yang memiliki sedikit data historis, seperti produk baru yang baru saja memasuki pasar ritel. Pola ini menggunakan algoritme peramalan Amazon SageMaker AI DeepAR untuk melatih model peramalan awal dingin dan menunjukkan cara melakukan peramalan pada item start dingin.

DeepAR adalah algoritma pembelajaran yang diawasi untuk meramalkan deret waktu skalar (satu dimensi) menggunakan jaringan saraf berulang (RNN). DeepAR mengambil pendekatan pelatihan satu model bersama-sama di semua rangkaian waktu deret waktu produk terkait.

Metode peramalan deret waktu tradisional seperti autoregressive integrated moving average (ARIMA) atau exponential smoothing (ETS) sangat bergantung pada deret waktu historis dari setiap produk individu. Oleh karena itu, metode tersebut tidak efektif untuk peramalan awal dingin. Ketika kumpulan data Anda berisi ratusan deret waktu terkait, DeepAR mengungguli metode ARIMA dan ETS standar. Anda juga dapat menggunakan model terlatih untuk menghasilkan perkiraan untuk deret waktu baru yang mirip dengan deret waktu yang telah dilatihnya.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Domain SageMaker AI Amazon.
- Aplikasi lab Amazon SageMaker AI Studio atau Jupiter.
- Bucket Amazon Simple Storage Service (Amazon S3) dengan izin baca dan tulis.
- Pengetahuan tentang pemrograman dengan Python.
- Pengetahuan tentang menggunakan notebook Jupyter.

Batasan

- Memanggil model perkiraan tanpa titik data historis akan mengembalikan kesalahan. Memohon model dengan titik data historis minimal akan mengembalikan prediksi yang tidak akurat dengan keyakinan tinggi. Pola ini menyarankan pendekatan untuk menyelesaikan keterbatasan peramalan awal dingin yang diketahui ini.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat layanan AWS menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Python versi 3.10 atau yang lebih baru.
- Notebook pola diuji di Amazon SageMaker AI Studio pada instance ml.t3.medium dengan kernel Python 3 (Data Science).

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Alur kerja melakukan tugas-tugas berikut:

1. File input data pelatihan dan pengujian disintesis dan kemudian diunggah ke bucket Amazon S3. Data ini mencakup beberapa deret waktu dengan fitur kategoris dan dinamis, bersama dengan nilai target (untuk diprediksi). Notebook Jupyter memvisualisasikan data untuk lebih memahami persyaratan data pelatihan dan nilai prediksi yang diharapkan.
2. Pekerjaan tuner hyperparameters dibuat untuk melatih model dan menemukan model terbaik berdasarkan metrik yang telah ditentukan.
3. File input diunduh dari bucket Amazon S3 ke setiap instance pekerjaan penyetelan hyperparameters.
4. Setelah pekerjaan tuner memilih model terbaik berdasarkan ambang batas yang telah ditentukan tuner, model tersebut digunakan sebagai titik akhir AI. SageMaker
5. Model yang diterapkan kemudian siap untuk dipanggil di mana prediksinya divalidasi terhadap data pengujian.

Notebook menunjukkan seberapa baik model memprediksi nilai target ketika jumlah titik data historis yang memadai tersedia. Namun, ketika kami menggunakan model dengan titik data historis yang lebih sedikit (yang mewakili produk dingin), prediksi model tidak cocok dengan data pengujian asli bahkan dalam tingkat kepercayaan model. Dalam polanya, model baru dibangun untuk produk dingin di mana panjang konteks awalnya (titik prediksi) didefinisikan sebagai jumlah titik sejarah yang tersedia, dan model baru dilatih secara iteratif saat titik data baru diperoleh. Notebook menunjukkan bahwa model akan memiliki prediksi yang akurat selama jumlah titik data historis mendekati panjang konteksnya.

Alat

Layanan AWS

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon SageMaker AI adalah layanan pembelajaran mesin terkelola (ML) yang membantu Anda membangun dan melatih model ML, lalu menerapkannya ke lingkungan host yang siap produksi.
- Amazon SageMaker AI Studio adalah lingkungan pengembangan terintegrasi (IDE) berbasis web untuk ML yang memungkinkan Anda membuat, melatih, men-debug, menerapkan, dan memantau model ML Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat lainnya

- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub DeepAR- ColdProduct -Pattern.

Praktik terbaik

- Latih model Anda di lingkungan virtual, dan selalu gunakan kontrol versi untuk upaya reproduktifitas tertinggi.

- Sertakan sebanyak mungkin fitur kategoris berkualitas tinggi untuk mendapatkan model prediktif tertinggi.
- Pastikan bahwa metadata berisi item kategoris yang serupa agar model dapat menyimpulkan prediksi produk start dingin secara memadai.
- Jalankan pekerjaan tuning hyperparameter untuk mendapatkan model prediktif tertinggi.
- Dalam pola ini, model yang Anda buat memiliki panjang konteks 24 jam, yang berarti akan memprediksi 24 jam ke depan. Jika Anda mencoba memprediksi 24 jam ke depan ketika Anda memiliki data kurang dari 24 jam secara historis, akurasi prediksi model menurun secara linier berdasarkan jumlah titik data historis. Untuk mengurangi masalah ini, buat model baru untuk setiap set titik data historis hingga angka ini mencapai panjang prediksi (konteks) yang diinginkan. Misalnya, mulailah dengan model panjang konteks 2 jam, kemudian tingkatkan model secara progresif menjadi 4 jam, 8 jam, 16 jam, dan 24 jam.

Epik

Mulai aplikasi SageMaker AI Studio Classic Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai lingkungan notebook Anda.	1. Masuk ke Konsol Manajemen AWS, dan buka beranda SageMaker AI Studio. Kemudian pilih Open Studio. 2. Di panel navigasi kiri, pilih ikon Studio Classic di Aplikasi. Kemudian, pilih tombol Buka pada daftar Aplikasi. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker AI Studio di dokumentasi SageMaker AI.	Ilmuwan data

Buat dan aktifkan notebook

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan lingkungan virtual Anda untuk pelatihan model.	Untuk mengatur lingkungan virtual Anda untuk pelatihan model, lakukan hal berikut: 1. Unduh `deepar_synthetic.ipynb` buku catatan dari GitHub repositori pola ini ke mesin lokal Anda. 2. Di Amazon SageMaker AI Studio Classic, pilih ikon Unggah File dari bilah menu Studio Classic dan pilih buku catatan yang diunduh. 3. Pilih buku catatan di File Browser di panel navigasi kiri. Ikuti petunjuk untuk menyiapkan lingkungan notebook. Pilih Data Science 3.0 Image dan Python 3 Kernel. Untuk informasi selengkapnya, lihat Mengunggah File ke SageMaker AI Studio Classic dalam dokumentasi SageMaker AI.	Ilmuwan data
Buat dan validasi model peramalan.	• Ikuti instruksi di buku catatan untuk membuat data pelatihan dan pengujian, melatih model, dan	Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kemudian memanggil model. • Amati seberapa akurat prediksi model ketika dilengkapi dengan titik data historis yang memadai.	

Sumber daya terkait

- DeepAR Hyperparameter
- Perkiraan permintaan untuk pengenalan produk baru dengan menggunakan layanan pembelajaran AWS mesin
- Luncurkan Amazon SageMaker AI Studio Classic
- Gunakan algoritma SageMaker peramalan AI DeepAR

Membangun MLOps alur kerja dengan menggunakan Amazon SageMaker AI dan Azure DevOps

Deepika Kumar, Sara van de Moosdijk, dan Philips Kokoh Prasetyo, Amazon Web Services

Ringkasan

Operasi pembelajaran mesin (MLOps) adalah seperangkat praktik yang mengotomatiskan dan menyederhanakan alur kerja dan penerapan pembelajaran mesin (ML). MLOps berfokus pada otomatisasi siklus hidup ML. Ini membantu memastikan bahwa model tidak hanya dikembangkan tetapi juga digunakan, dipantau, dan dilatih ulang secara sistematis dan berulang kali. Ini membawa DevOps prinsip ke ML. MLOps menghasilkan penerapan model ML yang lebih cepat, akurasi yang lebih baik dari waktu ke waktu, dan jaminan yang lebih kuat bahwa model tersebut memberikan nilai bisnis yang nyata.

Organizations sering memiliki DevOps alat dan solusi penyimpanan data yang ada sebelum memulai MLOps perjalanan mereka. Pola ini menampilkan cara memanfaatkan kekuatan Microsoft Azure dan. AWS Ini membantu Anda mengintegrasikan Azure DevOps dengan Amazon SageMaker AI untuk membuat MLOps alur kerja.

Solusinya menyederhanakan kerja antara Azure dan. AWS Anda dapat menggunakan Azure untuk pengembangan dan AWS pembelajaran mesin. Ini mempromosikan proses yang efektif untuk membuat model pembelajaran mesin dari awal hingga akhir, termasuk penanganan data, pelatihan, dan penyebaran. AWS Untuk efisiensi, Anda mengelola proses ini melalui DevOps pipa Azure. Solusi ini berlaku untuk operasi model dasar (FMOps) dan operasi model bahasa besar (LLMOps) dalam AI generatif, yang mencakup fine-tuning, database vektor, dan manajemen yang cepat.

Prasyarat dan batasan

Prasyarat

- Langganan Azure — Akses ke layanan Azure, seperti Azure DevOps, untuk menyiapkan pipeline integrasi berkelanjutan dan penyebaran berkelanjutan (CI/CD).
- Active AWS Active Account — Izin untuk menggunakan yang Layanan AWS digunakan dalam pola ini.
- Data — Akses ke data historis untuk melatih model pembelajaran mesin.
- Keakraban dengan konsep ML — Pemahaman tentang Python, Notebook Jupyter, dan pengembangan model pembelajaran mesin.

- Konfigurasi keamanan — Konfigurasi peran, kebijakan, dan izin yang tepat di Azure dan AWS untuk memastikan transfer dan akses data yang aman.
- (Opsional) Database vektor — Jika Anda menggunakan pendekatan Retrieval Augmented Generation (RAG) dan layanan pihak ketiga untuk database vektor, Anda memerlukan akses ke database vektor eksternal.

Batasan

- Panduan ini tidak membahas transfer data lintas cloud yang aman. Untuk informasi selengkapnya tentang transfer data lintas cloud, lihat AWS Solusi untuk Hybrid dan Multicloud.
- Solusi multicloud dapat meningkatkan latensi untuk pemrosesan data real-time dan inferensi model.
- Panduan ini memberikan salah satu contoh MLOps arsitektur multi-akun. Penyesuaian diperlukan berdasarkan pembelajaran dan AWS strategi mesin Anda.
- Panduan ini tidak menjelaskan penggunaan AI/ML layanan selain Amazon SageMaker AI.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Arsitektur target

Arsitektur target mengintegrasikan Azure dengan DevOps Amazon SageMaker AI, menciptakan alur kerja ML-cloud. Ini menggunakan Azure untuk CI/CD proses dan SageMaker AI untuk pelatihan dan penerapan model ML. Ini menguraikan proses memperoleh data (dari sumber seperti Amazon S3, Snowflake, dan Azure Data Lake) melalui pembuatan dan penyebaran model. Komponen utama termasuk CI/CD pipeline untuk pembuatan dan penerapan model, persiapan data, manajemen infrastruktur, dan Amazon SageMaker AI untuk pelatihan dan fine-tuning, evaluasi, dan penerapan model ML. Arsitektur ini dirancang untuk menyediakan alur kerja ML yang efisien, otomatis, dan terukur di seluruh platform cloud.

Arsitektur terdiri dari komponen-komponen berikut:

1. Ilmuwan data melakukan eksperimen ML di akun pengembangan untuk mengeksplorasi pendekatan yang berbeda untuk kasus penggunaan ML dengan menggunakan berbagai sumber

data. Ilmuwan data melakukan pengujian unit dan uji coba, dan untuk melacak eksperimen mereka, mereka dapat menggunakan Amazon SageMaker AI MLflow. Dalam pengembangan model AI generatif, ilmuwan data menyempurnakan model fondasi dari hub model SageMaker AI JumpStart Amazon. Setelah evaluasi model, ilmuwan data mendorong dan menggabungkan kode ke repositori Model Build, yang di-host di Azure. DevOps Repositori ini berisi kode untuk pipa bangunan model multi-langkah.

2. Di Azure DevOps, pipeline Model Build, yang menyediakan integrasi berkelanjutan (CI), dapat diaktifkan secara otomatis atau manual setelah penggabungan kode ke cabang utama. Di akun Otomasi, ini mengaktifkan pipeline SageMaker AI untuk pra-pemrosesan data, pelatihan model dan fine-tuning, evaluasi model, dan pendaftaran model bersyarat berdasarkan akurasi.
3. Akun Otomasi adalah akun pusat di seluruh platform MLyang meng-host lingkungan ML (Amazon ECR), model (Amazon S3), metadata model SageMaker (AI Model Registry), fitur (AI Feature Store), pipeline otomatis SageMaker (AI Pipelines), dan wawasan log ML SageMaker (). CloudWatch Untuk beban kerja AI generatif, Anda mungkin memerlukan evaluasi tambahan untuk petunjuk di aplikasi hilir. Aplikasi manajemen yang cepat membantu merampingkan dan mengotomatiskan proses. Akun ini memungkinkan penggunaan kembali aset ML dan memberlakukan praktik terbaik mempercepat pengiriman kasus penggunaan ML.
4. Versi model terbaru ditambahkan ke SageMaker AI Model Registry untuk ditinjau. Ini melacak versi model dan artefak masing-masing (garis keturunan dan metadata). Ini juga mengelola status model (menyetujui, menolak, atau menunggu), dan mengelola versi untuk penyebaran hilir.
5. Setelah model terlatih di Model Registry disetujui melalui antarmuka studio atau panggilan API, acara dapat dikirim ke Amazon. EventBridge EventBridge memulai pipa Model Deploy di DevOps Azure.
6. Pipeline Model Deploy, yang menyediakan continuous deployment (CD), memeriksa sumber dari repositori Model Deploy. Sumber berisi kode, konfigurasi untuk penerapan model, dan skrip pengujian untuk tolok ukur kualitas. Pipeline Model Deploy dapat disesuaikan dengan jenis inferensi Anda.
7. Setelah pemeriksaan kontrol kualitas, pipeline Model Deploy menyebarkan model ke akun Staging. Akun Staging adalah salinan akun Produksi, dan digunakan untuk pengujian dan evaluasi integrasi. Untuk transformasi batch, pipeline Model Deploy dapat secara otomatis memperbarui proses inferensi batch untuk menggunakan versi model terbaru yang disetujui. Untuk inferensi real-time, tanpa server, atau asinkron, ini mengatur atau memperbarui titik akhir model masing-masing.
8. Setelah pengujian berhasil di akun Staging, model dapat digunakan ke akun Produksi dengan persetujuan manual melalui pipeline Model Deploy. Pipeline ini menyediakan titik akhir produksi

dalam langkah Deploy to production, termasuk pemantauan model dan mekanisme umpan balik data.

9. Setelah model diproduksi, gunakan alat seperti SageMaker AI Model Monitor dan SageMaker AI Clarify untuk mengidentifikasi bias, mendeteksi penyimpangan, dan terus memantau kinerja model.

Otomatisasi dan skala

Gunakan infrastruktur sebagai kode (IAc) untuk secara otomatis menyebarkan ke beberapa akun dan lingkungan. Dengan mengotomatiskan proses pengaturan MLOps alur kerja, dimungkinkan untuk memisahkan lingkungan yang digunakan oleh tim ML yang mengerjakan proyek yang berbeda. AWS CloudFormationmembantu Anda memodelkan, menyediakan, dan mengelola AWS sumber daya dengan memperlakukan infrastruktur sebagai kode.

Alat

Layanan AWS

- Amazon SageMaker AI adalah layanan ML terkelola yang membantu Anda membangun dan melatih model ML dan kemudian menerapkannya ke lingkungan host yang siap produksi.
- AWS Glueadalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Dalam pola ini, Amazon S3 digunakan untuk penyimpanan data dan terintegrasi dengan SageMaker AI untuk pelatihan model dan objek model.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan. Dalam pola ini, Lambda digunakan untuk tugas pra-pemrosesan dan pasca-pemrosesan data.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal. Dalam pola ini, ia menyimpan wadah Docker yang digunakan SageMaker AI sebagai lingkungan pelatihan dan penerapan.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Dalam pola ini,

EventBridge mengatur alur kerja berbasis peristiwa atau berbasis waktu yang memulai pelatihan ulang atau penerapan model otomatis.

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun. Dalam pola ini, ini digunakan untuk membuat titik masuk tunggal yang menghadap eksternal untuk SageMaker titik akhir AI.
- Untuk aplikasi RAG, Anda dapat menggunakan Layanan AWS, seperti Amazon OpenSearch Service dan Amazon RDS for PostgreSQL, untuk menyimpan embeddings vektor yang menyediakan LLM dengan data internal Anda.

Alat-alat lainnya

- Azure DevOps membantu Anda mengelola CI/CD pipeline dan memfasilitasi pembuatan kode, pengujian, dan penerapan.
- Azure Data Lake Storage atau Snowflake adalah sumber data pelatihan pihak ketiga yang memungkinkan untuk model ML.
- Pinecone, Milvus, atau ChromaDB adalah database vektor pihak ketiga yang memungkinkan untuk menyimpan penyematan vektor.

Praktik terbaik

Sebelum menerapkan komponen apa pun dari MLOps alur kerja multicloud ini, selesaikan aktivitas berikut:

- Tentukan dan pahami alur kerja pembelajaran mesin dan alat yang diperlukan untuk mendukungnya. Kasus penggunaan yang berbeda memerlukan alur kerja dan komponen yang berbeda. Misalnya, feature store mungkin diperlukan untuk penggunaan kembali fitur dan inferensi latensi rendah dalam kasus penggunaan personalisasi, tetapi mungkin tidak diperlukan untuk kasus penggunaan lainnya. Memahami alur kerja target, persyaratan kasus penggunaan, dan metode kolaborasi pilihan tim ilmu data diperlukan untuk berhasil menyesuaikan arsitektur.
- Buat pemisahan tanggung jawab yang jelas untuk setiap komponen arsitektur. Menyebarkan penyimpanan data di Azure Data Lake Storage, Snowflake, dan Amazon S3 dapat meningkatkan kompleksitas dan biaya. Jika memungkinkan, pilih mekanisme penyimpanan yang konsisten. Demikian pula, hindari menggunakan kombinasi layanan Azure dan AWS, atau kombinasi DevOps layanan Azure dan AWS ML.

- Pilih satu atau beberapa model dan kumpulan data yang ada untuk melakukan end-to-end pengujian alur kerja. MLOps Artefak uji harus mencerminkan kasus penggunaan nyata yang dikembangkan tim ilmu data ketika platform memasuki produksi.

Epik

Rancang MLOps arsitektur Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi sumber data.	Berdasarkan kasus penggunaan saat ini dan masa depan, sumber data yang tersedia, dan jenis data (seperti data rahasia), dokumentasikan sumber data yang perlu diintegrasikan dengan MLOps platform. Data dapat disimpan di Amazon S3, Penyimpanan Danau Data Azure, Kepingan Salju, atau sumber lainnya. Untuk beban kerja AI generatif, data mungkin juga menyertak an basis pengetahuan yang mendasari respons yang dihasilkan. Data ini disimpan sebagai embeddings vektor dalam database vektor. Buat rencana untuk menginteg rasikan sumber-sumber ini dengan platform Anda dan mengamankan akses ke sumber daya yang benar.	Insinyur data, Ilmuwan data, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih layanan yang berlaku.	Sesuaikan arsitektur dengan menambahkan atau menghapus layanan berdasarkan alur kerja yang diinginkan dari tim ilmu data, sumber data yang berlaku, dan arsitektur cloud yang ada. Misalnya, insinyur data dan ilmuwan data dapat melakukan pra-pemrosesan data dan rekayasa fitur di SageMaker AI, AWS Glue, atau Amazon EMR. Tidak mungkin ketiga layanan tersebut diperlukan.	Administrator AWS, Insinyur data, ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menganalisis persyaratan keamanan.	Kumpulkan dan dokumenta sikan persyaratan keamanan. Ini termasuk menentukan: • Tim atau insinyur mana yang dapat mengakses sumber data tertentu • Tim atau insinyur mana yang dapat mengakses model pondasi yang telah dilatih sebelumnya • Apakah tim diizinkan untuk mengakses kode dan model tim lain • Izin mana (jika ada) yang harus dimiliki anggota tim untuk akun non-penge mbangan • Langkah-langkah keamanan mana yang perlu diterapka n untuk transfer data lintas cloud Untuk informasi lebih lanjut tentang mengamankan beban kerja AI generatif, lihat Mengamankan AI generatif: Pengantar Matriks Pelingkup an Keamanan AI Generatif (posting blog).AWS	Administrator AWS, arsitek Cloud

Mengatur AWS Organizations

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur AWS Organizations.	Siapkan AWS Organizations di root Akun AWS. Ini membantu Anda mengelola akun berikutnya yang Anda buat sebagai bagian dari MLOps strategi multi-akun. Lihat informasi yang lebih lengkap dalam dokumentasi AWS Organizations.	Administrator AWS

Mengatur lingkungan pengembangan dan pembuatan versi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat akun AWS pengembangan.	Buat Akun AWS tempat insinyur data dan ilmuwan data memiliki izin untuk bereksperimen dan membuat model ML. Untuk petunjuk, lihat Membuat akun anggota di organisasi Anda dalam AWS Organizations dokumentasi.	Administrator AWS
Buat repositori Model Build.	Buat repositori Git di Azure di mana ilmuwan data dapat mendorong pembuatan model dan kode penerapan mereka setelah fase eksperimen selesai. Untuk instruksi, lihat Mengatur repositori Git di dokumentasi DevOps Azure.	DevOps insinyur, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Model Deploy.	Buat repositori Git di Azure yang menyimpan kode dan templat penerapan standar. Ini harus menyertakan kode untuk setiap opsi penerapan yang digunakan organisasi, seperti yang diidentifikasi dalam fase desain. Misalnya, ini harus mencakup titik akhir waktu nyata, titik akhir asinkron, inferensi tanpa server, atau transformasi batch. Untuk instruksi, lihat Mengatur repositori Git di dokumentasi DevOps Azure.	DevOps insinyur, insinyur ML
Buat repositori Amazon ECR.	Siapkan repositori Amazon ECR yang menyimpan lingkungan ML yang disetujui sebagai gambar Docker. Izinkan ilmuwan data dan insinyur ML untuk menentukan lingkungan baru. Untuk petunjuknya, lihat Membuat repositori pribadi di dokumentasi Amazon ECR.	Insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan SageMaker AI Studio.	Siapkan SageMaker AI Studio di akun pengembangan sesuai dengan persyaratan keamanan yang ditentukan sebelumnya, alat ilmu data pilihan (seperti MLflow), dan lingkungan pengembangan terintegrasi (IDE) yang disukai. Gunakan konfigurasi siklus hidup untuk mengotoma tiskan pemasangan fungsiona litas utama dan menciptak an lingkungan pengemban gan yang seragam bagi ilmuwan data. Untuk informasi selengkapnya, lihat Amazon SageMaker AI Studio dan server MLflow pelacakan di dokumentasi SageMaker AI.	Ilmuwan data, insinyur ML, insinyur Prompt

Integrasikan CI/CD jaringan pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat akun Otomasi.	Buat Akun AWS tempat jaringan pipa dan pekerjaan otomatis berjalan. Anda dapat memberikan akses baca tim ilmu data ke akun ini. Untuk petunjuk, lihat Membuat akun anggota di organisasi Anda dalam AWS Organizations dokumentasi.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan registri model.	Siapkan Registri Model SageMaker AI di akun Otomasi. Registri ini menyimpan metadata untuk model ML dan membantu ilmuwan data atau tim tertentu untuk menyetujui atau menolak model. Untuk informasi selengkapnya, lihat Mendaftarkan dan menerapkan model dengan Model Registry di dokumentasi SageMaker AI.	Insinyur ML
Buat pipeline Model Build.	Buat CI/CD pipeline di Azure yang dimulai secara manual atau otomatis saat kode didorong ke repositori Model Build. Pipeline harus memeriksa kode sumber dan membuat atau memperbarui pipa SageMaker AI di akun Otomasi. Pipeline harus menambahkan model baru ke registri model. Untuk informasi selengkapnya tentang membuat pipeline, lihat dokumentasi Azure Pipelines.	DevOps insinyur, insinyur ML

Bangun tumpukan penerapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AWS akun pementasan dan penyebaran.	Buat Akun AWS untuk pementasan dan penerapan model ML. Akun ini harus identik untuk memungkinkan pengujian model yang akurat dalam pementasan sebelum pindah ke produksi. Anda dapat memberikan akses baca tim ilmu data ke akun pementasan. Untuk petunjuk, lihat Membuat akun anggota di organisasi Anda dalam AWS Organizations dokumentasi.	Administrator AWS
Siapkan bucket S3 untuk pemantauan model.	Selesaikan langkah ini jika Anda ingin mengaktifkan pemantauan model untuk model yang diterapkan yang dibuat oleh pipeline Model Deploy. Buat bucket Amazon S3 untuk menyimpan data input dan output. Untuk informasi selengkapnya tentang membuat bucket S3, lihat Membuat bucket di dokumentasi Amazon S3. Siapkan izin lintas akun sehingga pekerjaan pemantauan model otomatis berjalan di akun Otomasi. Untuk informasi selengkap nya, lihat Memantau data	Insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan kualitas model dalam dokumentasi SageMaker AI.	
Buat pipa Model Deploy.	Buat CI/CD pipeline di Azure yang dimulai saat model disetujui di registri model. Pipeline harus memeriksa kode sumber dan artefak model, membangun templat infrastruktur untuk menerapkan model di akun pementasan dan produksi, menyebarkan model di akun pementasan, menjalankan pengujian otomatis, menunggu persetujuan manual, dan menerapkan model yang disetujui ke dalam akun produksi. Untuk informasi selengkapnya tentang membuat pipeline, lihat dokumentasi Azure Pipelines.	DevOps insinyur, insinyur ML

(Opsional) Otomatiskan infrastruktur lingkungan ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bangun AWS CDK atau CloudFormation templat.	Tentukan AWS Cloud Development Kit (AWS CDK) atau AWS CloudFormation templat untuk semua lingkungan yang perlu digunakan secara otomatis. Ini mungkin termasuk lingkungan pengembangan, lingkunga	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n otomatisasi, dan lingkunga n pementasan dan penyebara n. Untuk informasi lebih lanjut, lihat AWS CDKdan CloudForm ationdokumentasi.	
Buat pipa Infrastruktur.	Buat CI/CD pipeline di Azure untuk penerapan infrastruktur. Administrator dapat memulai pipeline ini untuk membuat yang baru Akun AWS dan menyiapkan lingkungan yang dibutuhkan tim ML.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Pemantauan dan deteksi drift yang tidak memadai - Pemantauan yang tidak memadai dapat menyebabkan deteksi masalah kinerja model atau penyimpangan data yang tidak memadai.	Perkuat kerangka kerja pemantauan dengan alat seperti Amazon CloudWatch, SageMaker AI Model Monitor, dan SageMaker AI Clarify. Konfigurasikan peringatan untuk tindakan segera pada masalah yang diidentifikasi.
Kesalahan pemicu pipa CI — Pipa CI di Azure DevOps mungkin tidak dipicu saat penggabun gan kode karena kesalahan konfigurasi.	Periksa pengaturan DevOps proyek Azure untuk memastikan bahwa webhook diatur dengan benar dan menunjuk ke titik akhir AI yang benar. SageMaker
Tata Kelola — Akun Otomasi pusat mungkin tidak menerapkan praktik terbaik di seluruh platform ML, yang mengarah ke alur kerja yang tidak konsisten.	Audit pengaturan akun Automation, memastika n bahwa semua lingkungan dan model ML sesuai dengan praktik dan kebijakan terbaik yang telah ditentukan sebelumnya.

Isu	Solusi
Penundaan persetujuan registri model — Ini terjadi ketika ada penundaan dalam memeriksa dan menyetujui model, baik karena orang meluangkan waktu untuk memeriksanya atau karena masalah teknis.	Menerapkan sistem notifikasi untuk mengingat kan pemangku kepentingan model yang menunggu persetujuan, dan merampingkan proses peninjauan.
Kegagalan peristiwa penerapan model — Peristiwa yang dikirim untuk memulai pipeline penerapan model mungkin gagal, menyebabk an penundaan penerapan.	Konfirmasikan bahwa Amazon EventBridge memiliki izin dan pola peristiwa yang benar untuk menjalankan pipeline Azure DevOps dengan sukses.
Kemacetan penyebaran produksi — Proses persetujuan manual dapat menciptakan kemacetan, menunda penyebaran model produksi.	Optimalkan alur kerja persetujuan dalam pipeline penerapan model, mempromosikan tinjauan tepat waktu dan saluran komunikasi yang jelas.

Sumber daya terkait

AWS dokumentasi

- Dokumentasi Amazon SageMaker AI
- Machine Learning Lens (AWS Well Architected Framework)
- Perencanaan untuk sukses MLOps (Panduan AWS Preskriptif)

AWS Sumber daya lainnya

- MLOps peta jalan dasar untuk perusahaan dengan Amazon SageMaker AI (posting AWS blog)
- AWS Summit ANZ 2022 - End-to-end MLOps untuk arsitek (video) YouTube
- FMOps/LLMOps: Operasionalkan AI generatif dan perbedaan dengan MLOps (AWS posting blog)
- Operasionalkan Evaluasi LLM pada Skala menggunakan Amazon SageMaker AI Clarify dan MLOps layanan (posting blog)AWS
- Peran database vektor dalam aplikasi AI generatif (AWS posting blog)

Dokumentasi Azure

- Dokumentasi Azure DevOps
- Dokumentasi Azure Pipelines

Konfigurasikan pencatatan pemanggilan model di Amazon Bedrock dengan menggunakan AWS CloudFormation

Vikramaditya Bhatnagar, Amazon Web Services

Ringkasan

Anda dapat mengonfigurasi Amazon Bedrock untuk mengumpulkan log pemanggilan, data input model, dan data keluaran model untuk semua pemanggilan model di file Anda. Akun AWS Ini adalah praktik terbaik untuk membangun aplikasi AI generatif yang kuat dengan Amazon Bedrock. Anda dapat menyimpan log pemanggilan model di grup CloudWatch log Amazon Logs, di bucket Amazon Simple Storage Service (Amazon S3), atau di keduanya. Memiliki data CloudWatch log di Log membantu Anda membuat filter metrik kustom, alarm, dan dasbor. Amazon S3 sangat ideal untuk mereplikasi data di seluruh Wilayah AWS atau untuk penyimpanan jangka panjang, sebagaimana diatur oleh kebijakan organisasi Anda.

Pola ini menyediakan contoh AWS CloudFormation template yang menggunakan pendekatan infrastruktur sebagai kode (IAc) untuk mengonfigurasi pencatatan pemanggilan model untuk Amazon Bedrock. Template mengonfigurasi penyimpanan log di CloudWatch Log dan Amazon S3.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Izin berikut:
  - Izin untuk membuat tumpukan CloudFormation
  - Izin untuk mengakses Amazon Bedrock
  - Izin untuk membuat dan mengakses bucket Amazon S3
  - Izin untuk membuat dan mengakses grup CloudWatch log Log
  - Izin untuk membuat dan mengakses fungsi AWS Lambda
  - Izin untuk membuat dan mengakses AWS Key Management Service (AWS KMS) kunci

Batasan

Pola ini mencatat pemanggilan model ke CloudWatch Log dan Amazon S3. Itu tidak mendukung memilih hanya satu dari dua layanan ini.

Arsitektur

Arsitektur target

CloudFormation Template menyediakan sumber daya berikut dalam target Anda Akun AWS:

- Grup CloudWatch log Log untuk menyimpan log pemanggilan model
- Bucket Amazon S3 untuk menyimpan log pemanggilan model dan kebijakan bucket terkait
- Bucket Amazon S3 untuk menyimpan log akses sisi server dan kebijakan bucket terkait
- AWS Lambda Fungsi yang mengonfigurasi pengaturan logging di Amazon Bedrock
- Sebuah AWS KMS key dan alias kunci yang sesuai
- Peran layanan AWS Identity and Access Management (IAM) untuk Amazon Bedrock

Diagram berikut menunjukkan bagaimana log pemanggilan disimpan setelah Anda menyebarkan CloudFormation tumpukan yang terkait dengan pola ini. Amazon Bedrock menerbitkan data log saat model foundation mengirimkan teks, gambar, video, atau data penyematan. Seperti yang ditunjukkan pada diagram, bucket Amazon S3 dan grup CloudWatch log Log dienkripsi dengan file. AWS KMS key

Diagram menunjukkan alur kerja berikut:

1. Pengguna mengirimkan kueri ke model dasar di Amazon Bedrock.
2. Amazon Bedrock mengasumsikan peran layanan IAM.
3. Amazon Bedrock menghasilkan data log dan menyimpannya di grup CloudWatch log Log dan di bucket Amazon S3.
4. Jika pengguna membaca, mengunggah, atau menghapus file apa pun di bucket Amazon S3 yang berisi log pemanggilan model, aktivitas tersebut akan masuk ke bucket Amazon S3 lain untuk log akses sisi server.

Otomatisasi dan skala

Untuk menskalakan solusi ini, Anda dapat menerapkan CloudFormation template sebagai tumpukan yang disetel ke beberapa Wilayah AWS dan Akun AWS. Untuk informasi selengkapnya, lihat Mengelola tumpukan di seluruh akun dan Wilayah dengan StackSets CloudFormation dokumentasi.

Alat

Layanan AWS

- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari perusahaan AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri.

Alat-alat lainnya

- Git adalah sistem kontrol versi terdistribusi open source.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub enable-bedrock-logging-using-cloudformation.

Epik

Buat CloudFormation tumpukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh CloudFormation template.	Unduh CloudFormation template dari GitHub repositor i.	Arsitek awan
Menyebarkan template.	Buat tumpukan di akun target dan Wilayah Anda. Di bagian Parameter, tentukan nilai untuk parameter yang ditentukan dalam templat. Untuk petunjuk, lihat Membuat tumpukan dalam CloudForm ation dokumentasi.	Arsitek awan

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan akses model.	Di Amazon Bedrock, tambahkan akses ke model foundation. Untuk petunjukn ya, lihat Menambahkan atau menghapus akses ke model foundation Amazon Bedrock di dokumentasi Amazon Bedrock.	Arsitek awan
Jalankan prompt sampel.	Di taman bermain Amazon Bedrock, jalankan prompt sampel. Untuk petunjuknya, lihat Menghasilkan respons di konsol menggunakan taman	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bermain di dokumentasi Amazon Bedrock.	
Tinjau konfigurasi logging.	1. Masuk ke konsol Amazon Bedrock. 2. Di bilah navigasi, pilih Wilayah AWS tempat Anda menggunakan CloudFormation tumpukan. 3. Di panel navigasi kiri, di bawah Konfigurasi batuan dasar, pilih Pengaturan. 4. Konfirmasi hal berikut: • Pencatatan pemanggilan model diaktifkan. • Semua tipe data dipilih. • Untuk tujuan logging, Kedua S3 dan CloudWatch Log dipilih.	Arsitek awan
Tinjau ember Amazon S3.	1. Di bagian konfigurasi S3, pilih Browse S3. Ini membuka keranjang target di konsol Amazon S3. 2. Konfirmasikan bahwa data pencatatan hadir untuk prompt sampel yang sebelumnya Anda jalankan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau grup log.	1. Arahkan kembali ke halaman Pengaturan di konsol Amazon Bedrock. 2. Di bagian Konfigurasi CloudWatch Log, tinjau pengaturan untuk grup CloudWatch log Log. Catat nama grup log. 3. Buka konsol CloudWatch . 4. Di panel navigasi, di bawah Log, pilih Grup log. 5. Pilih nama grup log tempat Amazon Bedrock menerbitk an data log. 6. Konfirmasikan bahwa data pencatatan hadir untuk prompt sampel yang sebelumnya Anda jalankan.	Arsitek awan

Sumber daya terkait

AWS dokumentasi

- Mengakses bucket Amazon S3 (dokumentasi Amazon S3)
- Membuat dan mengelola tumpukan (CloudFormation dokumentasi)
- Memantau pemanggilan model (dokumentasi Amazon Bedrock)
- Bekerja dengan grup log dan aliran log (dokumentasi CloudWatch Log)

AWS posting blog

- Memantau aplikasi AI Generatif menggunakan Amazon Bedrock dan integrasi Amazon CloudWatch

- Praktik terbaik untuk membangun aplikasi AI generatif yang kuat dengan Amazon Bedrock Agents - Bagian 1
- Praktik terbaik untuk membangun aplikasi AI generatif yang kuat dengan Amazon Bedrock Agents - Bagian 2

Buat image container Docker kustom untuk SageMaker dan gunakan untuk pelatihan model di AWS Step Functions

Julia Bluszcz, Aubrey Oosthuizen, Mohan Gowda Purushothama, Neha Sharma, dan Mateusz Zaremba, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara membuat image container Docker untuk Amazon SageMaker dan menggunakannya untuk model pelatihan di AWS Step Functions. Dengan mengemas algoritme kustom dalam wadah, Anda dapat menjalankan hampir semua kode di SageMaker lingkungan, terlepas dari bahasa pemrograman, kerangka kerja, atau dependensi.

Dalam contoh SageMaker notebook yang disediakan, image container Docker kustom disimpan di Amazon Elastic Container Registry (Amazon ECR) Registry ECR). Step Functions kemudian menggunakan wadah yang disimpan di Amazon ECR untuk menjalankan skrip pemrosesan Python. SageMaker Kemudian, wadah mengekspor model ke Amazon Simple Storage Service (Amazon S3).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Peran AWS Identity and Access Management (IAM) SageMaker dengan izin Amazon S3
- Peran IAM untuk Step Functions
- Keakraban dengan Python
- Keakraban dengan Amazon SageMaker Python SDK
- Keakraban dengan AWS Command Line Interface (AWS CLI)
- Keakraban dengan AWS SDK for Python (Boto3)
- Keakraban dengan Amazon ECR
- Keakraban dengan Docker

Versi produk

- AWS Step Functions Data Science SDK versi 2.3.0
- Amazon SageMaker Python SDK versi 2.78.0

Arsitektur

Diagram berikut menunjukkan contoh alur kerja untuk membuat image container Docker untuk SageMaker, kemudian menggunakannya untuk model pelatihan di Step Functions:

Diagram menunjukkan alur kerja berikut:

1. Seorang ilmuwan atau DevOps insinyur data menggunakan SageMaker notebook Amazon untuk membuat gambar kontainer Docker kustom.
2. Seorang ilmuwan atau DevOps insinyur data menyimpan gambar kontainer Docker di repositori pribadi Amazon ECR yang ada di registri pribadi.
3. Seorang ilmuwan atau DevOps insinyur data menggunakan wadah Docker untuk menjalankan pekerjaan pemrosesan SageMaker Python dalam alur kerja Step Functions.

Otomatisasi dan skala

Contoh SageMaker notebook dalam pola ini menggunakan jenis instance `ml.m5.xlarge` notebook. Anda dapat mengubah jenis instance agar sesuai dengan kasus penggunaan Anda. Untuk informasi selengkapnya tentang jenis instans SageMaker notebook, lihat SageMaker Harga Amazon.

Alat

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon SageMaker adalah layanan pembelajaran mesin terkelola (ML) yang membantu Anda membuat dan melatih model ML, lalu menerapkannya ke lingkungan host yang siap produksi.
- Amazon SageMaker Python SDK adalah pustaka sumber terbuka untuk melatih dan menerapkan model pembelajaran mesin. SageMaker
- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi AWS Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis.
- AWS Step Functions Data Science Python SDK adalah pustaka open source yang membantu Anda membuat alur kerja Step Functions yang memproses dan menerbitkan model pembelajaran mesin.

Epik

Buat gambar kontainer Docker khusus dan simpan di Amazon ECR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Amazon ECR dan buat registri pribadi baru.	Jika Anda belum melakukan nya, siapkan Amazon ECR dengan mengikuti petunjuk dalam Menyiapkan dengan Amazon ECR di Panduan Pengguna Amazon ECR. Setiap akun AWS dilengkapi dengan registri ECR Amazon pribadi default.	DevOps insinyur
Buat repositori pribadi Amazon ECR.	Ikuti petunjuk dalam Membuat repositori pribadi di Panduan Pengguna Amazon ECR. **Note** Repositori yang Anda buat adalah tempat Anda akan menyimpan gambar kontainer Docker kustom Anda.	DevOps insinyur
Buat Dockerfile yang menyertakan spesifikasi yang diperlukan untuk menjalank an pekerjaan SageMaker pemrosesan Anda.	Buat Dockerfile yang menyertakan spesifika si yang diperlukan untuk menjalankan pekerjaan SageMaker pemrosesan Anda dengan mengonfig urasi Dockerfile. Untuk petunjuknya, lihat Mengadapt	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	asi wadah pelatihan Anda sendiri di Panduan SageMaker Pengembang Amazon.
	
	Untuk informasi selengkap nya tentang Dockerfiles, lihat Referensi Dockerfile dalam dokumentasi Docker.
	
	Contoh sel kode notebook Jupyter untuk membuat Dockerfile
	
	Sel 1
	
	`# Make docker folder`
	`!mkdir -p docker`
	
	Sel 2
	
	`%%writefile docker/Do`
	`ckerfile`
	
	`FROM python:3.7-slim-bu`
	`ster`
	
	`RUN pip3 install`
	` pandas==0.25.3 scikit-`
	`learn==0.21.3`
	`ENV PYTHONUNBUFFERED=T`
	`RUE`
	
	`ENTRYPOINT ["python3"]`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gambar kontainer Docker Anda dan dorong ke Amazon ECR.	1. Buat image container menggunakan Dockerfile yang Anda buat dengan menjalankan `docker build` perintah di AWS CLI. 2. Dorong gambar kontainer ke Amazon ECR dengan menjalankan `docker push` perintah. Untuk informasi selengkap nya, lihat Membangun dan mendaftarkan kontainer di Membangun wadah algoritme Anda sendiri GitHub. Contoh sel kode notebook Jupyter untuk membangun dan mendaftarkan gambar Docker ⚠ Important Sebelum menjalankan sel berikut, pastikan Anda telah membuat Dockerfile dan menyimpannya di direktori yang disebut. `docker` Selain itu, pastikan Anda telah membuat repositor i Amazon ECR, dan Anda mengganti	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`ecr_repository` nilai di sel pertama dengan nama repositori Anda.
	
	Sel 1
	
	`import boto3`
	`tag = ':latest'`
	`account_id = boto3.client('sts').get_caller_identity().get('Account')`
	`region = boto3.Session().region_name`
	`ecr_repository = 'byoc'`
	
	`image_uri = '{}.dkr.ecr.{}.amazonaws.com/{}'.format(account_id, region, ecr_repository + tag)`
	
	Sel 2
	
	`# Build docker image`
	`!docker build -t $image_uri docker`
	
	Sel 3
	
	`# Authenticate to ECR`
	`!aws ecr get-login-password --region {region} | docker login --username AWS --password-stdin`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`{account_id}.dkr.ecr.{region}.amazonaws.com`
	
	Sel 4
	
	`# Push docker image`
	`!docker push $image_uri`
	
	Note
	Anda harus mengautentikasi klien Docker Anda ke registri pribadi Anda sehingga Anda dapat menggunakan perintah `docker push` dan `docker pull`. Perintah ini mendorong dan menarik gambar ke dan dari repositori di registri Anda.	

Buat alur kerja Step Functions yang menggunakan gambar kontainer Docker kustom Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skrip Python yang mencakup pemrosesan kustom dan logika pelatihan model Anda.	Tulis logika pemrosesan khusus untuk dijalankan dalam skrip pemrosesan data Anda. Kemudian,	Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	simpan sebagai skrip Python bernama. `training.py`
	
	Untuk informasi selengkap nya, lihat Membawa model Anda sendiri dengan Mode SageMaker Skrip aktif GitHub.
	
	Contoh skrip Python yang mencakup pemrosesan kustom dan logika pelatihan model
	
	`%%writefile training.py`
	`from numpy import empty`
	`import pandas as pd`
	`import os`
	`from sklearn import datasets, svm`
	`from joblib import dump, load`
	
	`if __name__ == '__main__':`
	`digits = datasets.load_digits()`
	`#create classifier object`
	`clf = svm.SVC(gamma=0.001, C=100.)`
	
	`#fit the model`
	`clf.fit(digits.data[:-1], digits.target[:-1])`
	
	`#model output in binary format`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`output_path = os.path.join('/opt/ml/processing/model', "model.joblib") dump(clf, output_path)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat alur kerja Step Functions yang menyertakan pekerjaan SageMaker Processing Anda sebagai salah satu langkahnya.	Instal dan impor AWS Step Functions Data Science SDK dan unggah file training.py ke Amazon S3. Kemudian, gunakan Amazon SageMaker Python SDK untuk menentukan langkah pemrosesan di Step Functions.
	
	Important
	Pastikan Anda telah membuat peran eksekusi IAM untuk Step Functions di akun AWS Anda.
	
	Contoh pengaturan lingkungan dan skrip pelatihan khusus untuk diunggah ke Amazon S3
	
	`!pip install stepfunctions`
	
	`import boto3`
	`import stepfunctions`
	`import sagemaker`
	`import datetime`
	
	`from stepfunctions import steps`
	`from stepfunctions.inputs import ExecutionInput`
	`from stepfunctions.steps import (`
		Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Chain) from stepfunct ions.workflow import Workflow from sagemaker .processing import ScriptProcessor, ProcessingInput, ProcessingOutput sagemaker_session = sagemaker.Session() bucket = sagemaker _session.default_b ucket() role = sagemaker .get_execution_role() prefix = 'byoc-tra ining-model' # See prerequisites section to create this role workflow_execution_rol e = f"arn:aws:iam:: {account_id}:role/Ama zonSageMaker-StepF unctionsWorkflowEx ecutionRole" execution_input = ExecutionInput(schema={ "Preproce ssingJobName": str}) input_code = sagemaker _session.upload_data("training.py", bucket=bucket,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`key_prefix="preprocessing.py",` `)`	
	Contoh definisi langkah SageMaker pemrosesan yang menggunakan gambar Amazon ECR kustom dan skrip Python	
	Note Pastikan Anda menggunakan `execution_input` parameter untuk menentukan nama pekerjaan. Nilai parameter harus unik setiap kali pekerjaan berjalan. Juga, kode file training.py diteruskan sebagai `input` parameter ke`ProcessingStep` , yang berarti bahwa itu akan disalin di dalam wadah. Tujuan untuk `ProcessingInput` kode adalah sama dengan argumen kedua di dalam`container_entrypoint` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`script_processor = ScriptProcessor(command=['python3'], image_uri=image_uri, role=role, instance_count=1, instance_type='ml.m5.xlarge')`
	
	`processing_step = steps.ProcessingStep("training-step", processor=script_processor, job_name=execution_input["PreprocessingJobName"], inputs=[ProcessingInput(source=input_code, destination="/opt/ml/processing/input/code", input_name="code",),], outputs=[ProcessingOutput(source='/opt/ml/processing/model',`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>destinati
	on="s3://{}/{}".fo
	rmat(bucket, prefix),
	output_na
	me='byoc-example')
	],
	container_entrypoi
	nt=["python3", "/opt/
	ml/processing/input/c
	ode/training.py"],
	)</pre>	

	Contoh alur kerja Step Functions yang menjalankan pekerjaan SageMaker pemrosesan
	
	Note
	Contoh alur kerja ini mencakup langkah pekerjaan SageMaker pemrosesan saja, bukan alur kerja Step Functions yang lengkap. Untuk contoh alur kerja lengkap, lihat Contoh notebook SageMaker dalam dokumentasi AWS Step Functions Data Science SDK.
	
	<pre>workflow_graph =
	Chain([processing_
	step])</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>workflow = Workflow(
	    name="ProcessingWo
	rkflow",
	    definition=workflo
	w_graph,
	    role=workflow_exec
	ution_role
	)
	
	workflow.create()
	# Execute workflow
	execution = workflow.
	execute(
	    inputs={
	        "Preproce
	ssingJobName":
	 str(datetime.datet
	ime.now().strftime
	("%Y%m%d%H%M-%SS")),
	  # Each pre processin
	g job (SageMaker
	 processing job)
	 requires a unique name,
	    }
	)
	execution_output =
	 execution.get_outp
	ut(wait=True)</pre>	

Sumber daya terkait

- Memproses data (Panduan SageMaker Pengembang Amazon)
- Mengadaptasi wadah pelatihan Anda sendiri (Panduan SageMaker Pengembang Amazon)

Gunakan agen Amazon Bedrock untuk mengotomatiskan pembuatan kontrol entri akses di Amazon EKS melalui petunjuk berbasis teks

Keshav Ganesh dan Sudhanshu Saurav, Amazon Web Services

Ringkasan

Organizations menghadapi tantangan dalam mengelola kontrol akses dan penyediaan sumber daya ketika beberapa tim perlu bekerja dengan cluster Amazon Elastic Kubernetes Service (Amazon EKS) bersama. Layanan Kubernetes yang dikelola seperti Amazon EKS telah menyederhanakan operasi klaster. Namun, overhead administratif untuk mengelola akses tim dan izin sumber daya tetap kompleks dan memakan waktu.

Pola ini menunjukkan bagaimana agen Amazon Bedrock dapat membantu Anda mengotomatiskan manajemen akses klaster Amazon EKS. Otomatisasi ini memungkinkan tim pengembangan untuk fokus pada pengembangan aplikasi inti mereka daripada berurusan dengan pengaturan dan manajemen kontrol akses. Anda dapat menyesuaikan agen Amazon Bedrock untuk melakukan tindakan untuk berbagai tugas melalui petunjuk bahasa alami yang sederhana.

Dengan menggunakan AWS Lambda fungsi sebagai grup tindakan, agen Amazon Bedrock dapat menangani tugas seperti membuat entri akses pengguna dan mengelola kebijakan akses. Selain itu, agen Amazon Bedrock dapat mengonfigurasi asosiasi identitas pod yang memungkinkan akses ke sumber daya AWS Identity and Access Management (IAM) untuk pod yang berjalan di klaster. Dengan menggunakan solusi ini, organisasi dapat merampingkan administrasi klaster Amazon EKS mereka dengan petunjuk berbasis teks sederhana, mengurangi overhead manual, dan meningkatkan efisiensi pengembangan secara keseluruhan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Menetapkan peran dan izin IAM untuk proses penerapan. Ini termasuk izin untuk mengakses Amazon Bedrock foundation models (FM), membuat fungsi Lambda, dan sumber daya lain yang diperlukan di seluruh target. Akun AWS
- Akses diaktifkan di aktif Akun AWS ke Amazon Bedrock ini FMs: Amazon Titan Text Embeddings V2 dan Anthropic Claude 3 Haiku.

- AWS Command Line Interface (AWS CLI) versi 2.9.11 atau yang lebih baru, diinstal dan dikonfigurasi.
- eksctl 0.194.0 atau yang lebih baru, diinstal.

Batasan

- Pelatihan dan dokumentasi mungkin diperlukan untuk membantu memastikan kelancaran adopsi dan penggunaan teknik ini secara efektif. Menggunakan Amazon Bedrock, Amazon EKS, Lambda, OpenSearch Amazon Service, dan OpenAPI melibatkan kurva pembelajaran yang signifikan bagi pengembang dan tim. DevOps
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat layanan AWS menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Solusi ini melakukan langkah-langkah berikut:

1. Pengguna berinteraksi dengan agen Amazon Bedrock dengan mengirimkan prompt atau kueri yang berfungsi sebagai masukan bagi agen untuk memproses dan mengambil tindakan.
2. Berdasarkan prompt, agen Amazon Bedrock memeriksa skema OpenAPI untuk mengidentifikasi API yang benar untuk ditargetkan. Jika agen Amazon Bedrock menemukan panggilan API yang benar, permintaan akan masuk ke grup tindakan yang terkait dengan fungsi Lambda yang mengimplementasikan tindakan ini.
3. Jika API yang relevan tidak ditemukan, agen Amazon Bedrock akan menanyakan koleksi tersebut OpenSearch . OpenSearch Koleksinya menggunakan konten basis pengetahuan terindeks yang bersumber dari bucket Amazon S3 yang berisi Panduan Pengguna Amazon EKS.
4. OpenSearch Koleksi mengembalikan informasi kontekstual yang relevan ke agen Amazon Bedrock.
5. Untuk permintaan yang dapat ditindaklanjuti (yang cocok dengan operasi API), agen Amazon Bedrock mengeksekusi dalam cloud pribadi virtual (VPC) dan memicu fungsi Lambda.
6. Fungsi Lambda melakukan tindakan yang didasarkan pada input pengguna di dalam klaster Amazon EKS.

7. Bucket Amazon S3 untuk kode Lambda menyimpan artefak yang memiliki kode dan logika yang ditulis untuk fungsi Lambda.

Alat

Layanan AWS

- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon OpenSearch Service adalah layanan terkelola yang membantu Anda menyebarkan, mengoperasikan, dan menskalakan OpenSearch klaster di. AWS Cloud Fitur koleksinya membantu Anda mengatur data dan membangun basis pengetahuan komprehensif yang dapat digunakan asisten AI seperti agen Amazon Bedrock.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat lainnya

- eksctl adalah utilitas baris perintah untuk membuat dan mengelola cluster Kubernetes di Amazon EKS.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub eks-access-controls-bedrock-agent.

Praktik terbaik

- Pertahankan keamanan setinggi mungkin saat menerapkan pola ini. Pastikan klaster Amazon EKS bersifat pribadi, memiliki izin akses terbatas, dan semua sumber daya berada di dalam cloud pribadi virtual (VPC). Untuk informasi tambahan, lihat Praktik terbaik untuk keamanan dalam dokumentasi Amazon EKS.
- Gunakan kunci terkelola AWS KMS pelanggan sedapat mungkin, dan berikan izin akses terbatas kepada mereka.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori pola ini, jalankan perintah berikut di workstation lokal Anda: `git clone https://github.com/aws-samples/eks-access-controls-bedrock-agent.git`	AWS DevOps
Dapatkan Akun AWS ID.	Untuk mendapatkan Akun AWS ID, gunakan langkah-langkah berikut: 1. Buka shell di folder root repo kloning,. `eks-acces`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`s-controls-bedrock-agent`
	2. Untuk mendapatkan Akun AWS ID Anda, navigasikan ke direktori kloning dan jalankan perintah berikut:
	`AWS_ACCOUNT=$(aws sts get-caller-identity --query "Account" --output text)`
	Perintah ini menyimpan Akun AWS ID Anda dalam `AWS_ACCOUNT` variabel.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat bucket S3 untuk kode Lambda.	Untuk menerapkan solusi ini, Anda harus membuat tiga bucket Amazon S3 yang melayani tujuan berbeda, seperti yang ditunjukkan pada diagram arsitektur. Bucket S3 adalah untuk kode Lambda, basis pengetahuan, dan skema OpenAPI.
	
	Untuk membuat bucket kode Lambda, gunakan langkah-l angkah berikut:
	
	1. Untuk membuat bucket S3 untuk kode Lambda, jalankan perintah berikut:
	
	`aws s3 mb s3://bedrock-agent-lambda-artifacts-${AWS_ACCOUNT} --region us-east-1`
	
	2. Untuk menginstal ketergant ungan kode Lambda, jalankan perintah berikut:
	
	`cd eks-lambda`
	`npm install`
	`tsc`
	`cd .. && cd opensearch-lambda`
	`npm install`
	`tsc`
	`cd ..`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Untuk mengemas kode dan mengunggahnya ke bucket S3 untuk Lambda, jalankan perintah berikut: `aws cloudformation package \ --template-file eks-access-controls.yaml \ --s3-bucket bedrock-agent-lambda-artifacts-${AWS_ACCOUNT} \ --output-template-file eks-access-controls-template.yaml \ --region us-east-1` Perintah package membuat CloudFormation template baru (`eks-access-controls-template.yaml`) yang berisi: • Referensi ke kode fungsi Lambda yang disimpan di bucket S3 Anda. • Definisi untuk semua AWS infrastruktur yang diperlukan termasuk VPC, subnet, agen Amazon Bedrock, dan koleksi. OpenSearch Anda dapat menggunakan template ini untuk menyebarkan solusi lengkap	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan menggunakan CloudFormation.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat bucket S3 untuk basis pengetahuan.	Untuk membuat bucket Amazon S3 untuk basis pengetahuan, gunakan langkah-langkah berikut:
	1. Untuk membuat bucket Amazon S3 untuk basis pengetahuan, jalankan perintah berikut:
	`aws s3 mb s3://eks-knowledge-base-${AWS_ACCOUNT} --region us-east-1`
	2. Untuk mengunduh Panduan Pengguna Amazon EKS dan menyimpannya di direktori, jalankan perintah berikut:
	`mkdir dataSource`
	`cd dataSource`
	`curl https://docs.aws.amazon.com/pdfs/eks/latest/userguide/eks-ug.pdf -o eks-user-guide.pdf`
	3. Untuk mengunggah panduan pengguna ke bucket S3 yang Anda buat di langkah 1, jalankan perintah berikut:
	`aws s3 cp eks-user-guide.pdf s3://eks-`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`knowledge-base-${AWS_ACCOUNT} \`
	`--region us-east-1 \`
	4. Untuk kembali ke direktori root, jalankan perintah berikut:
	`cd ..`	

Buat bucket S3 untuk skema OpenAPI.	Untuk membuat bucket Amazon S3 untuk skema OpenAPI, gunakan langkah-langkah berikut:
	1. Untuk membuat bucket S3, jalankan perintah berikut:
	`aws s3 mb s3://eks-openapi-schema-${AWS_ACCOUNT} --region us-east-1`
	2. Untuk mengunggah skema OpenAPI ke bucket S3, jalankan perintah berikut:
	`aws s3 cp openapi-schema.yaml s3://eks-openapi-schema-${AWS_ACCOUNT} --region us-east-1`
		AWS DevOps

Menyebarkan tumpukan CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation tumpukan.	Untuk menyebarkan CloudFormation tumpukan, gunakan file CloudFormation template `eks-access-controls-template.yaml` yang Anda buat sebelumnya. Untuk petunjuk selengkapnya, lihat Membuat tumpukan dari CloudFormation konsol di CloudFormation dokumentasi. **Note** Penyediaan OpenSearch indeks dengan CloudFormation template membutuhkan waktu sekitar 10 menit. Setelah tumpukan dibuat, buat catatan `VPC_ID` dan `PRIVATE_SUBNET ID` s.	AWS DevOps
Buat cluster Amazon EKS.	Untuk membuat cluster Amazon EKS di dalam VPC, gunakan langkah-langkah berikut: 1. Buat salinan file `eks-config.yaml` konfigurasi, dan beri nama salinanny	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	a sebagai`eks-deploy.yaml` .
	2. Buka `eks-deploy.yaml` di editor teks. Kemudian, ganti nilai placeholder berikut dengan nilai dari tumpukan yang diterapkan:`VPC_ID`,, dan `PRIVATE_SUBNET1` `PRIVATE_SUBNET2`
	3. Untuk membuat cluster dengan menggunakan utilitas eksctl, jalankan perintah berikut:
	`eksctl create cluster -f eks-deploy.yaml`
	Note
	Proses pembuatan cluster ini bisa memakan waktu hingga 15-20 menit untuk menyelesaikannya.
	4. Untuk memverifikasi bahwa cluster berhasil dibuat, jalankan perintah berikut:
	`aws eks describe-cluster --name --query "cluster.status"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws eks update-ku beconfig --name -- region kubectl get nodes` Hasil yang diharapkan adalah sebagai berikut: • Status cluster adalah`ACTIVE`. • Perintah `kubectl get nodes` menunjukkan bahwa semua node dalam `Ready` keadaan.	

Hubungkan fungsi Lambda dan kluster Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat koneksi antara kluster Amazon EKS dan fungsi Lambda.	Untuk mengatur izin jaringan dan IAM agar fungsi Lambda dapat berkomunikasi dengan klaster Amazon EKS, gunakan langkah-langkah berikut: 1. Untuk mengidentifikasi peran IAM yang dilampirk an ke fungsi Lambda, buka Konsol Manajemen AWS dan temukan fungsi Lambda bernama. `bedrock-agent-eks-access-control` Catat Nama Sumber Daya	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Amazon (ARN) dari peran IAM.
	2. Untuk membuat entri akses di klaster Amazon EKS untuk peran IAM fungsi Lambda, jalankan perintah berikut:
	`aws eks create-access-entry --cluster-name eks-testing-cluster --principal-arn <principal-Role-ARN>`
	3. Untuk menetapkan `AmazonEKSClusterAdminPolicy` izin ke peran ini, jalankan perintah berikut:
	`aws eks associate-access-policy --cluster-name eks-testing-cluster --principal-arn <principal-Role-ARN> --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSClusterAdminPolicy --access-scope type=cluster`
	Untuk informasi selengkapnya, lihat Mengaitkan kebijakan akses dengan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	entri akses dan Amazon EKSCluster AdminPolicy di dokumentasi Amazon EKS. 4. Temukan grup keamanan klaster Amazon EKS. Tambahkan aturan masuk untuk mengizinkan lalu lintas jaringan masuk dari fungsi Lambda ke kluster Amazon EKS. Gunakan nilai berikut untuk aturan masuk: • Jenis - HTTPS • Rentang pelabuhan - 443 • Sumber - Grup keamanan Lambda Untuk informasi selengkapnya, lihat Mengonfigurasi aturan grup keamanan di dokumentasi Amazon VPC.	

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji agen Amazon Bedrock.	Sebelum menguji agen Amazon Bedrock, pastikan Anda melakukan hal berikut:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Uji dengan peran non-produksi terlebih dahulu. • Dokumentasikan setiap perubahan yang dilakukan pada akses cluster. • Miliki rencana untuk mengembalikan perubahan jika diperlukan. Untuk mengakses agen Amazon Bedrock, gunakan langkah-langkah berikut: 1. Masuk ke Konsol Manajemen AWS menggunakan peran IAM dengan izin Amazon Bedrock, dan buka konsol Amazon Bedrock di. https://console.aws.amazon.com/bedrock/ 2. Pilih Agen dari panel navigasi kiri. Kemudian, pilih agen yang dikonfigurasi di bagian Agen. 3. Untuk menguji agen, coba contoh petunjuk berikut yang Anda ganti `Principal-ARN-OF-ROLE` dengan ARN peran IAM yang sebenarnya: • Untuk membuat entri akses untuk setiap peran IAM	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang ingin Anda berikan akses ke kluster EKS, gunakan prompt berikut: `Create an access entry in cluster eks-testing-new for a role whose principal arn is <Principal-ARN-OF-ROLE> with access policy as AmazonEKSAdminPolicy` Hasil yang diharapkan: • Agen harus mengkonfirmasi pembuatan entri akses. • Untuk memverifikasi, periksa menggunakan Konsol Manajemen AWS atau menggunakan Amazon EKS API dan jalankan perintah berikut: `aws eks list-access-entries --cluster-name ekscluster` • Untuk menjelaskan entri akses yang Anda buat, gunakan prompt berikut: `Describe an access entry in cluster eks-testing-new whose principal arn is`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`<Principal-ARN-OF-ROLE>` Hasil yang diharapkan: • Agen harus mengembalikan detail tentang entri akses. • Detailnya harus sesuai dengan apa yang Anda konfigurasikan sebelumnya untuk entri akses. • Untuk menghapus entri akses yang Anda buat, gunakan prompt berikut: `Delete the access entry in cluster eks-testing-new whose principal arn is <Principal-ARN-OF-ROLE>` Hasil yang diharapkan: • Agen harus mengkonfirmasi penghapusan entri akses. • Untuk memverifikasi, periksa menggunakan Konsol Manajemen AWS atau menggunakan Amazon EKS API dan jalankan perintah berikut: `aws eks list-access-entries`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--cluster-name eksclusterAnda juga dapat meminta agen untuk melakukan tindakan untuk asosiasi EKS Pod Identity. Untuk detail selengkapnya, lihat Pelajari cara EKS Pod Identity memberikan akses Pod ke Layanan AWS dalam dokumentasi Amazon EKS.`	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Untuk membersihkan sumber daya yang dibuat pola ini, gunakan prosedur berikut. Tunggu setiap langkah penghapusan selesai sebelum melanjutkan ke langkah berikutnya. **Warning** Prosedur ini akan menghapus semua sumber daya yang dibuat oleh tumpukan ini secara permanen. Pastikan Anda telah mencadangkan data	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	penting apa pun sebelum melanjutkan.
	
	1. Untuk menghapus cluster Amazon EKS, jalankan perintah berikut:
	
	`eksctl delete cluster -f eks-deploy.yaml`
	
	Note
	Operasi ini bisa memakan waktu 15-20 menit untuk menyelesaikannya.
	
	2. Untuk menghapus bucket Amazon S3, jalankan perintah berikut:
	• Untuk mengosongkan ember Lambda:
	
	`aws s3 rm s3://bedrock-agent-lambda-artifacts-${AWS_ACCOUNT} --recursive`
	
	• Untuk mengosongkan ember basis pengetahuan:
	
	`aws s3 rm s3://eks-knowledge-base-`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>${AWS_ACCOUNT} –recursive</pre>
	• Untuk mengosongkan keranjang skema OpenAPI:
	<pre>aws s3 rm s3://bedrock-agent-openapi-schema-${AWS_ACCOUNT} –recursive</pre>
	• Untuk menghapus ember kosong:
	<pre>aws s3 rb s3://bedrock-agent-lambda-artifacts-${AWS_ACCOUNT}
	aws s3 rb s3://eks-knowledge-base-${AWS_ACCOUNT}
	aws s3 rb s3://bedrock-agent-openapi-schema-${AWS_ACCOUNT}</pre>
	3. Untuk menghapus CloudFormation tumpukan, jalankan perintah berikut:
	<pre>aws cloudformation delete-stack \ --stack-name</pre>
	4. Untuk memverifikasi penghapusan klaster Amazon EKS, jalankan perintah berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`eksctl get clusters`
	5. Untuk memverifikasi penghapusan bucket Amazon S3, jalankan perintah berikut:
	• Untuk memverifikasi penghapusan bucket Lambda:
	`aws s3 ls | grep "bedrock-agent-lambda-artifacts"`
	• Untuk memverifikasi penghapusan bucket basis pengetahuan:
	`aws s3 ls | grep "eks-knowledge-base"`
	• Untuk memverifikasi penghapusan bucket skema OpenAPI:
	`aws s3 ls | grep "bedrock-agent-openapi-schema"`
	6. Untuk memverifikasi penghapusan tumpukan, jalankan perintah berikut:
	`aws cloudformation list-stacks \--query`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`'StackSummaries[?StackName==``]'` Jika tumpukan gagal dihapus, lihat Pemecahan Masalah.	

Pemecahan masalah

Isu	Solusi
Kode kesalahan bukan nol dikembalikan selama pengaturan lingkungan.	Verifikasi bahwa Anda menggunakan folder yang benar saat menjalankan perintah apa pun untuk menerapkan solusi ini. Untuk informasi selengkapnya, lihat file First_Deploy.md di repositori pola ini.
Fungsi Lambda tidak dapat melakukan tugas.	Pastikan konektivitas diatur dengan benar dari fungsi Lambda ke cluster Amazon EKS.
Agen meminta tidak mengenali. APIs	Menerapkan kembali solusinya. Untuk informasi selengkapnya, lihat file RE_DEPLOY.md di repositori pola ini.
Tumpukan gagal dihapus.	Upaya awal untuk menghapus tumpukan mungkin gagal. Kegagalan ini dapat terjadi karena masalah ketergantungan dengan sumber daya khusus yang dibuat untuk OpenSearch koleksi yang melakukan pengindeksan untuk basis pengetahuan. Untuk menghapus tumpukan, coba lagi operasi hapus dengan mempertahankan sumber daya kustom.

Sumber daya terkait

AWS Blog

- Penyelaman mendalam ke kontrol manajemen akses Amazon EKS yang disederhanakan

Dokumentasi Amazon Bedrock

- Otomatiskan tugas dalam aplikasi Anda menggunakan agen AI
- Bagaimana Agen Amazon Bedrock bekerja
- Menguji dan memecahkan masalah perilaku agen
- Gunakan grup tindakan untuk menentukan tindakan yang harus dilakukan agen Anda

Dokumentasi Amazon EKS

- Pelajari cara kerja kontrol akses di Amazon EKS

Terapkan kasus penggunaan RAG AWS dengan menggunakan Terraform dan Amazon Bedrock

Martin Maritsch, Nicolas Jacob Baer, Olivier Brique, Julian Ferdinand Grueber, Alice Morano, dan Nicola D Orazio, Amazon Web Services

Ringkasan

AWS menyediakan berbagai opsi untuk membangun kasus penggunaan AI generatif yang diaktifkan Retrieval Augmented Generation (RAG). Pola ini memberi Anda solusi untuk aplikasi berbasis RAG berdasarkan LangChain dan kompatibel dengan Amazon Aurora PostgreSQL sebagai penyimpanan vektor. Anda dapat langsung menerapkan solusi ini dengan Terraform ke dalam Akun AWS dan menerapkan kasus penggunaan RAG sederhana berikut:

1. Pengguna mengunggah file secara manual ke bucket Amazon Simple Storage Service (Amazon S3), seperti file Microsoft Excel atau dokumen PDF. (Untuk informasi selengkapnya tentang jenis file yang didukung, lihat Dokumentasi tidak terstruktur.)
2. Isi file diekstraksi dan disematkan ke dalam database pengetahuan yang didasarkan pada Aurora PostgreSQL kompatibel tanpa server, yang mendukung konsumsi dokumen secara real-time ke dalam penyimpanan vektor. Pendekatan ini memungkinkan model RAG untuk mengakses dan mengambil informasi yang relevan untuk kasus penggunaan di mana latensi rendah penting.
3. Ketika pengguna terlibat dengan model pembuatan teks, itu meningkatkan interaksi melalui augmentasi pengambilan konten yang relevan dari file yang diunggah sebelumnya.

Pola ini menggunakan Amazon Titan Text Embeddings v2 sebagai model embedding dan Anthropic Claude 3 Soneta sebagai model pembuatan teks, keduanya tersedia di Amazon Bedrock.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi dengan Anda Akun AWS. Untuk petunjuk penginstalan, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLI dokumentasi. AWS CLI Untuk meninjau AWS kredensi Anda dan akses Anda ke akun Anda, lihat Pengaturan konfigurasi dan file kredensi dalam dokumentasi. AWS CLI

- Akses model yang diaktifkan untuk model bahasa besar yang diperlukan (LLMs) di konsol Amazon Bedrock Anda Akun AWS. Pola ini membutuhkan yang berikut LLMs:
  - `amazon.titan-embed-text-v2:0`
  - `anthropic.claude-3-sonnet-20240229-v1:0`

Batasan

- Contoh arsitektur ini tidak menyertakan antarmuka untuk menjawab pertanyaan terprogram dengan database vektor. Jika kasus penggunaan Anda memerlukan API, pertimbangkan untuk menambahkan Amazon API Gateway dengan AWS Lambda fungsi yang menjalankan tugas pengambilan dan penjawab pertanyaan.
- Contoh arsitektur ini tidak menyertakan fitur pemantauan untuk infrastruktur yang digunakan. Jika kasus penggunaan Anda memerlukan pemantauan, pertimbangkan untuk menambahkan layanan AWS pemantauan.
- Jika Anda mengunggah banyak dokumen dalam jangka waktu singkat ke bucket Amazon S3, fungsi Lambda mungkin mengalami batas tarif. Sebagai solusinya, Anda dapat memisahkan fungsi Lambda dengan antrian Amazon Simple Queue Service (Amazon SQS) di mana Anda dapat mengontrol laju pemanggilan Lambda.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- AWS CLI versi 2 atau yang lebih baru
- Docker versi 26.0.0 atau yang lebih baru
- Puisi versi 1.7.1 atau yang lebih baru
- Python versi 3.10 atau yang lebih baru
- Terraform versi 1.8.4 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Diagram ini menggambarkan hal berikut:

1. Saat objek dibuat di bucket Amazon S3`bedrock-rag-template-<account_id>`, notifikasi Amazon S3 akan memanggil fungsi Lambda. `data-ingestion-processor`
2. Fungsi Lambda `data-ingestion-processor` didasarkan pada image Docker yang disimpan di repositori Amazon Elastic Container Registry (Amazon ECR) Registry ECR). `bedrock-rag-template`

   Fungsi ini menggunakan LangChain S3 FileLoader untuk membaca file sebagai LangChain Dokumen. Kemudian, LangChain RecursiveCharacterTextSplitterpotongan setiap dokumen, diberi a `CHUNK_SIZE` dan a `CHUNK_OVERLAP` yang bergantung pada ukuran token maksimum model penyematan Amazon Titan Text Embedding V2. Selanjutnya, fungsi Lambda memanggil model embedding di Amazon Bedrock untuk menyematkan potongan ke dalam representasi vektor numerik. Terakhir, vektor-vektor ini disimpan dalam database Aurora PostgreSQL. Untuk mengakses database, fungsi Lambda pertama-tama mengambil nama pengguna dan kata sandi dari. AWS Secrets Manager
3. Pada instance notebook Amazon SageMaker AIaws-sample-bedrock-rag-template, pengguna dapat menulis prompt pertanyaan. Kode memanggil Claude 3 di Amazon Bedrock dan menambahkan informasi basis pengetahuan ke konteks prompt. Akibatnya, Claude 3 memberikan tanggapan menggunakan informasi dalam dokumen.

Pendekatan pola ini terhadap jaringan dan keamanan adalah sebagai berikut:

- Fungsi Lambda `data-ingestion-processor` ada di subnet pribadi dalam virtual private cloud (VPC). Fungsi Lambda tidak diizinkan untuk mengirim lalu lintas ke internet publik karena grup keamanannya. Akibatnya, lalu lintas ke Amazon S3 dan Amazon Bedrock dirutekan melalui titik akhir VPC saja. Akibatnya, lalu lintas tidak melintasi internet publik, yang mengurangi latensi dan menambahkan lapisan keamanan tambahan di tingkat jaringan.
- Semua sumber daya dan data dienkripsi kapan pun berlaku dengan menggunakan kunci AWS Key Management Service (AWS KMS) dengan alias. `aws-sample/bedrock-rag-template`

Otomatisasi dan skala

Pola ini menggunakan Terraform untuk menyebarkan infrastruktur dari repositori kode ke file. Akun AWS

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL. Dalam pola ini, Aurora PostgreSQL kompatibel menggunakan plugin pgvector sebagai database vektor.
- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal. Dalam pola ini, Amazon ECR meng-host image Docker untuk fungsi Lambda`data-ingestion-processor`.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan. Dalam pola ini, Lambda menyerap data ke dalam penyimpanan vektor.
- Amazon SageMaker AI adalah layanan pembelajaran mesin terkelola (ML) yang membantu Anda membuat dan melatih model ML, lalu menerapkannya ke lingkungan host yang siap produksi.
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur

yang dapat diskalakan. AWS VPC mencakup subnet dan tabel routing untuk mengontrol arus lalu lintas.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- Puisi adalah alat untuk manajemen ketergantungan dan pengemasan dengan Python.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub terraform-rag-template-using-amazon-bedrock.

Praktik terbaik

- Meskipun contoh kode ini dapat digunakan di mana pun Wilayah AWS, kami menyarankan Anda menggunakan US East (Virginia N.) — `us-east-1` atau US West (California N.) —. `us-west-1` Rekomendasi ini didasarkan pada ketersediaan fondasi dan model penyematan di Amazon Bedrock pada saat publikasi pola ini. Untuk up-to-date daftar dukungan model fondasi Amazon Bedrock di Wilayah AWS, lihat Dukungan model oleh Wilayah AWS dalam dokumentasi Amazon Bedrock. Untuk informasi tentang menerapkan contoh kode ini ke Wilayah lain, lihat Informasi tambahan.
- Pola ini hanya menyediakan proof-of-concept (PoC) atau demo percontohan. Jika Anda ingin membawa kode ke produksi, pastikan untuk menggunakan praktik terbaik berikut:
  - Aktifkan pencatatan akses server untuk Amazon S3.
  - Siapkan pemantauan dan peringatan untuk fungsi Lambda.
  - Jika kasus penggunaan Anda memerlukan API, pertimbangkan untuk menambahkan Amazon API Gateway dengan fungsi Lambda yang menjalankan tugas pengambilan dan penjawab pertanyaan.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Terapkan solusi dalam Akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning GitHub repositori yang disediakan dengan pola ini, gunakan perintah berikut: `git clone https://github.com/aws-samples/terraform-rag-template-using-amazon-bedrock`	AWS DevOps
Konfigurasikan variabel.	Untuk mengkonfigurasi parameter untuk pola ini, lakukan hal berikut: 1. Di komputer Anda, di GitHub repositori, gunakan perintah berikut untuk membuka folder`terraform` : `cd terraform` 2. Buka `commons.tfvars` file, dan sesuaikan parameter sesuai dengan kebutuhan Anda.	AWS DevOps
Menyebarkan solusinya.	Untuk menerapkan solusi, lakukan hal berikut: 1. Di `terraform` folder, gunakan perintah berikut	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	untuk menjalankan Terraform dan meneruska n variabel yang Anda sesuaikan:
	
	`terraform init`
	`terraform apply -var-file=commons.tfvars`
	
	2. Konfirmasikan bahwa sumber daya yang ditampilk an dalam diagram arsitektur berhasil digunakan.
	
	Penyebaran infrastruktur menyediakan instance SageMaker AI di dalam VPC dan dengan izin untuk mengakses database Aurora PostgreSQL.	

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan demo.	Setelah penerapan infrastru ktur sebelumnya berhasil, gunakan langkah-langkah berikut untuk menjalankan demo di notebook Jupyter: 1. Masuk ke Konsol Manajemen AWS Akun	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS tempat infrastruktur digunakan. 2. Buka instance notebook SageMaker AI `aws-sample-bedrock-rag-template` . 3. Pindahkan `rag_demo.ipynb` notebook Jupyter ke instance notebook SageMaker AI dengan menggunakan drag and drop. 4. Buka `rag_demo.ipynb` pada instance notebook SageMaker AI, dan pilih `conda_python3` kernel. 5. Untuk menjalankan demo, jalankan sel notebook. Notebook Jupyter memandu Anda melalui proses berikut: • Persyaratan pemasangan • Definisi penyematan • Koneksi basis data • Konsumsi data • Pengambilan generasi teks tambahan • Kueri dokumen yang relevan	

Bersihkan infrastur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan infrastruktur.	Untuk menghapus semua sumber daya yang Anda buat saat tidak lagi diperlukan, gunakan perintah berikut: `terraform destroy -var-file=commons.tfvars`	AWS DevOps

Sumber daya terkait

AWS sumber daya

- Membangun fungsi Lambda dengan Python
- Parameter inferensi untuk model pondasi
- Akses ke model fondasi Amazon Bedrock
- Peran database vektor dalam aplikasi AI generatif (Blog AWS Database)
- Bekerja dengan Amazon Aurora PostgreSQL

Sumber daya lainnya

- dokumentasi pgvector

Informasi tambahan

Menerapkan database vektor

Pola ini menggunakan Aurora PostgreSQL kompatibel untuk mengimplementasikan database vektor untuk RAG. Sebagai alternatif untuk Aurora PostgreSQL, AWS menyediakan kemampuan dan layanan lain untuk RAG, seperti Amazon Bedrock Knowledge Bases dan Amazon Service. OpenSearch Anda dapat memilih solusi yang paling sesuai dengan kebutuhan spesifik Anda:

- Amazon OpenSearch Service menyediakan mesin pencari dan analitik terdistribusi yang dapat Anda gunakan untuk menyimpan dan menanyakan data dalam jumlah besar.
- Amazon Bedrock Knowledge Bases dirancang untuk membangun dan menyebarkan basis pengetahuan sebagai abstraksi tambahan untuk menyederhanakan proses konsumsi dan pengambilan RAG. Pangkalan Pengetahuan Amazon Bedrock dapat bekerja dengan Aurora PostgreSQL dan Amazon Service. OpenSearch

Menyebarkan ke yang lain Wilayah AWS

Seperti yang dijelaskan dalam Arsitektur, kami menyarankan Anda menggunakan Region US East (Virginia N.) — `us-east-1` atau US West (California N.) — `us-west-1` untuk menyebarkan contoh kode ini. Namun, ada dua cara yang mungkin untuk menyebarkan contoh kode ini ke Wilayah selain `us-east-1` dan`us-west-1`. Anda dapat mengonfigurasi Wilayah penyebaran dalam `commons.tfvars` file. Untuk akses model pondasi lintas wilayah, pertimbangkan opsi berikut:

- Melintasi internet publik — Jika lalu lintas dapat melintasi internet publik, tambahkan gateway internet ke VPC. Kemudian, sesuaikan grup keamanan yang ditetapkan ke fungsi Lambda `data-ingestion-processor` dan instance notebook SageMaker AI untuk memungkinkan lalu lintas keluar ke internet publik.
- Tidak melintasi internet publik — Untuk menyebarkan sampel ini ke Wilayah mana pun selain `us-east-1` atau`us-west-1`, lakukan hal berikut:

1. Di `us-west-1` Wilayah `us-east-1` atau, buat VPC tambahan termasuk titik akhir VPC untuk. `bedrock-runtime`
2. Buat koneksi peering dengan menggunakan VPC peering atau gateway transit ke aplikasi VPC.
3. Saat mengonfigurasi klien `bedrock-runtime` boto3 dalam fungsi Lambda apa pun di luar `us-east-1` atau`us-west-1`, teruskan nama DNS pribadi dari titik akhir VPC untuk `bedrock-runtime` in `us-east-1` atau us-west-1 sebagai ke klien boto3. `endpoint_url`

Menerapkan logika preprocessing ke dalam model MLdalam satu titik akhir menggunakan pipeline inferensi di Amazon SageMaker

Mohan Gowda Purushothama, Gabriel Rodriguez Garcia, dan Mateusz Zaremba, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerapkan beberapa objek model pipeline dalam satu titik akhir dengan menggunakan pipeline inferensi di Amazon. SageMaker Objek model pipeline mewakili tahapan alur kerja machine learning (ML) yang berbeda, seperti preprocessing, inferensi model, dan postprocessing. Untuk mengilustrasikan penerapan objek model pipeline yang terhubung secara serial, pola ini menunjukkan kepada Anda cara menerapkan wadah Scikit-learn pra-pemrosesan dan model regresi berdasarkan algoritme pembelajar linier yang ada di dalamnya. SageMaker Penerapan di-host di belakang satu titik akhir di. SageMaker

> **Note**
>
> Penerapan dalam pola ini menggunakan tipe instance ml.m4.2xlarge. Sebaiknya gunakan jenis instans yang sesuai dengan persyaratan ukuran data dan kompleksitas alur kerja Anda. Untuk informasi selengkapnya, lihat SageMaker Harga Amazon. Pola ini menggunakan gambar Docker bawaan untuk Scikit-learn, tetapi Anda dapat menggunakan wadah Docker Anda sendiri dan mengintegrasikannya ke dalam alur kerja Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Python 3.9
- Perpustakaan Amazon SageMaker Python SDK dan Boto3
- Peran AWS Identity and Access Management (AWS IAM) dengan SageMaker izin dasar dan izin Amazon Simple Storage Service (Amazon S3)

Versi produk

- Amazon SageMaker Python SDK 2.49.2

Arsitektur

Tumpukan teknologi target

- Amazon Elastic Container Registry (Amazon ECR)
- Amazon SageMaker
- SageMaker Studio Amazon
- Amazon Simple Storage Service (Amazon S3)
- Titik akhir inferensi waktu nyata untuk Amazon SageMaker

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk penyebaran objek model SageMaker pipa Amazon.

Diagram menunjukkan alur kerja berikut:

1. SageMaker Notebook menyebarkan model pipa.
2. Ember S3 menyimpan artefak model.
3. Amazon ECR mendapatkan gambar wadah sumber dari bucket S3.

Alat

Alat AWS

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon SageMaker adalah layanan ML terkelola yang membantu Anda membuat dan melatih model ML, lalu menerapkannya ke lingkungan host yang siap produksi.
- Amazon SageMaker Studio adalah lingkungan pengembangan terintegrasi (IDE) berbasis web untuk ML yang memungkinkan Anda membuat, melatih, men-debug, menerapkan, dan memantau model ML Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Kode

Kode untuk pola ini tersedia di GitHub Inference Pipeline dengan repositori Scikit-learn dan Linear Learner.

Epik

Siapkan dataset

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan dataset untuk tugas regresi Anda.	Buka buku catatan di Amazon SageMaker Studio.
	
	Untuk mengimpor semua pustaka yang diperlukan dan menginisialisasi lingkungan kerja Anda, gunakan kode contoh berikut di buku catatan Anda:
	
	`import sagemaker`
	`from sagemaker import get_execution_role`
	
	`sagemaker_session = sagemaker.Session()`
	
	`# Get a SageMaker-compatible role used by this Notebook Instance.`
	`role = get_execution_role()`
	
	`# S3 prefix`
	`bucket = sagemaker_session.default_bucket()`
		Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`prefix = "Scikit-L inearLearner-pipel ine-abalone-example"`
	
	Untuk mengunduh kumpulan data sampel, tambahkan kode berikut ke buku catatan Anda:
	
	`! mkdir abalone_data`
	`! aws s3 cp s3://sage maker-sample-files /datasets/tabular/ uci_abalone/abalon e.csv ./abalone_data`
	
	Note
	Contoh dalam pola ini menggunakan Abalone Data Set dari UCI Machine Learning Repository.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Unggah kumpulan data ke bucket S3.	Di buku catatan tempat Anda menyiapkan kumpulan data sebelumnya, tambahkan kode berikut untuk mengunggah data sampel ke bucket S3:
	`WORK_DIRECTORY = "abalone_data"`
	
	`train_input = sagemaker_session.upload_data( path="{}/{}".format(WORK_DIRECTORY, "abalone.csv"), bucket=bucket, key_prefix="{}/{}".format(prefix, "train"), )`
		Ilmuwan data

Buat preprocessor data menggunakan SKLearn

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan skrip preprocessor.py.	1. Salin logika preprocessing dari file Python di GitHub repositori sklearn_abalone_featurizer.py, dan kemudian paste kode ke file Python terpisah yang disebut. `sklearn_abalone_featurizer.py` Anda dapat memodifikasi kode agar sesuai dengan dataset kustom dan alur kerja kustom Anda.	Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Simpan `sklearn_abalone_featurizer.py` file di direktori root proyek Anda (yaitu, di lokasi yang sama di mana Anda menjalankan SageMaker buku catatan Anda).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat objek SKLearn preprocessor.	Untuk membuat objek SKLearn preprocessor (disebut SKLearn Estimator) yang dapat Anda masukkan ke dalam pipeline inferensi akhir, jalankan kode berikut di buku catatan Anda: SageMaker
	`from sagemaker.sklearn.estimator import SKLearn`
	
	`FRAMEWORK_VERSION = "0.23-1"`
	`script_path = "sklearn_abalone_featurizer.py"`
	
	`sklearn_preprocessor = SKLearn(`
	`entry_point=script_path,`
	`role=role,`
	`framework_version=FRAMEWORK_VERSION,`
	`instance_type="ml.c4.xlarge",`
	`sagemaker_session=sagemaker_session,`
	`)`
	`sklearn_preprocessor.fit({"train": train_input})`
		Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji inferensi preprocessor.	Untuk mengonfirmasi bahwa preprosesor Anda didefinis ikan dengan benar, luncurkan pekerjaan transformasi batch dengan memasukkan kode berikut di SageMaker buku catatan Anda:
	
	`# Define a SKLearn Transformer from the trained SKLearn Estimator`
	`transformer = sklearn_preprocessor.transformer(`
	`instance_count=1, instance_type="ml.m5.xlarge", assemble_with="Line", accept="text/csv"`
	`)`
	
	`# Preprocess training input`
	`transformer.transform(train_input, content_type="text/csv")`
	`print("Waiting for transform job: " + transformer.latest_transform_job.job_name)`
	`transformer.wait()`
	`preprocessed_train = transformer.output_path`	

Buat model pembelajaran mesin

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat objek model.	Untuk membuat objek model berdasarkan algoritme pembelajar linier, masukkan kode berikut di SageMaker buku catatan Anda:
	`import boto3 from sagemaker .image_uris import retrieve`
	
	`ll_image = retrieve( "linear-learner", boto3.Session().region_name) s3_ll_output_key_prefix = "ll_training_output" s3_ll_output_location = "s3://{}/{}/{}/{}" .format( bucket, prefix, s3_ll_output_key_prefix, "ll_model" )`
	
	`ll_estimator = sagemaker.estimator.Estimator( ll_image, role, instance_count=1, instance_type="ml.m4.2xlarge", volume_size=20, max_run=3600, input_mode="File",`
		Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> output_path=s3_ll_
	output_location,
	    sagemaker_session=
	sagemaker_session,
	)
	
	ll_estimator.s
	et_hyperparameters
	(feature_dim=10,
	 predictor_type="re
	gressor", mini_batc
	h_size=32)
	
	ll_train_data =
	 sagemaker.inputs.T
	rainingInput(
	    preprocessed_train
	,
	    distribution="Full
	yReplicated",
	    content_type="text
	/csv",
	    s3_data_type="S3Pr
	efix",
	)
	
	data_channels =
	 {"train": ll_train_
	data}
	ll_estimator.fit(inpu
	ts=data_channels,
	 logs=True)</pre>	

	Kode sebelumnya mengambil image Amazon ECR Docker yang relevan dari Amazon ECR Registry publik untuk model, membuat objek estimator, dan kemudian	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggunakan objek tersebut untuk melatih model regresi.	

Menyebarkan pipa akhir

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan model pipa.	Untuk membuat objek model pipeline (yaitu objek preprocessor) dan menyebarkan objek, masukkan kode berikut di buku catatan Anda SageMaker :
	`from sagemaker.model import Model`
	`from sagemaker.pipeline import PipelineModel`
	`import boto3`
	`from time import gmtime, strftime`
	
	`timestamp_prefix = strftime("%Y-%m-%d-%H-%M-%S", gmtime())`
	
	`scikit_learn_inferencee_model = sklearn_preprocessor.create_model()`
	`linear_learner_model = ll_estimator.create_model()`
	
	`model_name = "inference-pipeline-" + timestamp_prefix`
		Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>endpoint_name =
	"inference-pipeline-
	ep-" + timestamp_prefix
	sm_model = PipelineM
	odel(
	name=model_name,
	role=role, models=
	[scikit_learn_infe
	rencee_model,
	linear_learner_model]
	)
	
	sm_model.deploy(init
	ial_instance_count
	=1, instance_type="ml.
	c4.xlarge", endpoint_
	name=endpoint_name)</pre>
	Note
	Anda dapat menyesuaikan jenis instance yang digunakan dalam objek model untuk memenuhi kebutuhan Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji inferensi.	Untuk mengonfirmasi bahwa titik akhir berfungsi dengan benar, jalankan contoh kode inferensi berikut di buku catatan Anda SageMaker :
	`from sagemaker.predicto r import Predictor from sagemaker.serializ ers import CSVSerial izer`
	
	`payload = "M, 0.44, 0.365, 0.125, 0.516, 0.2155, 0.114, 0.155" actual_rings = 10 predictor = Predictor( endpoint_name=endp oint_name, sagemaker _session=sagemaker _session, serialize r=CSVSerializer() )`
	
	`print(predictor .predict(payload))`
		Ilmuwan data

Sumber daya terkait

- Memproses data input sebelum membuat prediksi menggunakan pipeline SageMaker inferensi Amazon dan Scikit-learn (AWS Machine Learning Blog)
- Machine Learning ujung ke ujung dengan Amazon SageMaker (GitHub)

Menyebarkan validasi keamanan pengkodean real-time dengan menggunakan server MCP dengan Kiro dan asisten pengkodean lainnya

Ivan Girardi dan Iker Reina Fuente, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana menerapkan server Model Context Protocol (MCP) yang mengintegrasikan tiga alat pemindaian keamanan standar industri untuk menyediakan analisis keamanan kode yang komprehensif. Server memungkinkan asisten pengkodean AI (seperti Kiro, Pengembang Amazon Q, dan Cline) untuk secara otomatis memindai cuplikan kode dan infrastruktur sebagai konfigurasi kode (IAc). Dengan pemindaian ini, asisten pengkodean dapat membantu mengidentifikasi kerentanan keamanan, kesalahan konfigurasi, dan pelanggaran kepatuhan.

Generator kode AI yang dilatih pada jutaan cuplikan kode menciptakan titik buta keamanan—seberapa amankah data pelatihan itu? Pola ini memberikan validasi keamanan real-time selama pembuatan kode, membantu pengembang mengidentifikasi dan memahami potensi masalah keamanan saat mereka membuat kode. Pendekatan ini membantu pengembang mengatasi kerentanan langsung dan risiko warisan dari dependensi. Dengan menjembatani kesenjangan antara efisiensi AI dan kepatuhan keamanan, pola ini membantu memungkinkan adopsi alat pengembangan bertenaga AI yang aman.

Pola ini membantu organisasi meningkatkan praktik keamanan pengembangan mereka melalui alat pengkodean berbantuan AI, menyediakan kemampuan pemindaian keamanan berkelanjutan di berbagai bahasa pemrograman dan definisi infrastruktur. Solusinya menggabungkan kemampuan alat-alat berikut:

- Checkov untuk memindai file IAc, termasuk Terraform,, dan manifes AWS CloudFormation Kubernetes
- Semgrep untuk menganalisis beberapa bahasa pemrograman seperti Python,, Java JavaScript, dan lainnya
- Bandit untuk pemindaian keamanan Python khusus

Fitur utama dari solusi ini meliputi:

- Pemindaian delta segmen kode baru, mengurangi overhead komputasi

- Lingkungan alat keamanan yang terisolasi, mencegah kontaminasi lintas alat
- Integrasi tanpa batas dengan asisten pengkodean AI (Kiro, Pengembang Amazon Q, Cline, dan lainnya)
- Umpan balik keamanan waktu nyata selama pembuatan kode
- Aturan pemindaian yang dapat disesuaikan untuk kepatuhan organisasi

Pola ini menyediakan antarmuka terpadu untuk pemindaian keamanan dengan format respons standar, sehingga lebih mudah untuk mengintegrasikan pemeriksaan keamanan ke dalam alur kerja pengembangan. Pola ini menggunakan Python dan kerangka MCP untuk memberikan umpan balik keamanan otomatis. Pendekatan ini membantu pengembang mengidentifikasi dan mengatasi masalah keamanan di awal proses pengembangan sambil belajar tentang praktik terbaik keamanan melalui temuan terperinci.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan akses untuk menggunakan Kiro atau Amazon Q Developer, jika Anda ingin menggunakan salah satu asisten pengkodean tersebut
- Python versi 3.10 atau yang lebih baru diinstal
- uvmanajer paket diinstal
- Keakraban dengan alat dan konsep pemindaian keamanan
- Pemahaman dasar tentang IAc dan keamanan aplikasi

Batasan

- Pemindaian bandit terbatas pada file Python saja.
- Pemindaian waktu nyata dapat memengaruhi kinerja untuk basis kode yang besar.
- Keterbatasan khusus alat didasarkan pada format dan bahasa file yang didukung.
- Tinjauan manual diperlukan untuk memvalidasi temuan keamanan.
- Hasil pemindaian keamanan memerlukan keahlian keamanan untuk interpretasi yang tepat.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Python versi 3.10 atau yang lebih baru
- Checkov versi 3.0.0 atau yang lebih baru
- Semgrep versi 1.45.0 atau yang lebih baru
- Bandit versi 1.7.5 atau yang lebih baru
- MCP [cli] versi 1.11.0 atau yang lebih baru
- Pydantic versi 1.10.0 atau yang lebih baru
- Loguru versi 0.6.0 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan arsitektur untuk solusi ini.

Diagram menunjukkan alur kerja berikut:

1. Pengembang menggunakan asisten AI (misalnya, Kiro, Cline, Amazon Q Developer, atau Roo Code) untuk menghasilkan atau menganalisis kode. Asisten AI mengirimkan kode untuk pemindaian keamanan.
2. Server pemindai keamanan MCP memproses permintaan dengan merutekan ke pemindai khusus yang sesuai: Checkov untuk file IAC, Semgrep untuk analisis beberapa bahasa pemrograman, atau Bandit untuk pemindaian keamanan khusus Python.
3. Hasil pemindai dengan temuan keamanan, tingkat keparahan, deskripsi terperinci, dan perbaikan yang disarankan dikirim kembali ke pengembang melalui asisten AI.
4. Loop umpan balik berkelanjutan dibuat di mana pengembang menerima validasi keamanan waktu nyata, memungkinkan perbaikan otomatis melalui asisten AI dan mempromosikan praktik terbaik keamanan selama pengembangan.

Arsitektur mengurangi risiko keamanan umum berikut:

- Injeksi perintah
- Injeksi cepat
- Jalur traversal
- Serangan ketergantungan

- Kelelahan sumber daya

Arsitektur mengurangi risiko keamanan umum ini dengan menerapkan praktik terbaik berikut:

- Semua input model pengguna dan AI ditulis ke file sementara.
- Tidak ada input langsung yang disediakan untuk perintah antarmuka baris perintah (CLI).
- Akses sistem file dibatasi untuk direktori sementara dan file saja.
- File sementara secara otomatis dibersihkan.
- Respons pemindaian disanitasi.
- Isolasi proses yang membatasi kemampuan proses diberlakukan.
- Semua aktivitas pemindaian dicatat.

Otomatisasi dan skala

Pola ini mendukung otomatisasi melalui kemampuan berikut:

- Integrasi dengan asisten pengkodean AI untuk pemindaian kode otomatis
- Respons API standar untuk pemrosesan otomatis
- Konfigurasi melalui file konfigurasi MCP
- Support untuk pemrosesan batch dari beberapa file
- Pemindaian yang dapat diskalakan di berbagai bahasa pemrograman dan format IAC

Proses pemindaian dapat diotomatisasi melalui titik akhir API yang disediakan:

- `scan_with_checkov`untuk pemindaian IAc
- `scan_with_semgrep`untuk pemindaian kode multi-bahasa
- `scan_with_bandit`untuk pemindaian khusus Python
- `get_supported_formats`untuk validasi format

Saat memperluas alat pemindaian, ikuti prinsip desain dan praktik terbaik yang dijelaskan sebelumnya di bagian ini. Lihat juga Praktik terbaik.

Alat

Layanan AWS

- Kiro adalah layanan pengkodean agen yang bekerja bersama pengembang untuk mengubah prompt menjadi spesifikasi terperinci, kemudian menjadi kode kerja, dokumen, dan pengujian. Agen Kiro membantu pengembang memecahkan masalah yang menantang dan mengotomatiskan tugas-tugas seperti menghasilkan dokumentasi dan pengujian unit.
- Amazon Q Developer adalah asisten percakapan bertenaga AI generatif yang dapat membantu Anda memahami, membangun, memperluas, dan mengoperasikan aplikasi. AWS

Alat lainnya

- Bandit adalah alat pemindai keamanan Python khusus. Ini mendeteksi masalah keamanan Python umum seperti fungsi tidak aman, rahasia hardcode, dan kerentanan injeksi. Bandit memberikan peringkat kepercayaan dan tingkat keparahan yang terperinci.
- Checkov adalah alat analisis kode statis yang memeriksa IAc untuk kesalahan konfigurasi keamanan dan kepatuhan. Selain itu, Checkov mendeteksi pelanggaran kepatuhan dan praktik terbaik keamanan.
- Cline adalah asisten pengkodean bertenaga AI yang berjalan di VS Code.
- Loguru adalah pustaka validasi data untuk Python.
- Model Context Protocol (MCP) adalah kerangka kerja open source untuk membangun alat pengembangan yang dibantu AI.
- Pydantic adalah pustaka validasi data untuk Python.
- Semgrep menganalisis kode sumber untuk kerentanan keamanan dan bug. Ini mendukung beberapa bahasa pemrograman. Semgrep menggunakan aturan yang berfokus pada keamanan untuk analisis komprehensif. Ini memberikan peringkat kepercayaan dan tingkat keparahan yang terperinci.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub MCP Security Scanner: Real-Time Protection for AI Code Assistants. Repositori mencakup implementasi server MCP, detail tentang konfigurasi MCP untuk Kiro, Pengembang Amazon Q, Cline dan lainnya, contoh konfigurasi, dan utilitas pengujian.

Struktur repositori meliputi:

- `security_scanner_mcp_server/`- Implementasi server utama
- `docs/`- Dokumentasi dan materi demo
- `tests/`- File uji

- `mcp-config-example.json`- Contoh konfigurasi MCP
- `requirements.txt`- Ketergantungan proyek

Praktik terbaik

Implementasi pemindaian keamanan

- Tinjau temuan keamanan untuk memvalidasi dan memprioritaskan masalah.
- Tetap perbarui alat pemindaian (Checkov, Semgrep, dan Bandit) ke versi terbaru.
- Gunakan alat keamanan MCP pola ini bersama dengan langkah-langkah dan alat keamanan lainnya.
- Perbarui set aturan dan kebijakan keamanan secara teratur.

Manajemen konfigurasi

- Gunakan file konfigurasi MCP di sumber kontrol versi resmi.
- Dokumentasikan aturan dan konfigurasi kustom.

Integrasi

- Integrasikan pemindaian keamanan di awal siklus pengembangan.
- Siapkan pemindaian otomatis di kait pra-komit atau pipeline integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD).
- Konfigurasikan ambang batas keparahan yang sesuai untuk lingkungan Anda.
- Menetapkan prosedur yang jelas untuk menangani temuan keamanan.

Pertimbangan operasional

- Pantau kinerja pemindaian dan penggunaan sumber daya.
- Menerapkan penanganan kesalahan dan logging yang tepat.
- Pertahankan dokumentasi konfigurasi kustom.
- Menetapkan proses untuk meninjau dan memperbarui aturan keamanan.

Juga, ingatlah praktik terbaik berikut:

- Selalu validasi temuan keamanan dalam konteks spesifik Anda.
- Tetap perbarui alat keamanan dan dependensi.
- Gunakan beberapa alat keamanan untuk cakupan yang komprehensif.
- Ikuti praktik terbaik keamanan dalam proses pengembangan Anda.

Epik

(Pengguna Kiro) Siapkan server pemindai keamanan MCP

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan pengaturan MCP.	Anda dapat mengedit file konfigurasi di Kiro baik dengan (Opsi 1) secara manual menemukan file konfigura si atau (Opsi 2) dengan menggunakan IDE Kiro.
	• (Opsi 1) Cari dan edit file konfigurasi secara manual sebagai berikut:
	  • Jalur: `.kiro/set tings/mcp.json` (lokal) atau `~/.kiro/s ettings/mcp.json` (global)
	• (Opsi 2) Ubah file konfigura si dengan menggunakan Kiro IDE sebagai berikut:
	  • Pilih simbol Kiro di menu navigasi kiri.
	  • Arahkan ke bagian 'MCP SERVERS'.
	  • Edit pengaturan konfigura si. Perubahan yang
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dilakukan dengan IDE Kiro akan disimpan ke salah satu file konfigurasi ini: `.kiro/settings/mcp.json` (lokal) atau `~/.kiro/settings/mcp.json` (global) • (Kedua opsi) Setelah Anda menyelesaikan Opsi 1 atau Opsi 2, tambahkan konfigurasi pemindai keamanan `.kiro/settings/mcp.json` (lokal) atau `~/.kiro/settings/mcp.json` (global): `{ "mcpServers": { "security-scanner": { "command": "uvx", "args": [ "--from", "git+https://github.com/aws-samples/sample-mcp-security-scanner.git@main", "security_scanner_mcp_server" ], "env": { "FASTMCP_LOG_LEVEL": "ERROR" },`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"disabled": false, "autoApprove": [] } } }</pre>	

(Pengguna Pengembang Amazon Q) Siapkan server pemindai keamanan MCP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan MCP.	Untuk mengonfigurasi pengaturan MCP secara manual, gunakan langkah-langkah berikut: 1. Edit file konfigurasi MCP di `~/.aws/amazonq/mcp.json` (global) atau `.amazonq/mcp.json` (lokal). 2. Tambahkan konfigurasi pemindai keamanan berikut `~/.aws/amazonq/mcp.json` (global) atau `.amazonq/mcp.json` (lokal): <pre>{ "mcpServers": { "security-scanner" : { "command": "uvx", "args": ["--from",</pre>	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"git+http s://github.com/aws -samples/sample-mc p-security-scanner .git@main", "security _scanner_mcp_server"], "env": { "FASTMCP_ LOG_LEVEL": "ERROR" } } } }	

(Pengguna Cline) Siapkan server pemindai keamanan MCP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan MCP.	Untuk mengonfigurasi pengaturan MCP secara manual, gunakan langkah-langkah berikut: 1. Pilih ekstensi untuk membukanya dan kemudian pilih MCP Server. 2. Pilih tab Terinstal dan kemudian pilih Konfigurasi Server MCP untuk membuka `cline_mcp_settings.json` file. 3. Tambahkan konfigurasi pemindai keamanan	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	berikut: `cline_mcp_settings.json` `{ "mcpServers": { "security-scanner" : { "command": "uvx", "args": [ "--from", "git+https://github.com/aws-samples/sample-mcp-security-scanner.git@main", "security_scanner_mcp_server" ], "env": { "FASTMCP_LOG_LEVEL": "ERROR" }, "disabled": false, "autoApprove": [] } } }`	

Contoh analisis kode menggunakan Python dan Bandit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan analisis kode.	Untuk melakukan analisis kode dengan menggunakan Python dan Bandit, gunakan langkah-langkah berikut:	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	1. Buat file Python baru bernama `bandit_test.py` dengan konten berikut:
	
	`import pickle`
	`import yaml`
	`import subprocess`
	`from flask import Flask, request`
	`import hashlib`
	
	`app = Flask(__name__)`
	
	`@app.route("/unsafe_pickle", methods=["POST"])`
	`def unsafe_pickle_usage():`
	`    data = request.get_data()`
	`    return pickle.loads(data)`
	
	`@app.route("/command_injection", methods=["GET"])`
	`def command_injection():`
	`    command = request.args.get("cmd")`
	`    return subprocess.Popen(command, shell=True)`
	
	`def weak_crypto():`
	`    password = "secret_password"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre> return hashlib.md5(password.encode()).hexdigest() PASSWORD = "super_secret_password123"</pre> 2. Buka asisten pengkodean. Minta tindakan berikut dalam obrolan: Pindai skrip saat ini dan beri tahu saya hasilnya.	

Contoh analisis kode menggunakan Terraform dan Checkov

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lakukan analisis kode.	Untuk melakukan analisis kode dengan menggunakan Terraform dan Checkov, gunakan langkah-langkah berikut:
	
	1. Buat file Terraform baru bernama `terraform_test.tf` dengan konten berikut:
	
	<pre> resource "aws_s3_bucket" "insecure_bucket" {
	    bucket = "my-insecure-bucket"
	    acl = "public-read"
	}</pre>
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>resource "aws_secu
	rity_group"
	 "wide_open" {
	  name =
	 "allow_all"
	  description =
	 "Allow all inbound
	 traffic"
	
	  ingress {
	  from_port =
	 0
	  to_port =
	 0
	  protocol =
	 "-1"
	  cidr_blocks =
	 ["0.0.0.0/0"]
	  }
	}</pre>
	2. Buka asisten pengkodea n. Minta tindakan berikut dalam obrolan: Pindai skrip saat ini dan beri tahu saya hasilnya.	

Contoh kemampuan pemindaian tingkat lanjut

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan pemindaian yang ditargetkan.	Berikut ini adalah contoh permintaan yang dapat Anda gunakan untuk melakukan pemindaian yang ditargetkan:	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Pindai proyek saat ini dan beri tahu saya hasilnya. • Pindai baris 6-10 dalam skrip saat ini.	
Gunakan pemindaian keamanan dengan pembuatan kode.	Untuk menyelesaikan temuan keamanan dengan menggunakan loop pembuatan kode, gunakan langkah-langkah berikut (contoh ini menggunakan Kiro sebagai asisten pengkodean): 1. Di Kiro, tanyakan: Buat tabel DynamoDB di Terraform dan pindai kode dengan pemindai keamanan. 2. Di Kiro, tanyakan: Tinjau kode yang dihasilkan dan hasil pemindaian keamanan dan ikuti loop pembuatan kode yang menyelesaikan temuan keamanan.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Masalah pengaturan lingkungan	• Verifikasi bahwa Python versi 3.10 atau yang lebih baru diinstal. • Pastikan bahwa manajer uv paket diinstal.
Masalah pemindai	• Verifikasi bahwa format file didukung.

Isu	Solusi
	• Periksa apakah sintaks file valid. • Pastikan ekstensi file yang tepat digunakan. • Tinjau pengaturan batas waktu pemindaian.
Masalah integrasi	• Verifikasi bahwa server MCP sedang berjalan. • Periksa apakah file konfigurasi sudah benar. • Validasi titik akhir API. • Pastikan bahwa format respons valid.
Dukungan Tambahan	• Tinjau log untuk pesan kesalahan terperinci. • Periksa dokumentasi khusus alat. • Buat masalah di repositori.

Sumber daya terkait

AWS dokumentasi

- Infrastruktur sebagai kode (AWS Whitepaper Pengantar on) DevOps AWS

AWS Sumber daya lainnya

- Praktik Terbaik untuk Keamanan, Identitas, & Kepatuhan

Sumber daya lainnya

- Dokumentasi bandit
- Dokumentasi Checkov
- Dokumentasi Model Context Protocol (MCP)
- Praktik Pengkodean Aman OWASP (situs web OWASP Foundation)
- Dokumentasi Semgrep

Informasi tambahan

Contoh konfigurasi MCP dengan auto disetujui diaktifkan

Tanpa `autoApprove` dikonfigurasi, pengguna harus memberikan persetujuan untuk mengirim kode ke server keamanan MCP untuk pemindaian. Ketika `autoApprove` dikonfigurasi, asisten kode diizinkan untuk memanggil alat tanpa persetujuan pengguna. Alat-alat ini berjalan secara lokal di mesin, tidak ada data yang dikirim, dan hanya pemindaian kode yang dilakukan.

Konfigurasi berikut memungkinkan eksekusi otomatis semua fungsi pemindaian keamanan:

```
{
  "mcpServers": {
    "security-scanner": {
      "command": "uvx",
      "args": [
        "--from",
        "git+https://github.com/aws-samples/sample-mcp-security-scanner.git@main",
        "security_scanner_mcp_server"
      ],
      "env": {
        "FASTMCP_LOG_LEVEL": "ERROR"
      },
      "disabled": false,
      "autoApprove": [
        "scan_with_checkov",
        "scan_with_semgrep",
        "scan_with_bandit",
        "get_supported_formats"
      ]
    }
  }
}
```

Untuk mengaktifkan logging debug, setel `"FASTMCP_LOG_LEVEL"` ke`"DEBUG"`.

Format file yang didukung oleh alat pemindaian keamanan

Setiap alat pemindaian keamanan dalam solusi ini mendukung format file berikut:

Checkov (IAc)

- Terraform — .tf, .tfvars, .tfstate

- CloudFormation — .yaml, .yml.json, .template
- Kubernetes — .yaml, .yml
- Dockerfile — Dockerfile
- ARM — .json (Manajer Sumber Daya Azure)
- Bisep — .bisep
- Tanpa server — .yml.yaml
- Kemudi — .yaml, .yml.tpl
- GitHub Tindakan — .yl, .yaml
- GitLab_ci — .yml, .yaml
- Ansible — .yml, .yaml

Semgrep (Kode sumber)

- Python — .py
- JavaScript — .js
- TypeScript — .ts
- Jawa — .java
- Pergi — .go
- C — .c
- C++ — .cpp
- C # — .cs
- Ruby — .rb
- PHP — .php
- Scala — .scala
- Kotlin — .kt
- Karat — .rs

Bandit (hanya Python)

- Python — .py

Demo

Untuk pemindaian kode, coba contoh petunjuk berikut dengan asisten AI Anda:

- “Pindai skrip saat ini dan beri tahu saya hasilnya.”
- “Pindai garis 20-60 dan beri tahu saya hasilnya.”
- “Pindai sumber daya tabel Amazon DynamoDB ini dan beri tahu saya hasilnya.”

Untuk informasi selengkapnya, lihat demo pemindaian kode ini di GitHub repositori pola ini.

Untuk menghasilkan kode aman, coba contoh petunjuk berikut:

- “Hasilkan konfigurasi Terraform untuk membuat tabel DynamoDB dengan enkripsi diaktifkan dan pindai untuk masalah keamanan.”
- “Buat fungsi Lambda Python yang menulis ke DynamoDB dan memindai kerentanan.”
- “Buat CloudFormation template untuk bucket S3 dengan pengaturan keamanan yang tepat dan verifikasi itu melewati pemeriksaan keamanan.”
- “Tulis skrip Python untuk menanyakan DynamoDB dengan pagination dan pindai praktik terbaik keamanan.”
- “Buat manifes penerapan Kubernetes untuk layanan mikro dengan pengerasan keamanan dan validasi.”

Untuk informasi selengkapnya, lihat pembuatan kode ini dengan demo pemindaian keamanan di GitHub repositori pola ini.

Kembangkan asisten berbasis obrolan AI generatif tingkat lanjut dengan menggunakan RAG dan prompt ReAct

Praveen Kumar Jeyarajan, Shuai Cao, Noah Hamilton, Kiowa Jackson, Jundong Qiao, dan Kara Yang, Amazon Web Services

Ringkasan

Sebuah perusahaan tipikal memiliki 70 persen datanya yang terperangkap dalam sistem siloed. Anda dapat menggunakan asisten berbasis obrolan berbasis AI generatif untuk membuka wawasan dan hubungan antara silo data ini melalui interaksi bahasa alami. Untuk mendapatkan hasil maksimal dari AI generatif, output harus dapat dipercaya, akurat, dan inklusif dari data perusahaan yang tersedia. Asisten berbasis obrolan yang sukses bergantung pada hal-hal berikut:

- Model AI generatif (seperti Anthropic Claude 2)
- Vektorisasi sumber data
- Teknik penalaran tingkat lanjut, seperti ReAct kerangka kerja, untuk mendorong model

Pola ini menyediakan pendekatan pengambilan data dari sumber data seperti bucket Amazon Simple Storage Service (Amazon S3), AWS Glue, dan Amazon Relational Database Service (Amazon RDS). Nilai diperoleh dari data tersebut dengan menyisipkan Retrieval Augmented Generation (RAG) dengan metode. chain-of-thought Hasilnya mendukung percakapan asisten berbasis obrolan kompleks yang memanfaatkan keseluruhan data yang disimpan perusahaan Anda.

Pola ini menggunakan SageMaker manual Amazon dan tabel data harga sebagai contoh untuk mengeksplorasi kemampuan asisten berbasis obrolan AI generatif. Anda akan membangun asisten berbasis obrolan yang membantu pelanggan mengevaluasi SageMaker layanan dengan menjawab pertanyaan tentang harga dan kemampuan layanan. Solusinya menggunakan perpustakaan Streamlit untuk membangun aplikasi frontend dan LangChain kerangka kerja untuk mengembangkan backend aplikasi yang didukung oleh model bahasa besar (LLM).

Pertanyaan ke asisten berbasis obrolan dipenuhi dengan klasifikasi maksud awal untuk merutekan ke salah satu dari tiga kemungkinan alur kerja. Alur kerja paling canggih menggabungkan panduan penasehat umum dengan analisis harga yang kompleks. Anda dapat menyesuaikan pola agar sesuai dengan kasus penggunaan perusahaan, perusahaan, dan industri.

Prasyarat dan batasan

Prasyarat

- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi
- AWS Cloud Development Kit (AWS CDK) Toolkit 2.114.1 atau yang lebih baru diinstal dan dikonfigurasi
- Keakraban dasar dengan Python dan AWS CDK
- Git diinstal
- Docker diinstal
- Python 3.11 atau yang lebih baru diinstal dan dikonfigurasi (untuk informasi lebih lanjut, lihat bagian Alat)
- Bootstrap akun AWS aktif dengan menggunakan AWS CDK
- Akses model Amazon Titan dan Anthropic Claude diaktifkan di layanan Amazon Bedrock
- Kredensi keamanan AWS, termasuk`AWS_ACCESS_KEY_ID`, dikonfigurasi dengan benar di lingkungan terminal Anda

Batasan

- LangChain tidak mendukung setiap LLM untuk streaming. Model Anthropic Claude didukung, tetapi model dari AI21 Labs tidak.
- Solusi ini diterapkan ke satu akun AWS.
- Solusi ini hanya dapat digunakan di Wilayah AWS di mana Amazon Bedrock dan Amazon Kendra tersedia. Untuk informasi tentang ketersediaan, lihat dokumentasi untuk Amazon Bedrock dan Amazon Kendra.

Versi produk

- Python versi 3.11 atau yang lebih baru
- Streamlit versi 1.30.0 atau yang lebih baru
- Streamlit-chat versi 0.1.1 atau yang lebih baru
- LangChain versi 0.1.12 atau yang lebih baru
- AWS CDK versi 2.132.1 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- Amazon Athena
- Amazon Bedrock
- Amazon Elastic Container Service (Amazon ECS)
- AWS Glue
- AWS Lambda
- Amazon S3
- Amazon Kendra
- Elastic Load Balancing

Arsitektur target

Kode AWS CDK akan menerapkan semua sumber daya yang diperlukan untuk menyiapkan aplikasi asisten berbasis obrolan di akun AWS. Aplikasi asisten berbasis obrolan yang ditunjukkan dalam diagram berikut dirancang untuk menjawab pertanyaan SageMaker terkait dari pengguna. Pengguna terhubung melalui Application Load Balancer ke VPC yang berisi cluster Amazon ECS yang menghosting aplikasi Streamlit. Fungsi Lambda orkestrasi terhubung ke aplikasi. Sumber data bucket S3 menyediakan data ke fungsi Lambda melalui Amazon Kendra dan AWS Glue. Fungsi Lambda terhubung ke Amazon Bedrock untuk menjawab pertanyaan (pertanyaan) dari asisten pengguna berbasis obrolan.

1. Fungsi orkestrasi Lambda mengirimkan permintaan prompt LLM ke model Amazon Bedrock (Claude 2).
2. Amazon Bedrock mengirimkan respons LLM kembali ke fungsi Lambda orkestrasi.

Aliran logika dalam fungsi Lambda orkestrasi

Ketika pengguna mengajukan pertanyaan melalui aplikasi Streamlit, itu memanggil fungsi Lambda orkestrasi secara langsung. Diagram berikut menunjukkan aliran logika ketika fungsi Lambda dipanggil.

- Langkah 1 — Masukan `query` (pertanyaan) diklasifikasikan ke dalam salah satu dari tiga maksud:
  - Pertanyaan SageMaker panduan umum
  - Pertanyaan SageMaker harga umum (pelatihan/inferensi)
  - Pertanyaan kompleks terkait dengan SageMaker dan harga
- Langkah 2 — Input `query` memulai salah satu dari tiga layanan:
  - `RAG Retrieval service`, yang mengambil konteks yang relevan dari database vektor Amazon Kendra dan memanggil LLM melalui Amazon Bedrock untuk meringkas konteks yang diambil sebagai respons.
  - `Database Query service`, yang menggunakan- LLM, metadata database, dan baris sampel dari tabel yang relevan untuk mengubah input `query` menjadi kueri SQL. Layanan Kueri Database menjalankan kueri SQL terhadap database SageMaker harga melalui Amazon Athena dan merangkum hasil kueri sebagai respons.
  - `In-context ReACT Agent service`, yang memecah input `query` menjadi beberapa langkah sebelum memberikan respons. Agen menggunakan `RAG Retrieval service` dan `Database Query service` sebagai alat untuk mengambil informasi yang relevan selama proses penalaran. Setelah proses penalaran dan tindakan selesai, agen menghasilkan jawaban akhir sebagai respons.
- Langkah 3 - Respons dari fungsi Lambda orkestrasi dikirim ke aplikasi Streamlit sebagai output.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon Simple Storage Service (Amazon S3) dengan menggunakan SQL standar.
- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.

- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.
- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data. Pola ini menggunakan crawler AWS Glue dan tabel AWS Glue Data Catalog.
- Amazon Kendra adalah layanan pencarian cerdas yang menggunakan pemrosesan bahasa alami dan algoritme pembelajaran mesin canggih untuk mengembalikan jawaban spesifik atas pertanyaan pencarian dari data Anda.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone.

Repositori kode

Kode untuk pola ini tersedia di GitHub genai-bedrock-chatbotrepositori.

Repositori kode berisi file dan folder berikut:

- `assets`folder — Aset statis diagram arsitektur dan dataset publik
- `code/lambda-container`folder - Kode Python yang dijalankan dalam fungsi Lambda
- `code/streamlit-app`folder - Kode Python yang dijalankan sebagai gambar kontainer di Amazon ECS
- `tests`folder — File Python yang dijalankan untuk menguji unit konstruksi AWS CDK
- `code/code_stack.py`— AWS CDK membangun file Python yang digunakan untuk membuat sumber daya AWS
- `app.py`— File Python tumpukan AWS CDK yang digunakan untuk menyebarkan sumber daya AWS di akun AWS target

- `requirements.txt`— Daftar semua dependensi Python yang harus diinstal untuk AWS CDK
- `requirements-dev.txt`— Daftar semua dependensi Python yang harus diinstal untuk AWS CDK untuk menjalankan rangkaian unit-test
- `cdk.json`— File input untuk memberikan nilai yang diperlukan untuk memutar sumber daya

Catatan: Kode AWS CDK menggunakan konstruksi L3 (lapisan 3) dan kebijakan AWS Identity and Access Management (IAM) yang dikelola oleh AWS untuk menerapkan solusi.

Praktik terbaik

- Contoh kode yang diberikan di sini adalah untuk demo proof-of-concept (PoC) atau pilot saja. Jika Anda ingin membawa kode ke Produksi, pastikan untuk menggunakan praktik terbaik berikut:
  - Pencatatan akses Amazon S3 diaktifkan.
  - VPC Flow Logs diaktifkan.
  - Indeks Amazon Kendra Enterprise Edition diaktifkan.
- Siapkan pemantauan dan peringatan untuk fungsi Lambda. Untuk informasi selengkapnya, lihat Memantau dan memecahkan masalah fungsi Lambda. Untuk praktik terbaik umum saat bekerja dengan fungsi Lambda, lihat dokumentasi AWS.

Epik

Siapkan kredensi AWS di mesin lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekspor variabel untuk akun dan Wilayah AWS tempat tumpukan akan digunakan.	Untuk memberikan kredensi AWS untuk AWS CDK dengan menggunakan variabel lingkungan, jalankan perintah berikut. `export CDK_DEFAU LT_ACCOUNT=<12 Digit AWS Account Number>`	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`export CDK_DEFAU LT_REGION=<region>`	
Siapkan profil AWS CLI.	Untuk menyiapkan profil AWS CLI untuk akun, ikuti petunjuk dalam dokumentasi AWS.	DevOps insinyur, AWS DevOps

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repo di mesin lokal Anda.	Untuk mengkloning repositori, jalankan perintah berikut di terminal Anda. `git clone https://github.com/awslabs/genai-bedrock-chatbot.git`	DevOps insinyur, AWS DevOps

Siapkan lingkungan virtual Python dan instal dependensi yang diperlukan.	Untuk mengatur lingkungan virtual Python, jalankan perintah berikut.
	`cd genai-bedrock-chatbot`
	`python3 -m venv .venv`
	`source .venv/bin/activate`
	Untuk mengatur dependensi yang diperlukan, jalankan perintah berikut.
		DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`pip3 install -r requirements.txt`	
Siapkan lingkungan AWS CDK dan sintesis kode AWS CDK.	1. Untuk mengatur lingkungan AWS CDK di akun AWS Anda, jalankan perintah berikut. `cdk bootstrap aws://ACCOUNT-NUMBER/REGION` 2. Untuk mengonversi kode ke konfigurasi AWS CloudFormation stack, jalankan perintah`cdk synth`.	DevOps insinyur, AWS DevOps

Konfigurasikan dan terapkan aplikasi asisten berbasis obrolan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Penyediaan akses model Claude.	Untuk mengaktifkan akses model Anthropic Claude untuk akun AWS Anda, ikuti petunjuk dalam dokumentasi Amazon Bedrock.	AWS DevOps
Menyebarkan sumber daya di akun.	Untuk menerapkan sumber daya di akun AWS dengan menggunakan AWS CDK, lakukan hal berikut: 1. Di root repositori kloning, dalam `cdk.json`	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	file, berikan input untuk parameter. `logging` Contoh nilai adalah`INFO`,`DEBUG`,`WARN`, dan`ERROR`. Nilai-nilai ini menentuka n pesan log-level untuk fungsi Lambda dan aplikasi Streamlit. 2. `app.py`File di root repositor i kloning berisi nama CloudFormation tumpukan AWS yang digunakan untuk penerapan. Nama stack default adalah`chatbot-s tack` . 3. Untuk menyebarkan sumber daya, jalankan perintah`cdk deploy`. `cdk deploy`Perintah ini menggunakan konstruksi L3 untuk membuat beberapa fungsi Lambda untuk menyalin dokumen dan file dataset CSV ke bucket S3. 4. Setelah perintah selesai, masuk ke AWS Management Console, buka CloudFormation konsol, dan tinjau apakah tumpukan berhasil diterapkan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Setelah penerapan berhasil, Anda dapat mengakses aplikasi asisten berbasis obrolan dengan menggunakan URL yang disediakan di bagian Output. CloudFormation	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan crawler AWS Glue dan buat tabel Katalog Data.	Crawler AWS Glue digunakan untuk menjaga skema data tetap dinamis. Solusi ini membuat dan memperbarui partisi di tabel AWS Glue Data Catalog dengan menjalank an crawler sesuai permintaa n. Setelah file kumpulan data CSV disalin ke bucket S3, jalankan crawler AWS Glue dan buat skema tabel Katalog Data untuk pengujian: 1. Arahkan ke konsol AWS Glue. 2. Di panel navigasi, di bawah Katalog Data, pilih Crawler. 3. Pilih crawler dengan akhiran `sagemaker-pricing-crawler` . 4. Jalankan crawler. 5. Setelah crawler berjalan dengan sukses, crawler akan membuat tabel AWS Glue Data Catalog. **Note** Kode AWS CDK mengonfigurasi crawler AWS Glue agar berjalan sesuai permintaan, tetapi Anda juga dapat	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menjadwalkannya untuk dijalankan secara berkala.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai pengindeksan dokumen.	Setelah file disalin ke bucket S3, gunakan Amazon Kendra untuk merayapi dan mengindeksnya: 1. Arahkan ke konsol Amazon Kendra. 2. Pilih indeks dengan akhiran `chatbot-index` . 3. Di panel navigasi, pilih Sumber data, dan pilih konektor sumber data dengan akhiran `chatbot-index` . 4. Pilih Sinkronkan Sekarang untuk memulai proses pengindeksan. **Note** Kode AWS CDK mengonfigurasi sinkronisasi indeks Amazon Kendra agar berjalan sesuai permintaan, tetapi Anda juga dapat menjalankannya secara berkala dengan menggunakan parameter Jadwal.	AWS DevOps, DevOps insinyur

Bersihkan semua sumber daya AWS dalam solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya AWS.	Setelah Anda menguji solusinya, bersihkan sumber daya: 1. Untuk menghapus sumber daya AWS yang digunakan oleh solusi, jalankan perintah`cdk destroy`. 2. Hapus semua objek dari dua ember S3, lalu lepaskan ember. Untuk informasi selengkap nya, lihat Menghapus bucket.	DevOps insinyur, AWS DevOps

Pemecahan masalah

Isu	Solusi
AWS CDK mengembalikan kesalahan.	Untuk bantuan terkait masalah AWS CDK, lihat Memecahkan masalah umum AWS CDK.

Sumber daya terkait

- Batuan Dasar Amazon:
  - Akses model
  - Parameter inferensi untuk model pondasi
- Membangun fungsi Lambda dengan Python
- Memulai dengan AWS CDK
- Bekerja dengan AWS CDK dengan Python

- Pembuat Aplikasi AI Generatif di AWS
- LangChain dokumentasi
- Dokumentasi Streamlit

Informasi tambahan

Perintah AWS CDK

Saat bekerja dengan AWS CDK, ingatlah perintah berguna berikut ini:

- Daftar semua tumpukan di aplikasi

```
cdk ls
```

- Memancarkan template AWS yang disintesis CloudFormation

```
cdk synth
```

- Menerapkan tumpukan ke akun AWS default dan Wilayah

```
cdk deploy
```

- Membandingkan tumpukan yang diterapkan dengan status saat ini

```
cdk diff
```

- Membuka dokumentasi AWS CDK

```
cdk docs
```

- Menghapus CloudFormation tumpukan dan menghapus sumber daya yang diterapkan AWS

```
cdk destroy
```

Kembangkan asisten berbasis obrolan otomatis sepenuhnya dengan menggunakan agen dan basis pengetahuan Amazon Bedrock

Jundong Qiao, Shuai Cao, Noah Hamilton, Kiowa Jackson, Praveen Kumar Jeyarajan, dan Kara Yang, Amazon Web Services

Ringkasan

Banyak organisasi menghadapi tantangan saat membuat asisten berbasis obrolan yang mampu mengatur beragam sumber data untuk menawarkan jawaban yang komprehensif. Pola ini menyajikan solusi untuk mengembangkan asisten berbasis obrolan yang mampu menjawab pertanyaan dari dokumentasi dan database, dengan penerapan langsung.

Dimulai dengan Amazon Bedrock, layanan kecerdasan buatan generatif (AI) generatif yang dikelola sepenuhnya ini menyediakan beragam model pondasi canggih ()FMs. Ini memfasilitasi pembuatan aplikasi AI generatif yang efisien dengan fokus yang kuat pada privasi dan keamanan. Dalam konteks pengambilan dokumentasi, Retrieval Augmented Generation (RAG) adalah fitur penting. Ini menggunakan basis pengetahuan untuk menambah permintaan FM dengan informasi yang relevan secara kontekstual dari sumber eksternal. Indeks Amazon OpenSearch Tanpa Server berfungsi sebagai basis data vektor di belakang basis pengetahuan untuk Amazon Bedrock. Integrasi ini ditingkatkan melalui rekayasa cepat yang cermat untuk meminimalkan ketidakakuratan dan memastikan bahwa tanggapan tertanam dalam dokumentasi faktual. Untuk kueri database, Amazon Bedrock mengubah pertanyaan tekstual menjadi kueri SQL terstruktur, menggabungkan parameter tertentu. FMs Ini memungkinkan pengambilan data yang tepat dari database yang dikelola oleh database AWS Glue. Amazon Athena digunakan untuk pertanyaan ini.

Untuk menangani pertanyaan yang lebih rumit, mencapai jawaban yang komprehensif menuntut informasi yang bersumber dari dokumentasi dan database. Agen untuk Amazon Bedrock adalah fitur AI generatif yang membantu Anda membangun agen otonom yang dapat memahami tugas kompleks dan memecahnya menjadi tugas yang lebih sederhana untuk orkestrasi. Kombinasi wawasan yang diambil dari tugas-tugas yang disederhanakan, difasilitasi oleh agen otonom Amazon Bedrock, meningkatkan sintesis informasi, yang mengarah ke jawaban yang lebih menyeluruh dan lengkap. Pola ini menunjukkan cara membangun asisten berbasis obrolan dengan menggunakan Amazon Bedrock dan layanan serta fitur AI generatif terkait dalam solusi otomatis.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Docker, diinstal
- AWS Cloud Development Kit (AWS CDK), diinstal dan di-bootstrap ke atau Wilayah AWS `us-east-1` `us-west-2`
- AWS CDK Toolkit versi 2.114.1 atau yang lebih baru, diinstal
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Python versi 3.11 atau yang lebih baru, diinstal
- Di Amazon Bedrock, aktifkan akses ke Claude 2, Claude 2.1, Claude Instant, dan Titan Embeddings G1 — Teks

Batasan

- Solusi ini diterapkan ke satu akun AWS.
- Solusi ini hanya dapat diterapkan di Wilayah AWS di mana Amazon Bedrock dan Amazon Tanpa OpenSearch Server didukung. Untuk informasi selengkapnya, lihat dokumentasi untuk Amazon Bedrock dan Amazon Tanpa OpenSearch Server.

Versi produk

- LLAMA-index versi 0.10.6 atau yang lebih baru
- Sqlalchemy versi 2.0.23 atau yang lebih baru
- OpenSearch-PY versi 2.4.2 atau yang lebih baru
- Requests_AWS4Auth versi 1.2.3 atau yang lebih baru
- AWS SDK untuk Python (Boto3) versi 1.34.57 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak open source untuk mendefinisikan infrastruktur cloud dalam kode dan menyediakannya melalui AWS.

CloudFormation Tumpukan AWS CDK yang digunakan dalam pola ini menerapkan sumber daya AWS berikut:

- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)
- AWS Glue Data Catalog, untuk komponen database AWS Glue
- AWS Lambda
- AWS Identity and Access Management (IAM)
- Amazon Tanpa OpenSearch Server
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon Elastic Container Service (Amazon ECS)
- AWS Fargate
- Amazon Virtual Private Cloud (Amazon VPC)
- Penyeimbang Beban Aplikasi

Arsitektur target

Diagram menunjukkan penyiapan cloud-native AWS yang komprehensif dalam satu Wilayah AWS, menggunakan beberapa layanan AWS. Antarmuka utama untuk asisten berbasis obrolan adalah aplikasi Streamlit yang dihosting di cluster Amazon ECS. Application Load Balancer mengelola aksesibilitas. Kueri yang dibuat melalui antarmuka ini mengaktifkan fungsi `Invocation` Lambda, yang kemudian berinteraksi dengan agen untuk Amazon Bedrock. Agen ini menanggapi pertanyaan pengguna dengan berkonsultasi dengan basis pengetahuan untuk Amazon Bedrock atau dengan menjalankan fungsi Lambda. `Agent executor` Fungsi ini memicu serangkaian tindakan yang terkait dengan agen, mengikuti skema API yang telah ditentukan sebelumnya. Basis pengetahuan untuk Amazon Bedrock menggunakan indeks OpenSearch Tanpa Server sebagai fondasi basis data vektor mereka. Selain itu, `Agent executor` fungsi ini menghasilkan kueri SQL yang dijalankan terhadap database AWS Glue melalui Amazon Athena.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon Simple Storage Service (Amazon S3) dengan menggunakan SQL standar.

- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone.
- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data. Pola ini menggunakan crawler AWS Glue dan tabel AWS Glue Data Catalog.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon OpenSearch Serverless adalah konfigurasi tanpa server sesuai permintaan untuk Amazon Service. OpenSearch Dalam pola ini, indeks OpenSearch Tanpa Server berfungsi sebagai database vektor untuk basis pengetahuan untuk Amazon Bedrock.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- Streamlit adalah kerangka kerja Python open source untuk membuat aplikasi data.

Repositori kode

Kode untuk pola ini tersedia di GitHub genai-bedrock-agent-chatbotrepositori. Repositori kode berisi file dan folder berikut:

- `assets`folder — Aset statis, seperti diagram arsitektur dan dataset publik.
- `code/lambdas/action-lambda`folder — Kode Python untuk fungsi Lambda yang bertindak sebagai tindakan untuk agen Amazon Bedrock.
- `code/lambdas/create-index-lambda`folder — Kode Python untuk fungsi Lambda yang menciptakan indeks Tanpa Server. OpenSearch
- `code/lambdas/invoke-lambda`folder — Kode Python untuk fungsi Lambda yang memanggil agen Amazon Bedrock, yang dipanggil langsung dari aplikasi Streamlit.
- `code/lambdas/update-lambda`folder — Kode Python untuk fungsi Lambda yang memperbarui atau menghapus sumber daya setelah sumber daya AWS digunakan melalui AWS CDK.
- `code/layers/boto3_layer`folder — Tumpukan AWS CDK yang membuat lapisan Boto3 yang dibagikan di semua fungsi Lambda.
- `code/layers/opensearch_layer`folder — Tumpukan AWS CDK yang membuat lapisan OpenSearch Tanpa Server yang menginstal semua dependensi untuk membuat indeks.
- `code/streamlit-app`folder - Kode Python yang dijalankan sebagai gambar kontainer di Amazon ECS
- `code/code_stack.py`— AWS CDK membangun file Python yang membuat sumber daya AWS.
- `app.py`— AWS CDK menumpuk file Python yang menyebarkan sumber daya AWS di akun AWS target.
- `requirements.txt`— Daftar semua dependensi Python yang harus diinstal untuk AWS CDK.
- `cdk.json`— File input untuk memberikan nilai-nilai yang diperlukan untuk membuat sumber daya. Juga, di `context/config` bidang, Anda dapat menyesuaikan solusi yang sesuai. Untuk informasi selengkapnya tentang penyesuaian, lihat bagian Informasi tambahan.

Praktik terbaik

- Contoh kode yang diberikan di sini adalah untuk proof-of-concept (PoC) atau tujuan percontohan saja. Jika Anda ingin membawa kode ke produksi, pastikan untuk menggunakan praktik terbaik berikut:
  - Aktifkan pencatatan akses Amazon S3
  - Aktifkan Log Aliran VPC
- Siapkan pemantauan dan peringatan untuk fungsi Lambda. Untuk informasi selengkapnya, lihat Memantau dan memecahkan masalah fungsi Lambda. Untuk praktik terbaik, lihat Praktik terbaik untuk bekerja dengan fungsi AWS Lambda.

Epik

Siapkan kredensyal AWS di workstation lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Variabel ekspor untuk akun dan Wilayah.	Untuk memberikan kredensi AWS untuk AWS CDK dengan menggunakan variabel lingkungan, jalankan perintah berikut. `export CDK_DEFAU LT_ACCOUNT=<12-digit AWS account number> export CDK_DEFAU LT_REGION=<Region>`	AWS DevOps, DevOps insinyur
Siapkan profil bernama AWS CLI.	Untuk mengatur profil bernama AWS CLI untuk akun, ikuti petunjuk dalam pengaturan Konfigurasi dan file kredensi.	AWS DevOps, DevOps insinyur

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repo ke workstation lokal Anda.	Untuk mengkloning repositor i, jalankan perintah berikut di terminal Anda. `git clone https://g ithub.com/awslabs/ genai-bedrock-agent- chatbot.git`	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan lingkungan virtual Python.	Untuk mengatur lingkungan virtual Python, jalankan perintah berikut.
	`cd genai-bedrock-agent-chatbot`
	`python3 -m venv .venv`
	`source .venv/bin/activate`
	Untuk mengatur dependensi yang diperlukan, jalankan perintah berikut.
	`pip3 install -r requirements.txt`
		DevOps insinyur, AWS DevOps

Siapkan lingkungan AWS CDK.	Untuk mengonversi kode ke CloudFormation template AWS, jalankan perintah`cdk synth`.	AWS DevOps, DevOps insinyur

Konfigurasikan dan terapkan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan sumber daya di akun.	Untuk menerapkan sumber daya di akun AWS dengan menggunakan AWS CDK, lakukan hal berikut: 1. Di root repositori kloning, dalam `cdk.json` file, berikan input untuk parameter	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	logging. Contoh nilai adalah`INFO`,`DEBUG`,`WARN`, dan`ERROR`. Nilai-nilai ini menentuka n pesan log-level untuk fungsi Lambda dan aplikasi Streamlit. 2. `cdk.json`File di root repositori kloning berisi nama CloudFormation tumpukan AWS yang digunakan untuk penerapan . Nama stack default adalah`chatbot-s tack` . Nama agen Amazon Bedrock default adalah`ChatbotBe drockAgent` , dan alias agen Amazon Bedrock default adalah. `Chatbot_A gent` 3. Untuk menyebarkan sumber daya, jalankan perintah`cdk deploy`. `cdk deploy`Perintah ini menggunakan konstruksi lapisan-3 untuk membuat beberapa fungsi Lambda untuk menyalin dokumen dan file dataset CSV ke bucket S3. Ini juga menyebarkan agen Amazon Bedrock, basis pengetahu	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an, dan fungsi `Action group` Lambda untuk agen Amazon Bedrock. 4. Masuk ke AWS Management Console, lalu buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation/. 5. Konfirmasikan bahwa tumpukan berhasil digunakan. Untuk petunjuknya, lihat Meninjau tumpukan Anda di CloudFormation konsol AWS. Setelah penerapan berhasil, Anda dapat mengakses aplikasi asisten berbasis obrolan dengan menggunakan URL yang disediakan pada tab Output di konsol. CloudFormation	

Bersihkan semua sumber daya AWS dalam solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya AWS.	Setelah Anda menguji solusinya, untuk membersihkan sumber daya, jalankan perintah`cdk destroy`.	AWS DevOps, DevOps insinyur

Sumber daya terkait

Dokumentasi AWS

- Sumber daya Amazon Bedrock:
  - Akses model
  - Parameter inferensi untuk model pondasi
  - Agen untuk Amazon Bedrock
  - Basis pengetahuan untuk Amazon Bedrock
- Membangun fungsi Lambda dengan Python
- Sumber daya AWS CDK:
  - Memulai dengan AWS CDK
  - Memecahkan masalah umum AWS CDK
  - Bekerja dengan AWS CDK dengan Python
- Pembuat Aplikasi AI Generatif di AWS

Sumber daya AWS lainnya

- Mesin Vektor untuk Amazon Tanpa OpenSearch Server

Sumber daya lainnya

- LlamaIndex dokumentasi
- Dokumentasi Streamlit

Informasi tambahan

Sesuaikan asisten berbasis obrolan dengan data Anda sendiri

Untuk mengintegrasikan data kustom Anda untuk menerapkan solusi, ikuti panduan terstruktur ini. Langkah-langkah ini dirancang untuk memastikan proses integrasi yang mulus dan efisien, memungkinkan Anda untuk menerapkan solusi secara efektif dengan data yang dipesan lebih dahulu.

Untuk integrasi data basis pengetahuan

Persiapan data

1. Temukan `assets/knowledgebase_data_source/` direktori.
2. Tempatkan dataset Anda di dalam folder ini.

Penyesuaian konfigurasi

1. Buka file `cdk.json`.
2. Arahkan ke `context/configure/paths/knowledgebase_file_name` bidang, lalu perbarui sesuai dengan itu.
3. Arahkan ke `bedrock_instructions/knowledgebase_instruction` bidang, lalu perbarui untuk secara akurat mencerminkan nuansa dan konteks kumpulan data baru Anda.

Untuk integrasi data struktural

Organisasi data

1. Di dalam `assets/data_query_data_source/` direktori, buat subdirektori, sepertitabular_data.
2. Masukkan kumpulan data terstruktur Anda (format yang dapat diterima termasuk CSV, JSON, ORC, dan Parket) ke dalam subfolder yang baru dibuat ini.
3. Jika Anda terhubung ke database yang ada, perbarui fungsi `create_sql_engine()` `code/lambda/action-lambda/build_query_engine.py` untuk terhubung ke database Anda.

Pembaruan konfigurasi dan kode

1. Dalam `cdk.json` file, perbarui `context/configure/paths/athena_table_data_prefix` bidang untuk menyelaraskan dengan jalur data baru.
2. Merevisi `code/lambda/action-lambda/dynamic_examples.csv` dengan memasukkan text-to-SQL contoh baru yang sesuai dengan dataset Anda.
3. Merevisi `code/lambda/action-lambda/prompt_templates.py` untuk mencerminkan atribut kumpulan data terstruktur Anda.
4. Dalam `cdk.json` file, perbarui `context/configure/bedrock_instructions/action_group_description` bidang untuk menjelaskan tujuan dan fungsionalitas fungsi `Action group` Lambda.

5. Dalam `assets/agent_api_schema/artifacts_schema.json` file, jelaskan fungsionalitas baru fungsi `Action group` Lambda Anda.

Pembaruan umum

Dalam `cdk.json` file, di `context/configure/bedrock_instructions/agent_instruction` bagian ini, berikan deskripsi komprehensif tentang fungsionalitas dan tujuan desain agen Amazon Bedrock yang dimaksudkan, dengan mempertimbangkan data yang baru terintegrasi.

Dokumentasikan pengetahuan kelembagaan dari input suara dengan menggunakan Amazon Bedrock dan Amazon Transcribe

Praveen Kumar Jeyarajan, Jundong Qiao, Rajiv Upadhyay, dan Megan Wu, Amazon Web Services

Ringkasan

Menangkap pengetahuan kelembagaan sangat penting untuk memastikan keberhasilan dan ketahanan organisasi. Pengetahuan kelembagaan mewakili kebijaksanaan kolektif, wawasan, dan pengalaman yang dikumpulkan oleh karyawan dari waktu ke waktu, seringkali bersifat diam-diam dan diturunkan secara informal. Kekayaan informasi ini mencakup pendekatan unik, praktik terbaik, dan solusi untuk masalah rumit yang mungkin tidak didokumentasikan di tempat lain. Dengan memformalkan dan mendokumentasikan pengetahuan ini, perusahaan dapat melestarikan memori kelembagaan, mendorong inovasi, meningkatkan proses pengambilan keputusan, dan mempercepat kurva pembelajaran bagi karyawan baru. Selain itu, mempromosikan kolaborasi, memberdayakan individu, dan menumbuhkan budaya perbaikan berkelanjutan. Pada akhirnya, memanfaatkan pengetahuan institusional membantu perusahaan menggunakan aset mereka yang paling berharga — kecerdasan kolektif tenaga kerja mereka — untuk menavigasi tantangan, mendorong pertumbuhan, dan mempertahankan keunggulan kompetitif dalam lingkungan bisnis yang dinamis.

Pola ini menjelaskan cara menangkap pengetahuan kelembagaan melalui rekaman suara dari karyawan senior. Ini menggunakan Amazon Transcribe dan Amazon Bedrock untuk dokumentasi dan verifikasi sistematis. Dengan mendokumentasikan pengetahuan informal ini, Anda dapat melestarikannya dan membagikannya dengan kelompok karyawan berikutnya. Upaya ini mendukung keunggulan operasional dan meningkatkan efektivitas program pelatihan melalui penggabungan pengetahuan praktis yang diperoleh melalui pengalaman langsung.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Docker, diinstal
- AWS Cloud Development Kit (AWS CDK) versi 2.114.1 atau yang lebih baru, diinstal dan di-bootstrap ke Wilayah AWS atau AWS `us-east-1` `us-west-2`
- AWS CDK Toolkit versi 2.114.1 atau yang lebih baru, diinstal

- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Python versi 3.12 atau yang lebih baru, diinstal
- Izin untuk membuat Amazon Transcribe, Amazon Bedrock, Amazon Simple Storage Service (Amazon S3), dan sumber daya AWS Lambda

Batasan

- Solusi ini diterapkan ke satu akun AWS.
- Solusi ini hanya dapat diterapkan di Wilayah AWS di mana Amazon Bedrock dan Amazon Transcribe tersedia. Untuk informasi tentang ketersediaan, lihat dokumentasi untuk Amazon Bedrock dan Amazon Transcribe.
- File audio harus dalam format yang didukung Amazon Transcribe. Untuk daftar format yang didukung, lihat Format media dalam dokumentasi Transcribe.

Versi produk

- AWS SDK untuk Python (Boto3) versi 1.34.57 atau yang lebih baru
- LangChain versi 0.1.12 atau yang lebih baru

Arsitektur

Arsitektur mewakili alur kerja tanpa server di AWS. AWS Step Functions mengatur fungsi Lambda untuk pemrosesan audio, analisis teks, dan pembuatan dokumen. Diagram berikut menunjukkan alur kerja Step Functions, juga dikenal sebagai mesin negara.

Setiap langkah dalam mesin negara ditangani oleh fungsi Lambda yang berbeda. Berikut ini adalah langkah-langkah dalam proses pembuatan dokumen:

1. Fungsi `preprocess` Lambda memvalidasi input yang diteruskan ke Step Functions dan mencantumkan semua file audio yang ada di path folder Amazon S3 URI yang disediakan. Fungsi Lambda hilir dalam alur kerja menggunakan daftar file untuk memvalidasi, meringkas, dan menghasilkan dokumen.
2. Fungsi `transcribe` Lambda menggunakan Amazon Transcribe untuk mengonversi file audio menjadi transkrip teks. Fungsi Lambda ini bertanggung jawab untuk memulai proses transkripsi

dan secara akurat mengubah ucapan menjadi teks, yang kemudian disimpan untuk diproses selanjutnya.

3. Fungsi `validate` Lambda menganalisis transkrip teks, menentukan relevansi tanggapan terhadap pertanyaan awal. Dengan menggunakan model bahasa besar (LLM) melalui Amazon Bedrock, ini mengidentifikasi dan memisahkan jawaban berdasarkan topik dari tanggapan di luar topik.
4. Fungsi `summarize` Lambda menggunakan Amazon Bedrock untuk menghasilkan ringkasan jawaban pada topik yang koheren dan ringkas.
5. Fungsi `generate` Lambda merakit ringkasan menjadi dokumen yang terstruktur dengan baik. Ini dapat memformat dokumen sesuai dengan templat yang telah ditentukan dan menyertakan konten atau data tambahan yang diperlukan.
6. Jika salah satu fungsi Lambda gagal, Anda menerima pemberitahuan email melalui Amazon Simple Notification Service (Amazon SNS).

Sepanjang proses ini, AWS Step Functions memastikan bahwa setiap fungsi Lambda dimulai dalam urutan yang benar. Mesin state ini memiliki kapasitas untuk pemrosesan paralel untuk meningkatkan efisiensi. Bucket Amazon S3 bertindak sebagai repositori penyimpanan pusat, mendukung alur kerja dengan mengelola berbagai format media dan dokumen yang terlibat.

Alat

Layanan AWS

- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi AWS Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis.
- Amazon Transcribe adalah layanan pengenalan suara otomatis yang menggunakan model pembelajaran mesin untuk mengonversi audio menjadi teks.

Alat lainnya

- LangChainadalah kerangka kerja untuk mengembangkan aplikasi yang didukung oleh model bahasa besar (LLMs).

Repositori kode

Kode untuk pola ini tersedia di GitHub genai-knowledge-capturerepositori.

Repositori kode berisi file dan folder berikut:

- `assets`folder — Aset statis untuk solusi, seperti diagram arsitektur dan dataset publik
- `code/lambdas`folder - Kode Python untuk semua fungsi Lambda
  - `code/lambdas/generate`folder - Kode Python yang menghasilkan dokumen dari data yang diringkas dalam ember S3
  - `code/lambdas/preprocess`folder - Kode Python yang memproses input untuk mesin status Step Functions
  - `code/lambdas/summarize`folder - Kode Python yang merangkum data yang ditranskripsikan dengan menggunakan layanan Amazon Bedrock
  - `code/lambdas/transcribe`folder - Kode Python yang mengubah data ucapan (file audio) menjadi teks dengan menggunakan Amazon Transcribe
  - `code/lambdas/validate`folder - Kode Python yang memvalidasi apakah semua jawaban berkaitan dengan topik yang sama
- `code/code_stack.py`— AWS CDK membangun file Python yang digunakan untuk membuat sumber daya AWS
- `app.py`— File Python aplikasi AWS CDK yang digunakan untuk menyebarkan sumber daya AWS di akun AWS target
- `requirements.txt`— Daftar semua dependensi Python yang harus diinstal untuk AWS CDK
- `cdk.json`— File input untuk memberikan nilai yang diperlukan untuk membuat sumber daya

Praktik terbaik

Contoh kode yang diberikan adalah untuk proof-of-concept (PoC) atau tujuan percontohan saja. Jika Anda ingin mengambil solusi untuk produksi, gunakan praktik terbaik berikut:

- Aktifkan pencatatan akses Amazon S3
- Aktifkan Log Aliran VPC

Epik

Menyiapkan kredensi AWS di workstation lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekspor variabel untuk akun dan Wilayah AWS.	Untuk memberikan kredensi AWS untuk AWS CDK dengan menggunakan variabel lingkungan, jalankan perintah berikut. `export CDK_DEFAU LT_ACCOUNT=<12-digit AWS account number> export CDK_DEFAU LT_REGION=<Region>`	AWS DevOps, DevOps insinyur
Siapkan profil bernama AWS CLI.	Untuk mengatur profil bernama AWS CLI untuk akun, ikuti petunjuk dalam pengaturan Konfigurasi dan file kredensyal.	AWS DevOps, DevOps insinyur

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repo ke workstation lokal Anda.	Untuk mengkloning genai-knowledge-capturerepositori, jalankan perintah berikut di terminal Anda. `git clone https://github.com/aws-samples/genai-knowledge-capture`	AWS DevOps, DevOps insinyur
(Opsional) Ganti file audio.	Untuk menyesuaikan aplikasi sampel untuk memasukkan data Anda sendiri, lakukan hal berikut: 1. Arahkan ke `assets/audio_samples` folder di repositori kloning. 2. Hapus folder yang berisi file audio sampel. 3. Buat folder untuk setiap topik yang ingin Anda analisis. 4. Transfer file audio Anda ke folder masing-masing.	AWS DevOps, DevOps insinyur

Siapkan lingkungan virtual Python.	Untuk mengatur lingkungan virtual Python, jalankan perintah berikut.
	`cd genai-knowledge-capture`
	`python3 -m venv .venv`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`source .venv/bin/activate` `pip install -r requirements.txt`	
Sintesis kode AWS CDK.	Untuk mengonversi kode ke konfigurasi AWS CloudFormation stack, jalankan perintah berikut. `cdk synth`	AWS DevOps, DevOps insinyur

Konfigurasikan dan terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Penyediaan akses model pondasi.	Aktifkan akses ke model Anthropic Claude 3 Sonnet untuk akun AWS Anda. Untuk petunjuk, lihat Menambahkan akses model dalam dokumentasi Bedrock.	AWS DevOps
Menyebarkan sumber daya di akun.	Untuk menerapkan sumber daya di akun AWS dengan menggunakan AWS CDK, lakukan hal berikut: 1. (Opsional) Di root repositori kloning, dalam `app.py` file, perbarui nama tumpukan CloudFormation AWS. Nama stack default adalah`genai-kno`	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`wledge-capture-sta ck .` 2. Untuk menyebarkan sumber daya, jalankan perintah`cdk deploy`. `cdk deploy`Perintah ini menggunakan konstruks i layer-3 untuk membuat satu set fungsi Lambda, bucket S3, topik Amazon SNS, dan mesin status Step Functions. File audio dalam `assets/audio_sampl es` folder disalin ke bucket S3 selama penerapan. 3. Masuk ke AWS Management Console, lalu buka CloudFormation konsol di https://console.aw s.amazon.com/cloud formation/. 4. Konfirmasikan bahwa tumpukan berhasil digunakan. Untuk petunjukn ya, lihat Meninjau tumpukan Anda di CloudFormation konsol AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Langganan topik Amazon SNS.	Untuk berlangganan topik Amazon SNS untuk pemberitahuan, lakukan hal berikut: 1. Di CloudFormation konsol, di panel navigasi, pilih Tumpukan. 2. Pilih `genai-knowledge-capture-stack` tumpukan. 3. Pilih tab Output. 4. Temukan nama topik Amazon SNS dengan kuncinya. `SNSTopicName` 5. Konfigurasikan alamat email untuk menerima pemberitahuan dengan mengikuti petunjuk di Berlangganan alamat email ke topik Amazon SNS.	AWS Umum

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan mesin negara.	1. Buka Konsol Step Functions. 2. Pada halaman mesin Negara, pilih genai-knowledge-capture-stack-state-machine. 3. Pilih Mulai Eksekusi.	Pengembang aplikasi, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. (Opsional) Di Nama kotak, masukkan nama untuk eksekusi. 5. Di area Input, masukkan objek JSON berikut dengan mengganti teks placeholder, di mana: • <Name>adalah apa yang ingin Anda beri nama dokumen. • <S3 bucket name>adalah nama bucket Amazon S3 yang berisi file audio. • <Folder path>adalah direktori yang berisi file audio. `{ "documentName": "<Name>", "audioFileFolderUri": "s3://<S3 bucket name>/<Folder path>" }` 6. Pilih Mulai Eksekusi. 7. Pada halaman detail eksekusi, tinjau hasilnya dan tunggu eksekusi selesai.	

Bersihkan semua sumber daya AWS dalam solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya AWS.	Setelah Anda menguji solusinya, bersihkan sumber daya: 1. Hapus semua objek dari bucket S3, lalu hapus bucket. Untuk informasi selengkapnya, lihat Menghapus bucket. 2. Dari repositori kloning, jalankan perintah. `cdk destroy`	AWS DevOps, DevOps insinyur

Sumber daya terkait

Dokumentasi AWS

- Sumber daya Amazon Bedrock:
  - Akses model
  - Parameter inferensi untuk model pondasi
- Sumber daya AWS CDK:
  - Memulai AWS CDK
  - Bekerja dengan AWS CDK dengan Python
  - Memecahkan masalah umum AWS CDK
  - Perintah Toolkit
- Sumber daya AWS Step Functions:
  - Memulai AWS Step Functions
  - Pemecahan Masalah
- Membangun fungsi Lambda dengan Python
- Pembuat Aplikasi AI Generatif di AWS

Sumber daya lainnya

- LangChain dokumentasi

Buat rekomendasi yang dipersonalisasi dan diberi peringkat ulang menggunakan Amazon Personalize

Mason Cahill, Matthew Chasse, dan Tayo Olajide, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara menggunakan Amazon Personalize untuk menghasilkan rekomendasi yang dipersonalisasi—termasuk rekomendasi peringkat ulang—untuk pengguna Anda berdasarkan konsumsi data interaksi pengguna real-time dari pengguna tersebut. Contoh skenario yang digunakan dalam pola ini didasarkan pada situs web adopsi hewan peliharaan yang menghasilkan rekomendasi untuk penggunanya berdasarkan interaksi mereka (misalnya, hewan peliharaan apa yang dikunjungi pengguna). Dengan mengikuti skenario contoh, Anda belajar menggunakan Amazon Kinesis Data Streams untuk menyerap data interaksi, AWS Lambda untuk menghasilkan rekomendasi dan memberi peringkat ulang rekomendasi, dan Amazon Data Firehose untuk menyimpan data dalam bucket Amazon Simple Storage Service (Amazon S3). Anda juga belajar menggunakan AWS Step Functions untuk membuat mesin status yang mengelola versi solusi (yaitu, model terlatih) yang menghasilkan rekomendasi Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan AWS Cloud Development Kit (AWS CDK) yang di-bootstrap
- AWS Command Line Interface (AWS CLI) dengan kredensi yang dikonfigurasi
- Python 3.9

Versi produk

- Python 3.9
- AWS CDK 2.23.0 atau yang lebih baru
- AWS CLI 2.7.27 atau yang lebih baru

Arsitektur

Tumpukan teknologi

- Amazon Data Firehose
- Amazon Kinesis Data Streams
- Amazon Personalize
- Amazon Simple Storage Service (Amazon S3)
- AWS Cloud Development Kit (AWS CDK)
- Antarmuka Baris Perintah AWS (AWS CLI)
- AWS Lambda
- AWS Step Functions

Arsitektur target

Diagram berikut mengilustrasikan pipeline untuk memasukkan data real-time ke Amazon Personalize. Pipeline kemudian menggunakan data tersebut untuk menghasilkan rekomendasi yang dipersonalisasi dan diberi peringkat ulang bagi pengguna.

Diagram menunjukkan alur kerja berikut:

1. Kinesis Data Streams menyerap data pengguna real-time (misalnya, peristiwa seperti hewan peliharaan yang dikunjungi) untuk diproses oleh Lambda dan Firehose.
2. Fungsi Lambda memproses rekaman dari Kinesis Data Streams dan membuat panggilan API untuk menambahkan interaksi pengguna dalam rekaman ke pelacak peristiwa di Amazon Personalize.
3. Aturan berbasis waktu memanggil mesin status Step Functions dan menghasilkan versi solusi baru untuk model rekomendasi dan peringkat ulang dengan menggunakan peristiwa dari pelacak peristiwa di Amazon Personalize.
4. Kampanye Amazon Personalisasi diperbarui oleh mesin status untuk menggunakan versi solusi baru.
5. Lambda memberi peringkat ulang daftar item yang direkomendasikan dengan memanggil kampanye peringkat ulang Amazon Personalize.
6. Lambda mengambil daftar item yang direkomendasikan dengan memanggil kampanye rekomendasi Amazon Personalize.
7. Firehose menyimpan peristiwa ke bucket S3 di mana mereka dapat diakses sebagai data historis.

Alat

Alat AWS

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Data Firehose membantu Anda mengirimkan data streaming real-time ke layanan AWS lainnya, titik akhir HTTP kustom, dan titik akhir HTTP yang dimiliki oleh penyedia layanan pihak ketiga yang didukung.
- Amazon Kinesis Data Streams membantu Anda mengumpulkan dan memproses aliran besar catatan data secara real time.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Personalize adalah layanan machine learning (ML) yang dikelola sepenuhnya yang membantu Anda menghasilkan rekomendasi item untuk pengguna berdasarkan data Anda.
- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis.

Alat-alat lainnya

- pytest adalah kerangka kerja Python untuk menulis tes kecil yang dapat dibaca.
- Python adalah bahasa pemrograman komputer tujuan umum.

Kode

Kode untuk pola ini tersedia di repositori GitHub Animal Recommender. Anda dapat menggunakan CloudFormation template AWS dari repositori ini untuk menerapkan sumber daya untuk solusi contoh.

> **Note**
>
> Versi solusi Amazon Personalisasi, pelacak peristiwa, dan kampanye didukung oleh sumber daya khusus (dalam infrastruktur) yang memperluas sumber daya asli. CloudFormation

Epik

Buat infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat lingkungan Python yang terisolasi.	Pengaturan Mac/Linux 1. Untuk membuat lingkungan virtual secara manual, jalankan `$ python3 -m venv .venv` perintah dari terminal Anda. 2. Setelah proses init selesai, jalankan `$ source .venv/bin/activate` perintah untuk mengaktifkan lingkungan virtual. Pengaturan Windows Untuk membuat lingkungan virtual secara manual, jalankan `% .venv\Scripts\activate.bat` perintah dari terminal Anda.	DevOps insinyur
Sintesis CloudFormation template.	1. Untuk menginstal dependensi yang diperlukan, jalankan `$ pip`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`install -r requirements.txt` perintah dari terminal Anda. 2. Di AWS CLI, atur variabel lingkungan berikut: • `export ACCOUNT_ID=123456789` • `export CDK_DEPLOY_REGION=us-east-1` • `export CDK_ENVIRONMENT=dev` 3. Dalam `config/{env}.yml` file, perbarui `vpcId` agar sesuai dengan ID virtual private cloud (VPC) Anda. 4. Untuk mensintesis CloudFormation template untuk kode ini, jalankan `$ cdk synth` perintah. **Note** Pada langkah 2, `CDK_ENVIRONMENT` mengacu pada `config/{env}.yml` file.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan sumber daya dan membuat infrastruktur.	Untuk menyebarkan sumber daya solusi, jalankan `./deploy.sh` perintah dari terminal Anda. Perintah ini menginstal dependensi Python yang diperlukan. Skrip Python membuat bucket S3 dan kunci AWS Key Managemen t Service (AWS KMS), lalu menambahkan data seed untuk pembuatan model awal. Akhirnya, skrip berjalan `cdk deploy` untuk membuat infrastruktur yang tersisa. **Note** Pelatihan model awal terjadi selama pembuatan tumpukan. Diperlukan waktu hingga dua jam agar tumpukan selesai dibuat.	DevOps insinyur

Sumber daya terkait

- Rekomendasi Hewan () GitHub
- Dokumentasi Referensi AWS CDK
- Dokumentasi Boto3

- Optimalkan rekomendasi yang dipersonalisasi untuk metrik bisnis pilihan Anda dengan Amazon Personalize (AWS Machine Learning Blog)

Informasi tambahan

Contoh muatan dan tanggapan

Rekomendasi fungsi Lambda

Untuk mengambil rekomendasi, kirimkan permintaan ke fungsi Lambda rekomendasi dengan muatan dalam format berikut:

```
{
  "userId": "3578196281679609099",
  "limit": 6
}
```

Contoh respon berikut berisi daftar kelompok hewan:

```
[{"id": "1-domestic short hair-1-1"},
{"id": "1-domestic short hair-3-3"},
{"id": "1-domestic short hair-3-2"},
{"id": "1-domestic short hair-1-2"},
{"id": "1-domestic short hair-3-1"},
{"id": "2-beagle-3-3"},
```

Jika Anda meninggalkan `userId` bidang, fungsi mengembalikan rekomendasi umum.

Peringkat ulang fungsi Lambda

Untuk menggunakan peringkat ulang, kirimkan permintaan ke fungsi Lambda peringkat ulang. Muatan berisi semua item `userId` IDs yang akan diberi peringkat ulang dan metadatanya. Contoh data berikut menggunakan kelas Oxford Pets untuk `animal_species_id` (1=cat, 2=dog) dan bilangan bulat 1-5 untuk dan: `animal_age_id` `animal_size_id`

```
{
   "userId":"12345",
   "itemMetadataList":[
      {
         "itemId":"1",
```

```
         "animalMetadata":{
            "animal_species_id":"2",
            "animal_primary_breed_id":"Saint_Bernard",
            "animal_size_id":"3",
            "animal_age_id":"2"
         }
      },
      {
         "itemId":"2",
         "animalMetadata":{
            "animal_species_id":"1",
            "animal_primary_breed_id":"Egyptian_Mau",
            "animal_size_id":"1",
            "animal_age_id":"1"
         }
      },
      {
         "itemId":"3",
         "animalMetadata":{
            "animal_species_id":"2",
            "animal_primary_breed_id":"Saint_Bernard",
            "animal_size_id":"3",
            "animal_age_id":"2"
         }
      }
   ]
}
```

Fungsi Lambda memberi peringkat ulang item ini, lalu mengembalikan daftar terurut yang menyertakan item IDs dan respons langsung dari Amazon Personalize. Ini adalah daftar peringkat kelompok hewan tempat item berada dan skornya. Amazon Personalize menggunakan resep Personalisasi Pengguna dan Personalisasi Peringkat untuk menyertakan skor untuk setiap item dalam rekomendasi. Skor ini mewakili kepastian relatif yang dimiliki Amazon Personalize tentang item mana yang akan dipilih pengguna selanjutnya. Skor yang lebih tinggi mewakili kepastian yang lebih besar.

```
{
   "ranking":[
      "1",
      "3",
      "2"
   ],
```

```
   "personalizeResponse":{
      "ResponseMetadata":{
         "RequestId":"a2ec0417-9dcd-4986-8341-a3b3d26cd694",
         "HTTPStatusCode":200,
         "HTTPHeaders":{
            "date":"Thu, 16 Jun 2022 22:23:33 GMT",
            "content-type":"application/json",
            "content-length":"243",
            "connection":"keep-alive",
            "x-amzn-requestid":"a2ec0417-9dcd-4986-8341-a3b3d26cd694"
         },
         "RetryAttempts":0
      },
      "personalizedRanking":[
         {
            "itemId":"2-Saint_Bernard-3-2",
            "score":0.8947961
         },
         {
            "itemId":"1-Siamese-1-1",
            "score":0.105204
         }
      ],
      "recommendationId":"RID-d97c7a87-bd4e-47b5-a89b-ac1d19386aec"
   }
}
```

Payload Amazon Kinesis

Payload untuk dikirim ke Amazon Kinesis memiliki format berikut:

```
{
    "Partitionkey": "randomstring",
    "Data": {
        "userId": "12345",
        "sessionId": "sessionId4545454",
        "eventType": "DetailView",
        "animalMetadata": {
            "animal_species_id": "1",
            "animal_primary_breed_id": "Russian_Blue",
            "animal_size_id": "1",
            "animal_age_id": "2"
        },
        "animal_id": "98765"
```

```
    }
}
```

> **Note**
>
> `userId`Bidang dihapus untuk pengguna yang tidak diautentikasi.

Merampingkan alur kerja pembelajaran mesin dari pengembangan lokal hingga eksperimen yang dapat diskalakan dengan menggunakan SageMaker AI dan Hydra

David Sauerwein, Marco Geiger, dan Julian Ferdinand Grueber, Amazon Web Services

Ringkasan

Pola ini memberikan pendekatan terpadu untuk mengonfigurasi dan menjalankan algoritma pembelajaran mesin (ML) dari pengujian lokal hingga produksi di Amazon AI. SageMaker Algoritma ML adalah fokus dari pola ini, tetapi pendekatannya meluas ke fitur rekayasa, inferensi, dan seluruh jaringan pipa ML. Pola ini menunjukkan transisi dari pengembangan skrip lokal ke pekerjaan pelatihan SageMaker AI melalui kasus penggunaan sampel.

Alur kerja MS yang khas adalah mengembangkan dan menguji solusi pada mesin lokal, menjalankan eksperimen skala besar (misalnya, dengan parameter berbeda) di cloud, dan menerapkan solusi yang disetujui di cloud. Kemudian, solusi yang diterapkan harus dipantau dan dipelihara. Tanpa pendekatan terpadu untuk alur kerja ini, pengembang sering perlu memfaktorkan ulang kode mereka di setiap tahap. Jika solusinya bergantung pada sejumlah besar parameter yang mungkin berubah pada setiap tahap alur kerja ini, akan menjadi semakin sulit untuk tetap terorganisir dan konsisten.

Pola ini mengatasi tantangan-tantangan ini. Pertama, ini menghilangkan kebutuhan akan refactoring kode antar lingkungan dengan menyediakan alur kerja terpadu yang tetap konsisten baik berjalan di mesin lokal, dalam wadah, atau pada AI. SageMaker Kedua, menyederhanakan manajemen parameter melalui sistem konfigurasi Hydra, di mana parameter didefinisikan dalam file konfigurasi terpisah yang dapat dengan mudah dimodifikasi dan digabungkan, dengan logging otomatis dari setiap konfigurasi run. Untuk detail selengkapnya tentang cara pola ini mengatasi tantangan ini, lihat Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Peran pengguna AWS Identity and Access Management (IAM) untuk menerapkan dan memulai pekerjaan pelatihan SageMaker AI
- AWS Command Line Interface (AWS CLI) versi 2.0 atau yang lebih baru diinstal dan dikonfigurasi

- Puisi versi 1.8 atau lebih baru, tetapi lebih awal dari 2.0, diinstal
- Docker diinstal
- Python versi 3.10.x

Batasan

- Kode saat ini hanya menargetkan pekerjaan pelatihan SageMaker AI. Memperluasnya ke pekerjaan pemrosesan dan seluruh saluran SageMaker AI sangatlah mudah.
- Untuk pengaturan SageMaker AI yang sepenuhnya diproduksi, detail tambahan harus ada. Contohnya bisa berupa kustom AWS Key Management Service (AWS KMS) kunci untuk komputasi dan penyimpanan, atau konfigurasi jaringan. Anda juga dapat mengonfigurasi opsi tambahan ini dengan menggunakan Hydra di subfolder khusus folder. `config`
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menggambarkan arsitektur solusi.

Diagram menunjukkan alur kerja berikut:

1. Ilmuwan data dapat mengulangi algoritma dalam skala kecil di lingkungan lokal, menyesuaikan parameter, dan menguji skrip pelatihan dengan cepat tanpa perlu Docker atau SageMaker AI. (Untuk detail selengkapnya, lihat tugas “Jalankan secara lokal untuk pengujian cepat” di Epik.)
2. Setelah puas dengan algoritme, ilmuwan data membangun dan mendorong image Docker ke repositori Amazon Elastic Container Registry (Amazon ECR) bernama. `hydra-sm-artifact` (Untuk detail selengkapnya, lihat “Jalankan alur kerja pada SageMaker AI” di Epics.)
3. Ilmuwan data memulai pekerjaan pelatihan SageMaker AI atau pekerjaan optimasi hyperparameter (HPO) dengan menggunakan skrip Python. Untuk pekerjaan pelatihan reguler, konfigurasi yang disesuaikan ditulis ke bucket Amazon Simple Storage Service (Amazon S3) bernama. `hydra-sample-config` Untuk pekerjaan HPO, set konfigurasi default yang terletak di `config` folder diterapkan.
4. Pekerjaan pelatihan SageMaker AI menarik image Docker, membaca data input dari bucket Amazon S3, dan mengambil konfigurasi dari `hydra-sample-data` bucket Amazon S3 atau

menggunakan konfigurasi default`hydra-sample-config`. Setelah pelatihan, pekerjaan menyimpan data output ke bucket Amazon S3. `hydra-sample-data`

Otomatisasi dan skala

- Untuk pelatihan otomatis, pelatihan ulang, atau inferensi, Anda dapat mengintegrasikan AWS CLI kode dengan layanan seperti AWS Lambda, AWS CodePipeline, atau Amazon. EventBridge
- Penskalaan dapat dicapai dengan mengubah konfigurasi untuk ukuran instance atau dengan menambahkan konfigurasi untuk pelatihan terdistribusi.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda. Untuk pola ini, AWS CLI berguna untuk konfigurasi sumber daya awal dan pengujian.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon SageMaker AI adalah layanan pembelajaran mesin terkelola (ML) yang membantu Anda membangun dan melatih model ML, lalu menerapkannya ke lingkungan host yang siap produksi. SageMaker Pelatihan AI adalah layanan ML yang dikelola sepenuhnya dalam SageMaker AI yang memungkinkan pelatihan model ML dalam skala besar. Alat ini dapat menangani tuntutan komputasi model pelatihan secara efisien, memanfaatkan skalabilitas bawaan dan integrasi dengan yang lain. Layanan AWS SageMaker Pelatihan AI juga mendukung algoritme dan wadah khusus, membuatnya fleksibel untuk berbagai alur kerja ML.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah. Ini digunakan

dalam pola ini untuk memastikan lingkungan yang konsisten di berbagai tahap, mulai dari pengembangan hingga penerapan, dan untuk mengemas dependensi dan kode dengan andal. Kontainerisasi Docker memungkinkan penskalaan dan kontrol versi yang mudah di seluruh alur kerja.

- Hydra adalah alat manajemen konfigurasi yang memberikan fleksibilitas untuk menangani beberapa konfigurasi dan manajemen sumber daya dinamis. Ini berperan penting dalam mengelola konfigurasi lingkungan, memungkinkan penyebaran yang mulus di berbagai lingkungan. Untuk detail selengkapnya tentang Hydra, lihat Informasi tambahan.
- Python adalah bahasa pemrograman komputer tujuan umum. Python digunakan untuk menulis kode ML dan alur kerja penerapan.
- Puisi adalah alat untuk manajemen ketergantungan dan pengemasan dengan Python.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub configuring-sagemaker-training-jobs-with-hydra.

Praktik terbaik

- Pilih peran IAM untuk menerapkan dan memulai pekerjaan pelatihan SageMaker AI yang mengikuti prinsip hak istimewa paling sedikit dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.
- Gunakan kredensi sementara untuk mengakses peran IAM di terminal.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan aktifkan lingkungan virtual.	Untuk membuat dan mengaktifkan lingkungan virtual, jalankan perintah berikut di root repositori: `poetry install`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`poetry shell`	
Menyebarkan infrastruktur.	Untuk menyebarkan infrastruktur menggunakan CloudFormation, jalankan perintah berikut: `aws cloudformation deploy --template-file infra/hydra-sagemaker-setup.yaml --stack-name hydra-sagemaker-setup --capabilities CAPABILITY_NAMED_IAM`	AWS Umum, DevOps insinyur
Unduh data sampel.	Untuk men-download data input dari openmlke mesin lokal Anda, jalankan perintah berikut: `python scripts/download_data.py`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan secara lokal untuk pengujian cepat.	Untuk menjalankan kode pelatihan secara lokal untuk pengujian, jalankan perintah berikut: `python mypackage/train.py data.train_data_path=data/train.csv evaluation.base_dir_path=data` Log dari semua eksekusi disimpan oleh waktu eksekusi dalam folder bernama`outputs`. Untuk informasi lebih lanjut, lihat bagian “Output” di GitHub repositori. Anda juga dapat melakukan beberapa pelatihan secara paralel, dengan parameter yang berbeda, dengan menggunakan `--multirun` fungsionalitas. Untuk detail selengkapnya, lihat dokumentasi Hydra.	Ilmuwan data

Jalankan alur kerja di AI SageMaker

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur variabel lingkungan.	Untuk menjalankan pekerjaan Anda di SageMaker AI, atur variabel lingkungan berikut,	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	berikan Akun AWS ID Anda Wilayah AWS dan Anda:
	
	`export ECR_REPO_NAME=hydra-sm-artifact`
	`export image_tag=latest`
	`export AWS_REGION="<your_aws_region>" # for instance, us-east-1`
	`export ACCOUNT_ID="<your_account_id>"`
	`export BUCKET_NAME_DATA=hydra-sample-data-$ACCOUNT_ID`
	`export BUCKET_NAME_CONFIG=hydra-sample-config-$ACCOUNT_ID`
	`export AWS_DEFAULT_REGION=$AWS_REGION`
	`export ROLE_ARN=arn:aws:iam::${ACCOUNT_ID}:role/hydra-sample-sagemaker`
	`export INPUT_DATA_S3_PATH=s3://$BUCKET_NAME_DATA/hydra-on-sm/input/`
	`export OUTPUT_DATA_S3_PATH=s3://$BUCKET_NAME_DATA/hydra-on-sm/output/`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat dan dorong gambar Docker.	Untuk membuat image Docker dan mendorongnya ke repositori Amazon ECR, jalankan perintah berikut:
	`chmod +x scripts/create_and_push_image.sh`
	`scripts/create_and_push_image.sh $ECR_REPO_NAME $image_tag $AWS_REGION $ACCOUNT_ID`
	Tugas ini mengasumsikan bahwa Anda memiliki kredensi yang valid di lingkungan Anda. Gambar Docker didorong ke repositori Amazon ECR yang ditentukan dalam variabel lingkungan di tugas sebelumnya dan digunakan untuk mengaktifkan wadah SageMaker AI tempat pekerjaan pelatihan akan berjalan.
		Insinyur ML, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin data input ke Amazon S3.	Pekerjaan pelatihan SageMaker AI perlu mengambil data input. Untuk menyalin data input ke bucket Amazon S3 untuk data, jalankan perintah berikut: `aws s3 cp data/train.csv "${INPUT_DATA_S3_PATH}train.csv"`	Insinyur data, AWS Umum
Kirimkan pekerjaan pelatihan SageMaker AI.	Untuk menyederhanakan eksekusi skrip Anda, tentukan parameter konfigurasi default dalam file. `default.yaml` Selain memastikan konsistensi di seluruh proses, pendekatan ini juga menawarkan fleksibilitas untuk dengan mudah mengganti pengaturan default sesuai kebutuhan. Lihat contoh berikut ini: `python scripts/start_sagemaker_training_job.py sagemaker.role_arn=$ROLE_ARN sagemaker.config_s3_bucket=$BUCKET_NAME_CONFIG sagemaker.input_data_s3_path=$INPUT_DATA_S3_PATH sagemaker.output_data_s3_path=$OUTPUT_DATA_S3_PATH`	AWS Umum, insinyur ML, Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan tuning hyperparameter SageMaker AI.	Menjalankan tuning hyperparameter SageMaker AI mirip dengan mengirimkan pekerjaan pelatihan AII. SageMaker Namun, skrip eksekusi berbeda dalam beberapa hal penting seperti yang Anda lihat di file start_sagemaker_hpo_job.py. Hyperparameter yang akan disetel harus melewati muatan boto3, bukan saluran ke pekerjaan pelatihan. Untuk memulai pekerjaan optimasi hyperparameter (HPO), jalankan perintah berikut: `python scripts/start_sagemaker_hpo_job.py sagemaker.role_arn=$ROLE_ARN sagemaker.config_s3_bucket=$BUCKET_NAME_CONFIG sagemaker.input_data_s3_path=$INPUT_DATA_S3_PATH sagemaker.output_data_s3_path=$OUTPUT_DATA_S3_PATH`	Ilmuwan data

Pemecahan masalah

Isu	Solusi
Token kadaluwarsa	Ekspor AWS kredensi baru.
Kurangnya izin IAM	Pastikan Anda mengekspor kredensional peran IAM yang memiliki semua izin IAM yang diperlukan untuk menerapkan CloudForm ation template dan untuk memulai pekerjaan pelatihan AI. SageMaker

Sumber daya terkait

- Latih model dengan Amazon SageMaker AI (AWS dokumentasi)
- Apa itu Tuning Hyperparameter?

Informasi tambahan

Pola ini membahas tantangan berikut:

Konsistensi dari pengembangan lokal hingga penerapan skala besar — Dengan pola ini, pengembang dapat menggunakan alur kerja yang sama, terlepas dari apakah mereka menggunakan skrip Python lokal, menjalankan kontainer Docker lokal, melakukan eksperimen besar pada AI, atau menerapkan dalam produksi pada SageMaker AI. SageMaker Konsistensi ini penting karena alasan berikut:

- Iterasi lebih cepat — Ini memungkinkan eksperimen lokal yang cepat tanpa perlu penyesuaian besar saat meningkatkan skala.
- Tidak ada refactoring - Transisi ke eksperimen yang lebih besar pada SageMaker AI mulus, tidak memerlukan perombakan pengaturan yang ada.
- Peningkatan berkelanjutan - Mengembangkan fitur baru dan terus meningkatkan algoritme sangatlah mudah karena kodenya tetap sama di seluruh lingkungan.

Manajemen konfigurasi — Pola ini menggunakan Hydra, alat manajemen konfigurasi, untuk memberikan manfaat berikut:

- Parameter didefinisikan dalam file konfigurasi, terpisah dari kode.
- Set parameter yang berbeda dapat ditukar atau digabungkan dengan mudah.
- Pelacakan eksperimen disederhanakan karena setiap konfigurasi run dicatat secara otomatis.
- Eksperimen cloud dapat menggunakan struktur konfigurasi yang sama dengan proses lokal, memastikan konsistensi.

Dengan Hydra, Anda dapat mengelola konfigurasi secara efektif, mengaktifkan fitur-fitur berikut:

- Membagi konfigurasi - Pecahkan konfigurasi proyek Anda menjadi potongan-potongan yang lebih kecil dan dapat dikelola yang dapat dimodifikasi secara independen. Pendekatan ini membuatnya lebih mudah untuk menangani proyek yang kompleks.
- Sesuaikan default dengan mudah — Ubah konfigurasi dasar Anda dengan cepat, sehingga lebih mudah untuk menguji ide-ide baru.
- Sejajarkan input CLI dan file konfigurasi - Gabungkan input baris perintah dengan file konfigurasi Anda dengan lancar. Pendekatan ini mengurangi kekacauan dan kebingungan, membuat proyek Anda lebih mudah dikelola dari waktu ke waktu.

Terjemahkan bahasa alami ke dalam kueri DSL untuk OpenSearch dan kueri Elasticsearch

Tabby Ward, Nicholas Switzer, dan Breanne Warner, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menggunakan model bahasa besar (LLMs) untuk mengubah kueri bahasa alami menjadi bahasa khusus domain kueri (query DSL), yang memudahkan pengguna untuk berinteraksi dengan layanan pencarian seperti OpenSearch dan Elasticsearch tanpa pengetahuan luas tentang bahasa kueri. Sumber daya ini sangat berharga bagi pengembang dan ilmuwan data yang ingin meningkatkan aplikasi berbasis pencarian dengan kemampuan kueri bahasa alami, yang pada akhirnya meningkatkan pengalaman pengguna dan fungsionalitas pencarian.

Pola ini menggambarkan teknik untuk rekayasa yang cepat, penyempurnaan berulang, dan penggabungan pengetahuan khusus, yang semuanya sangat penting dalam pembuatan data sintetis. Meskipun pendekatan ini berfokus terutama pada konversi kueri, pendekatan ini secara implisit menunjukkan potensi augmentasi data dan produksi data sintetis yang dapat diskalakan. Fondasi ini dapat diperluas ke tugas pembuatan data sintetis yang lebih komprehensif, untuk menyoroti kekuatan LLMs dalam menjembatani input bahasa alami yang tidak terstruktur dengan output khusus aplikasi yang terstruktur.

Solusi ini tidak melibatkan alat migrasi atau penerapan dalam pengertian tradisional. Sebaliknya, ini berfokus pada mendemonstrasikan bukti konsep (PoC) untuk mengonversi kueri bahasa alami menjadi kueri DSL dengan menggunakan. LLMs

- Pola menggunakan notebook Jupyter sebagai step-by-step panduan untuk mengatur lingkungan dan mengimplementasikan konversi. text-to-query
- Ini menggunakan Amazon Bedrock untuk mengakses LLMs, yang sangat penting untuk menafsirkan bahasa alami dan menghasilkan kueri yang sesuai.
- Solusinya dirancang untuk bekerja dengan Amazon OpenSearch Service. Anda dapat mengikuti proses serupa untuk Elasticsearch, dan kueri yang dihasilkan berpotensi diadaptasi untuk mesin pencari serupa.

Query DSL adalah bahasa penelusuran berbasis JSON yang fleksibel yang digunakan untuk membangun kueri kompleks di Elasticsearch dan. OpenSearch Ini memungkinkan Anda untuk menentukan kueri dalam parameter kueri operasi pencarian, dan mendukung berbagai jenis kueri.

Kueri DSL mencakup kueri daun dan kueri majemuk. Kueri daun mencari nilai tertentu di bidang tertentu dan mencakup teks lengkap, tingkat istilah, geografis, penggabungan, rentang, dan kueri khusus. Kueri majemuk bertindak sebagai pembungkus untuk beberapa klausa daun atau majemuk, dan menggabungkan hasilnya atau memodifikasi perilakunya. Query DSL mendukung pembuatan pencarian canggih, mulai dari kueri sederhana yang cocok untuk semua kueri kompleks dan multi-klausa yang menghasilkan hasil yang sangat spesifik. Query DSL sangat berharga untuk proyek yang membutuhkan kemampuan pencarian lanjutan, konstruksi kueri fleksibel, dan struktur kueri berbasis JSON.

Pola ini menggunakan teknik seperti prompt beberapa tembakan, prompt sistem, output terstruktur, rantai prompt, penyediaan konteks, dan petunjuk khusus tugas untuk konversi DSL. text-to-query Untuk definisi dan contoh teknik ini, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

Untuk menggunakan notebook Jupyter secara efektif untuk mengonversi kueri bahasa alami menjadi kueri DSL kueri, Anda memerlukan:

- Keakraban dengan notebook Jupyter. Pemahaman dasar tentang cara menavigasi dan menjalankan kode di lingkungan notebook Jupyter.
- Lingkungan Python. Lingkungan Python yang berfungsi, lebih disukai Python 3.x, dengan pustaka yang diperlukan diinstal.
- Elasticsearch atau OpenSearch pengetahuan. Pengetahuan dasar tentang Elasticsearch atau OpenSearch, termasuk arsitekturnya dan cara melakukan kueri.
- Akun AWS. Aktif Akun AWS untuk mengakses Amazon Bedrock dan layanan terkait lainnya.
- Perpustakaan dan dependensi. Pemasangan pustaka tertentu yang disebutkan dalam buku catatan, seperti `boto3` untuk AWS interaksi, dan dependensi lain yang diperlukan untuk integrasi LLM.
- Akses model dalam Amazon Bedrock. Pola ini menggunakan tiga Claude LLMs dari Anthropic. Buka konsol Amazon Bedrock dan pilih Akses model. Pada layar berikutnya, pilih Aktifkan model tertentu dan pilih ketiga model ini:
  - Claude 3 Soneta
  - Claude 3.5 Soneta
  - Claude 3 Haiku

- Kebijakan IAM yang tepat dan peran IAM. Untuk menjalankan buku catatan di Akun AWS, peran AWS Identity and Access Management (IAM) Anda memerlukan `SagemakerFullAccess` kebijakan serta kebijakan yang disediakan di bagian Informasi tambahan, yang dapat Anda beri nama`APGtext2querydslpolicy`. Kebijakan ini termasuk berlangganan tiga model Claude yang terdaftar.

Memiliki prasyarat ini memastikan pengalaman yang lancar saat Anda bekerja dengan notebook dan mengimplementasikan fungsionalitasnya. text-to-query

Batasan

- Bukti status konsep. Proyek ini terutama dimaksudkan sebagai bukti konsep (PoC). Ini menunjukkan potensi penggunaan LLMs untuk mengubah kueri bahasa alami menjadi kueri DSL, tetapi mungkin tidak sepenuhnya dioptimalkan atau siap produksi.
- Keterbatasan model:

  Kendala jendela konteks. Saat menggunakan LLMs yang tersedia di Amazon Bedrock, perhatikan batasan jendela konteks:

  Model Claude (per September 2024):

  - Claude 3 Opus: 200.000 token
  - Claude 3 Soneta: 200.000 token
  - Claude 3 Haiku: 200.000 token

  Model lain di Amazon Bedrock mungkin memiliki ukuran jendela konteks yang berbeda. Selalu periksa dokumentasi terbaru untuk informasi terbaru.

  Ketersediaan model.Ketersediaan model tertentu di Amazon Bedrock dapat bervariasi. Pastikan bahwa Anda memiliki akses ke model yang diperlukan sebelum Anda menerapkan solusi ini.
- Keterbatasan tambahan
  - Kompleksitas kueri. Efektivitas bahasa alami untuk kueri konversi DSL dapat bervariasi tergantung pada kompleksitas kueri input.
  - Kompatibilitas versi. Kueri DSL yang dihasilkan mungkin memerlukan penyesuaian berdasarkan versi Elasticsearch tertentu atau OpenSearch yang Anda gunakan.
  - Kinerja. Pola ini menyediakan implementasi PoC, sehingga kecepatan dan akurasi pembuatan kueri mungkin tidak optimal untuk penggunaan produksi skala besar.

- Biaya. Menggunakan LLMs di Amazon Bedrock menimbulkan biaya. Waspadai struktur harga untuk model yang Anda pilih. Untuk informasi selengkapnya, lihat harga Amazon Bedrock.
- Pemeliharaan. Pembaruan rutin untuk petunjuk dan pemilihan model mungkin diperlukan untuk mengikuti kemajuan teknologi LLM dan perubahan sintaks DSL kueri.

Versi produk

Solusi ini diuji di Amazon OpenSearch Service. Jika Anda ingin menggunakan Elasticsearch, Anda mungkin harus membuat beberapa perubahan untuk mereplikasi fungsionalitas yang tepat dari pola ini.

- OpenSearch kompatibilitas versi. OpenSearch mempertahankan kompatibilitas mundur dalam versi utama. Contoh:
  - OpenSearch Klien 1.x umumnya kompatibel dengan cluster OpenSearch 1.x.
  - OpenSearch Klien 2.x umumnya kompatibel dengan cluster OpenSearch 2.x.

  Namun, selalu yang terbaik adalah menggunakan versi minor yang sama untuk klien dan cluster jika memungkinkan.
- OpenSearch Kompatibilitas API. OpenSearch mempertahankan kompatibilitas API dengan Elasticsearch OSS 7.10.2 untuk sebagian besar operasi. Namun, ada beberapa perbedaan, terutama di versi yang lebih baru.
- OpenSearch Pertimbangan upgrade:
  - Penurunan versi langsung tidak didukung. Gunakan snapshot untuk rollback jika diperlukan.
  - Saat Anda memutakhirkan, periksa matriks kompatibilitas dan catatan rilis untuk setiap perubahan yang melanggar.

Pertimbangan Elasticsearch

- Versi Elasticsearch. Versi utama Elasticsearch yang Anda gunakan sangat penting, karena sintaks kueri dan fitur dapat berubah di antara versi utama. Saat ini, versi stabil terbaru adalah Elasticsearch 8.x. Pastikan kueri Anda kompatibel dengan versi Elasticsearch spesifik Anda.
- Versi perpustakaan DSL kueri Elasticsearch. Jika Anda menggunakan pustaka DSL Python kueri Elasticsearch, pastikan versinya cocok dengan versi Elasticsearch Anda. Contoh:
  - Untuk Elasticsearch 8.x, gunakan `elasticsearch-dsl` versi yang lebih besar atau sama dengan 8.0.0 tetapi lebih kecil dari 9.0.0.

  - Untuk Elasticsearch 7.x, gunakan `elasticsearch-dsl` versi yang lebih besar atau sama dengan 7.0.0 tetapi lebih kecil dari 8.0.0.
- Versi pustaka klien. Baik Anda menggunakan klien Elasticsearch resmi atau klien khusus bahasa, pastikan bahwa itu kompatibel dengan versi Elasticsearch Anda.
- Versi kueri DSL. Query DSL berkembang dengan versi Elasticsearch. Beberapa jenis kueri atau parameter mungkin tidak digunakan lagi atau diperkenalkan dalam versi yang berbeda.
- Versi pemetaan. Cara Anda mendefinisikan pemetaan untuk indeks Anda dan mengubah antar versi. Selalu periksa dokumentasi pemetaan untuk versi Elasticsearch spesifik Anda.
- Versi alat analisis. Jika Anda menggunakan penganalisis, tokenizer, atau alat analisis teks lainnya, perilaku atau ketersediaannya mungkin berubah antar versi.

Arsitektur

Arsitektur target

Diagram berikut menggambarkan arsitektur untuk pola ini.

di mana:

1. Input pengguna dan prompt sistem dengan beberapa contoh yang diminta. Prosesnya dimulai dengan pengguna yang menyediakan kueri bahasa alami atau permintaan untuk pembuatan skema.
2. Batuan Dasar Amazon. Input dikirim ke Amazon Bedrock, yang berfungsi sebagai antarmuka untuk mengakses Claude LLM.
3. Claude 3 Soneta LLM. Amazon Bedrock menggunakan Claude 3 Soneta dari keluarga Claude 3 untuk memproses input. LLMs Ini menafsirkan dan menghasilkan Elasticsearch atau OpenSearch query DSL yang sesuai. Untuk permintaan skema, ini menghasilkan Elasticsearch sintetis atau pemetaan. OpenSearch
4. Generasi kueri DSL. Untuk kueri bahasa alami, aplikasi mengambil output LLM dan memformatnya menjadi DSL kueri Elasticsearch atau OpenSearch Service yang valid.
5. Pembuatan data sintetis. Aplikasi ini juga mengambil skema untuk membuat Elasticsearch sintetis atau OpenSearch data yang akan dimuat ke dalam koleksi Tanpa OpenSearch Server untuk pengujian.

6. OpenSearch atau Elasticsearch. Kueri DSL yang dihasilkan ditanyakan terhadap koleksi OpenSearch Tanpa Server di semua indeks. Output JSON berisi data yang relevan dan jumlah klik dari data yang berada dalam koleksi Tanpa OpenSearch Server.

Otomatisasi dan skala

Kode yang disediakan dengan pola ini dibangun secara ketat untuk tujuan PoC. Daftar berikut memberikan beberapa saran untuk mengotomatisasi dan menskalakan solusi lebih lanjut dan memindahkan kode ke produksi. Peningkatan ini berada di luar cakupan pola ini.

- Kontainerisasi:
  - Dockerize aplikasi untuk memastikan konsistensi di berbagai lingkungan.
  - Gunakan platform orkestrasi kontainer seperti Amazon Elastic Container Service (Amazon ECS) atau Kubernetes untuk penerapan yang dapat diskalakan.
- Arsitektur tanpa server:
  - Ubah fungsionalitas inti menjadi AWS Lambda fungsi.
  - Gunakan Amazon API Gateway untuk membuat RESTful titik akhir untuk input kueri bahasa alami.
- Pemrosesan asinkron:
  - Menerapkan Amazon Simple Queue Service (Amazon SQS) untuk mengantri kueri yang masuk.
  - Gunakan AWS Step Functions untuk mengatur alur kerja kueri pemrosesan dan menghasilkan kueri DSL.
- Caching:
  - Menerapkan mekanisme untuk men-cache petunjuknya.
- Pemantauan dan pencatatan:
  - Gunakan Amazon CloudWatch untuk pemantauan dan peringatan.
  - Menerapkan logging terpusat dengan Amazon CloudWatch Logs atau Amazon OpenSearch Service untuk analisis log.
- Peningkatan keamanan:
  - Menerapkan peran IAM untuk kontrol akses berbutir halus.
  - Gunakan AWS Secrets Manager untuk menyimpan dan mengelola kunci API dan kredensyal dengan aman.
- Penyebaran Multi-Wilayah:

- Pertimbangkan untuk menerapkan solusi di beberapa Wilayah AWS untuk meningkatkan latensi dan pemulihan bencana.
- Gunakan Amazon Route 53 untuk perutean permintaan cerdas.

Dengan menerapkan saran ini, Anda dapat mengubah PoC ini menjadi solusi yang kuat, terukur, dan siap produksi. Kami menyarankan Anda menguji secara menyeluruh setiap komponen dan seluruh sistem sebelum penerapan penuh.

Alat

Alat

- Notebook Amazon SageMaker AI adalah notebook Jupyter yang dikelola sepenuhnya untuk pengembangan pembelajaran mesin. Pola ini menggunakan notebook sebagai lingkungan interaktif untuk eksplorasi data, pengembangan model, dan eksperimen di Amazon AI. SageMaker Notebook menyediakan integrasi tanpa batas dengan fitur SageMaker AI lainnya dan. Layanan AWS
- Python adalah bahasa pemrograman komputer tujuan umum. Pola ini menggunakan Python sebagai bahasa inti untuk mengimplementasikan solusinya.
- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu. Amazon Bedrock menyediakan akses LLMs untuk pemrosesan bahasa alami. Pola ini menggunakan model Anthropic Claude 3.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda, termasuk Amazon Layanan AWS Bedrock.
- Amazon OpenSearch Service adalah layanan terkelola yang membantu Anda menerapkan, mengoperasikan, dan menskalakan kluster OpenSearch Layanan di AWS Cloud. Pola ini menggunakan OpenSearch Service sebagai sistem target untuk menghasilkan kueri DSL.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Prompt Engineering Text-to-QueryDSL Using Claude 3 Models. Contohnya menggunakan aplikasi media sosial kesehatan yang membuat posting untuk pengguna dan profil pengguna yang terkait dengan aplikasi kesehatan.

Praktik terbaik

Saat bekerja dengan solusi ini, pertimbangkan hal berikut:

- Kebutuhan akan AWS kredensyal dan izin yang tepat untuk mengakses Amazon Bedrock
- Potensi biaya yang terkait dengan penggunaan Layanan AWS dan LLMs
- Pentingnya memahami kueri DSL untuk memvalidasi dan berpotensi memodifikasi kueri yang dihasilkan

Epik

Mengatur lingkungan dan menyiapkan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyiapkan lingkungan pengembangan.	**Note** Untuk instruksi dan kode terperinci untuk ini dan langkah-l angkah lain dalam pola ini, lihat panduan komprehensif di repositori. GitHub 1. Instal paket Python yang diperlukan, termasuk,**`boto3`**,, **`numpy`** **`awscliopensearch-py`**, dan **`requests-aws4auth`** dengan menggunakan pip. 2. Impor modul yang diperluka n seperti `boto3` `json`,`os`,, `opensearch` `RequestsH`	Python, pip, AWS SDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ttpConnection` `dariopensearchpy` , `bulk` dari `Opensearc hpy opensearc hpy.helpers` , `darisagemaker` ,`time`,`random`,`re`, dan `AWS4Auth` dari`requests_ aws4auth` .	
Mengatur AWS akses.	Siapkan klien Amazon Bedrock dan sesi SageMaker AI. Ambil Amazon Resource Name (ARN) untuk SageMaker peran eksekusi AI untuk digunakan nanti dalam membuat OpenSearch koleksi Tanpa Server.	IAM, AWS CLI, Batuan Dasar Amazon, Amazon SageMaker
Muat skema aplikasi kesehatan.	Membaca dan mengurai skema JSON untuk posting kesehatan dan profil pengguna dari file yang telah ditentukan. Ubah skema menjadi string untuk digunakan nanti dalam prompt.	DevOps insinyur, AWS Umum, Python, JSON

Menghasilkan data sintetis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat generator data berbasis LLM.	Menerapkan fungsi generate_ data () untuk memanggil Amazon Bedrock Converse	Python, Amazon Bedrock API, LLM meminta

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	API dengan model Claude 3. Siapkan model IDs untuk Soneta, Soneta 3.5, dan Haiku:
	`model_id_sonnet3_5 = "anthropic.claude-3-5-sonnet-20240620-v1:0"`
	`model_id_sonnet = "anthropic.claude-3-sonnet-20240229-v1:0"`
	`model_id_haiku = "anthropic.claude-3-haiku-20240307-v1:0"`	

Buat posting kesehatan sintetis.	Gunakan fungsi generate_data () dengan prompt pesan tertentu untuk membuat entri pos kesehatan sintetis berdasarkan skema yang disediakan. Panggilan fungsi terlihat seperti ini: `health_post_data = generate_data(bedrock_rt, model_id_sonnet, system_prompt, message_healthpost, inference_config)`	Python, JSON

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat profil pengguna sintetis.	Gunakan fungsi generate_ data () dengan prompt pesan tertentu untuk membuat entri profil pengguna sintetis berdasarkan skema yang disediakan. Ini mirip dengan pembuatan pos kesehatan, tetapi menggunakan prompt yang berbeda.	Python, JSON

Mengatur OpenSearch dan menelan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan koleksi OpenSearch Tanpa Server.	Gunakan Boto3 untuk membuat koleksi OpenSearch Tanpa Server dengan enkripsi, jaringan, dan kebijakan akses yang sesuai. Pembuatan koleksi terlihat seperti ini: `collection = aoss_client.create_collection(name=es_name, type='SEARCH')` Untuk informasi selengkapnya tentang OpenSearch Tanpa Server, lihat dokumentasi.AWS	OpenSearch Tanpa Server, IAM
Tentukan OpenSearch indeks.	Buat indiexes untuk posting kesehatan dan profil pengguna dengan menggunakan OpenSearch klien,	OpenSearch, JSON

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	berdasarkan pemetaan skema yang telah ditentuka n. Pembuatan indeks terlihat seperti ini: `response_health = oss_client.indices .create(healthpost _index, body=heal thpost_body)`	
Muat data ke dalam OpenSearch.	Jalankan fungsi ingest_data () untuk menyisipkan postingan kesehatan sintetis dan profil pengguna secara massal ke dalam indeks masing-ma sing. OpenSearch Fungsi ini menggunakan pembantu massal dari`opensearch-py`: `success, failed = bulk(oss_client, actions)`	Python, OpenSearch API, operasi data massal

Menghasilkan dan menjalankan kueri

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Desain beberapa contoh cepat.	Hasilkan contoh kueri dan pertanyaan bahasa alami yang sesuai dengan menggunak an model Claude 3 untuk dijadikan beberapa contoh untuk pembuatan kueri.	LLM meminta, kueri DSL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Prompt sistem mencakup contoh-contoh ini: `system_prompt_query_generation = [{"text": f"""You are an expert query dsl generator. ... Examples: {example_prompt} ..."""}]`	
Buat konverter text-to-query DSL.	Terapkan prompt sistem, yang mencakup skema, data, dan beberapa contoh tembakan, untuk pembuatan kueri. Gunakan prompt sistem untuk mengonversi kueri bahasa alami ke kueri DSL. Panggilan fungsi terlihat seperti ini: `query_response = generate_data(bedrock_rt, model_id, system_prompt_query_generation, query, inference_config)`	Python, Amazon Bedrock API, LLM meminta

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kueri uji DSL aktif. OpenSearch	Jalankan fungsi query_oss () untuk menjalankan kueri DSL yang dihasilkan terhadap koleksi Tanpa OpenSearch Server dan mengembalikan hasil. Fungsi ini menggunakan metode pencarian OpenSearch klien: `response = oss_client.search(index="_all", body=temp)`	Python, OpenSearch API, kueri DSL

Uji dan evaluasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat set kueri uji.	Gunakan Claude 3 untuk menghasilkan beragam pertanyaan tes berdasarkan data dan skema sintetis: `test_queries = generate_data(bedrock_rt, model_id_sonnet, query_system_prompt, query_prompt, inference_config)`	LLM mendorong
Nilai keakuratan konversi DSL kueri.	Uji kueri DSL yang dihasilkan dengan menjalankan kueri terhadap OpenSearch dan menganalisis hasil yang dikembalikan untuk relevansi dan akurasi. Ini melibatka	Python, analisis data, kueri DSL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n menjalankan kueri dan memeriksa klik: `output = query_oss (response1) print("Response after running query against Opensearch") print(output)`	
Benchmark Claude 3 model.	Bandingkan kinerja model Claude 3 yang berbeda (Haiku, Soneta, Soneta 3.5) untuk pembuatan kueri dalam hal akurasi dan latensi. Untuk membandingkan, ubah `model_id` saat Anda memanggil generate_data () dan ukur waktu eksekusi.	Python, pembandingan kinerja

Membersihkan dan mendokumentasikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembangkan proses pembersihan.	Hapus semua indeks dari koleksi OpenSearch Tanpa Server setelah digunakan.	Python, AWS SDK, API OpenSearch

Sumber daya terkait

- Kueri DSL (OpenSearch dokumentasi)
- Dokumentasi OpenSearch Layanan Amazon
- OpenSearch Koleksi tanpa server
- Dokumentasi Amazon Bedrock

- Dokumentasi Amazon SageMaker AI
- AWS SDK for Python (Boto3) dokumentasi

Informasi tambahan

Kebijakan IAM

Berikut `APGtext2querydslpolicy` kebijakan untuk peran IAM yang digunakan dalam pola ini:

```
{
  "Version": "2012-10-17",
  "Statement": [
    { "Effect": "Allow",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
      "Resource": "*"
    },
    { "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::sagemaker-*",
        "arn:aws:s3:::sagemaker-*/*"
      ]
    },
    { "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/sagemaker/*"
    },
    { "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
```

```
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    },
    { "Effect": "Allow",
      "Action": [
        "aoss:*"
      ],
      "Resource": "*"
    },
    { "Effect": "Allow",
      "Action": [
        "iam:PassRole",
        "sagemaker:*"
      ],
      "Resource": [
        "arn:aws:iam::*:role/*", "*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "sagemaker.amazonaws.com"
          }
      }
    },
    { "Effect": "Allow",
      "Action": [
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "codecommit:ListRepositories"
      ],
      "Resource": "*"
    },
    { "Effect": "Allow",
      "Action": [
        "aws-marketplace:Subscribe"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws-marketplace:ProductId": [
            "prod-6dw3qvchef7zy",
            "prod-m5ilt4siql27k",
```

```
            "prod-ozonys2hmmpeu"
          ]
        }
      }
    },
    { "Effect": "Allow",
      "Action": [
        "aws-marketplace:Unsubscribe",
        "aws-marketplace:ViewSubscriptions"
      ],
      "Resource": "*"
    },
    { "Effect": "Allow",
      "Action": "iam:*",
      "Resource": "*"
    }
  ]
}
```

Teknik cepat dengan model Anthropic Claude 3

Pola ini menunjukkan teknik prompt berikut untuk konversi text-to-query DSL menggunakan model Claude 3.

- Permintaan beberapa tembakan: Few-shot prompt adalah teknik yang ampuh untuk meningkatkan kinerja model Claude 3 di Amazon Bedrock. Pendekatan ini melibatkan penyediaan model dengan sejumlah kecil contoh yang menunjukkan input/output perilaku yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Saat Anda menggunakan model Claude 3 di Amazon Bedrock, beberapa bidikan yang diminta bisa sangat efektif untuk tugas yang memerlukan pemformatan, pola penalaran, atau pengetahuan domain tertentu. Untuk menerapkan teknik ini, Anda biasanya menyusun prompt Anda dengan dua komponen utama: bagian contoh dan kueri yang sebenarnya. Bagian contoh berisi satu atau beberapa input/output pasangan yang menggambarkan tugas, dan bagian kueri menyajikan input baru yang Anda inginkan respons. Metode ini membantu Claude 3 memahami konteks dan format keluaran yang diharapkan, dan seringkali menghasilkan respons yang lebih akurat dan konsisten.

  Contoh:

```
"query": {
  "bool": {
    "must": [
```

```
      {"match": {"post_type": "recipe"}},
      {"range": {"likes_count": {"gte": 100}}},
      {"exists": {"field": "media_urls"}}
    ]
  }
}
Question: Find all recipe posts that have at least 100 likes and include media URLs.
```

- Permintaan sistem: Selain beberapa bidikan yang diminta, model Claude 3 di Amazon Bedrock juga mendukung penggunaan prompt sistem. Prompt sistem adalah cara untuk memberikan konteks, instruksi, atau pedoman keseluruhan untuk model sebelum menyajikannya dengan input pengguna tertentu. Mereka sangat berguna untuk mengatur nada, mendefinisikan peran model, atau menetapkan kendala untuk seluruh percakapan. Untuk menggunakan prompt sistem dengan Claude 3 di Amazon Bedrock, Anda memasukkannya ke dalam `system` parameter permintaan API Anda. Ini terpisah dari pesan pengguna dan berlaku untuk seluruh interaksi. Permintaan sistem terperinci digunakan untuk mengatur konteks dan memberikan pedoman untuk model.

  Contoh:

```
You are an expert query dsl generator. Your task is to take an input question and
 generate a query dsl to answer the question. Use the schemas and data below to
 generate the query.

Schemas: [schema details]
Data: [sample data]
Guidelines:
- Ensure the generated query adheres to DSL query syntax
- Do not create new mappings or other items that aren't included in the provided
 schemas.
```

- Output terstruktur: Anda dapat menginstruksikan model untuk memberikan output dalam format tertentu, seperti JSON atau dalam tag XHTML.

  Contoh:

```
Put the query in json tags
```

- Prompt chaining: Notebook menggunakan output dari satu panggilan LLM sebagai input untuk yang lain, seperti menggunakan data sintetis yang dihasilkan untuk membuat contoh pertanyaan.

- Ketentuan konteks: Konteks yang relevan, termasuk skema dan data sampel, disediakan dalam petunjuk.

  Contoh:

  ```
  Schemas: [schema details]
  Data: [sample data]
  ```

- Permintaan khusus tugas: Permintaan yang berbeda dibuat untuk tugas tertentu, seperti menghasilkan data sintetis, membuat contoh pertanyaan, dan mengonversi kueri bahasa alami menjadi kueri DSL.

  Contoh untuk menghasilkan pertanyaan tes:

  ```
  Your task is to generate 5 example questions users can ask the health app based on
   provided schemas and data. Only include the questions generated in the response.
  ```

Gunakan Amazon Q Developer sebagai asisten pengkodean untuk meningkatkan produktivitas Anda

Ram Kandaswamy, Amazon Web Services

Ringkasan

Pola ini menggunakan tic-tac-toe game untuk menunjukkan bagaimana Anda dapat menerapkan Amazon Q Developer di berbagai tugas pengembangan. Ini menghasilkan kode untuk tic-tac-toe game sebagai aplikasi satu halaman (SPA), meningkatkan UI, dan membuat skrip untuk menyebarkan aplikasi. AWS

Amazon Q Developer berfungsi sebagai asisten pengkodean untuk membantu mempercepat alur kerja pengembangan perangkat lunak dan meningkatkan produktivitas bagi pengembang dan non-pengembang. Terlepas dari keahlian teknis Anda, ini membantu Anda membuat arsitektur dan solusi desain untuk masalah bisnis, bootstrap lingkungan kerja Anda, membantu Anda menerapkan fitur baru, dan menghasilkan kasus uji untuk validasi. Ini menggunakan instruksi bahasa alami dan kemampuan AI untuk memastikan kode yang konsisten dan berkualitas tinggi dan untuk mengurangi tantangan pengkodean terlepas dari keterampilan pemrograman Anda.

Keuntungan utama Amazon Q Developer adalah kemampuannya untuk membebaskan Anda dari tugas pengkodean berulang. Saat Anda menggunakan `@workspace` anotasi, Amazon Q Developer menyerap dan mengindeks semua file kode, konfigurasi, dan struktur proyek di lingkungan pengembangan terintegrasi (IDE) Anda, dan memberikan respons yang disesuaikan untuk membantu Anda fokus pada pemecahan masalah secara kreatif. Anda dapat mendedikasikan lebih banyak waktu untuk merancang solusi inovatif dan meningkatkan pengalaman pengguna. Jika Anda tidak teknis, Anda dapat menggunakan Amazon Q Developer untuk merampingkan alur kerja dan berkolaborasi secara lebih efektif dengan tim pengembangan. Fitur kode Amazon Q Developer Explain menawarkan instruksi dan ringkasan terperinci, sehingga Anda dapat menavigasi basis kode yang kompleks.

Selain itu, Amazon Q Developer menyediakan pendekatan agnostik bahasa yang membantu pengembang tingkat junior dan menengah memperluas keahlian mereka. Anda dapat berkonsentrasi pada konsep inti dan logika bisnis alih-alih sintaks khusus bahasa. Ini mengurangi kurva belajar saat Anda beralih teknologi.

Prasyarat dan batasan

Prasyarat

- IDE (misalnya, WebStorm atau Visual Studio Code) dengan plugin Amazon Q Developer diinstal. Untuk petunjuk, lihat Menginstal ekstensi atau plugin Amazon Q Developer di IDE Anda di dokumentasi Amazon Q Developer.
- Akun AWS Pengaturan aktif dengan Pengembang Amazon Q. Untuk petunjuknya, lihat Memulai dalam dokumentasi Pengembang Amazon Q.
- npm diinstal. Untuk petunjuk, lihat dokumentasi npm. Pola ini diuji dengan npm versi 10.8.
- AWS Command Line Interface (AWS CLI) diinstal. Untuk instruksi, lihat AWS CLI dokumentasi.

Batasan

- Amazon Q Developer hanya dapat melakukan satu tugas pengembangan pada satu waktu.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Alat

- Pola ini membutuhkan IDE seperti Visual Studio Code atau WebStorm. Untuk daftar yang didukung IDEs, lihat dokumentasi Pengembang Amazon Q.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

Praktik terbaik

Lihat Praktik pengkodean terbaik dengan Pengembang Amazon Q di Panduan AWS Preskriptif. Selain itu:

- Saat Anda memberikan petunjuk kepada Amazon Q Developer, pastikan instruksi Anda jelas dan tidak ambigu. Tambahkan cuplikan kode dan anotasi seperti `@workspace` ke prompt untuk memberikan lebih banyak konteks untuk permintaan Anda.
- Sertakan pustaka yang relevan dan impor untuk menghindari konflik atau tebakan yang salah oleh sistem.
- Jika kode yang dihasilkan tidak akurat atau seperti yang diharapkan, gunakan opsi Berikan umpan balik & regenerasi. Coba pecahkan petunjuk menjadi instruksi yang lebih kecil.

Epik

Mengatur lingkungan kerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat proyek baru.	Untuk membuat proyek baru di lingkungan kerja Anda, jalankan perintah berikut dan terima pengaturan default untuk semua pertanyaan: `npx create-next-app@latest`	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak
Uji aplikasi dasar.	Jalankan perintah berikut dan konfirmasikan bahwa aplikasi dasar berhasil dimuat di browser: `npm run dev`	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak
Bersihkan kode dasar.	Arahkan ke `page.tsx` file di `src/app` folder dan hapus konten default untuk mendapatkan halaman kosong. Setelah penghapusan, file akan terlihat seperti ini: `export default function Home() { return (<div></div>); }`	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Gunakan Amazon Q Developer untuk merancang proyek tic-tac-toe game

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan ikhtisar langkah-langkah.	1. Di IDE Anda, buka proyek Anda, dan pilih ikon Amazon Q untuk membuka panel obrolan. 2. Di panel obrolan Amazon Q Developer, mintalah ikhtisar untuk membuat aplikasi satu halaman (SPA). Contoh: `I would like to create a single-page application involving Next.js React framework for tic-tac-toe game. What are the steps?`	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Hasilkan kode untuk tic-tac-toe.	Di panel obrolan, mulailah tugas pengembangan dengan menggunakan `/dev` perintah diikuti dengan deskripsi tugas. Contoh:
	`/dev Create a React-based single-page application written in TypeScript for a tic-tac-toe game with the following specifications:`
	`1. Design an aesthetically pleasing interface with the game grid`
		Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`centered vertically and horizontally on the page.`
	`2. Include a heading and clear instructions on how to play the game.`
	`3. Implement color-coding for X and O marks to distinguish them easily.`
	
	Pengembang Amazon Q menghasilkan kode berdasarkan instruksi Anda.	

Periksa dan terima kode yang dihasilkan.	Periksa kode secara visual, dan pilih Terima kode untuk mengganti `page.tsx` file secara otomatis. Jika Anda mengalami masalah, pilih Berikan umpan balik & buat ulang dan jelaskan masalah yang Anda temui.	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbaiki kesalahan serat.	Contoh tic-tac-toe permainan termasuk grid. Kode yang dihasilkan Amazon Q Developer mungkin menggunakan tipe default`any`. Anda dapat menambahk an keamanan tipe dengan meminta Pengembang Amazon Q sebagai berikut: `/dev Ensure proper TypeScript typing for the onSquare Click event handler to resolve any 'any' type issues.`	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan daya tarik visual.	Anda dapat memecah persyaratan asli menjadi fragmen yang lebih kecil. Misalnya, Anda dapat meningkatkan UI game dengan petunjuk berikut dalam tugas dev. Prompt ini meningkatkan gaya Cascading Style Sheets (CSS) dan mengekspor aplikasi untuk penerapan.
	`/dev Debug and fix any CSS issues to correctly display the game grid and overall layout.`
	
	`Simplify the code by removing game history functionality and related components.`
	
	`Implement static file export to an 'out' directory for easy deployment. The solution should be fully functional, visually appealing, and free of typing errors or layout issues.`
		Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji lagi.	1. Sekarang setelah Anda menyelesaikan siklus hidup pengembangan, uji kode untuk mengonfir masi bahwa kode berfungsi seperti yang diharapkan. Untuk menjalankan aplikasi secara lokal, gunakan perintah:
	`npm run dev`
	2. Jika aplikasi berfungsi seperti yang diharapkan, gunakan `build` perintah untuk mengekspor seluruh aplikasi ke folder output sebagai persiapan untuk penyebaran:
	`npm run build`
		Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Menyebarkan aplikasi ke AWS Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat folder dan file untuk penyebaran.	Dalam proyek di lingkunga n kerja Anda, buat folder penyebaran dan dua file di dalamnya: `pushtos3.sh` dan`cloudformation.yml` :	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`mkdir deployment && cd deployment`
	`touch pushtos3.sh && chmod +x pushtos3.sh`
	`touch cloudformation.yml`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hasilkan kode otomatisasi.	1. Di panel obrolan untuk Pengembang Amazon Q, berikan prompt berikut:
	
	`/dev Generate a Cloudformation template that creates two resources: tictactoe artifact bucket and CloudFront. CloudFront should be configured with this bucket as origin.`
	`Add cache policy appropriate for Amazon S3 and default root object as index.html.`
	`Ensure that origin access control is used and no public bucket is created.`
	`Output all the resources and their ARNs.`
	
	2. Tinjau dan terima kode yang dihasilkan. `cloudformation.yml` File yang Anda buat sebelumnya sekarang harus diisi dengan CloudFormation skrip yang membuat sumber daya untuk file. AWS Cloud
		Administrator AWS, AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan konten skrip.	Untuk membuat skrip penyebaran, gunakan prompt berikut: `/dev Modify the pushtos3 shell script so that it can use AWS CLI commands to create a CloudFormation stack named tictactoe-stack if it does not exist already, and use cloudformation.yml as the source template. Wait for the stack to complete and sync the contents from the out folder to the S3 bucket. Perform invalidation of the CloudFront origin.`	Pengembang aplikasi, Pemrogram, Pengembang perangkat lunak
Menyebarkan aplikasi ke. AWS Cloud	1. Konfigurasikan lingkungan kerja dengan AWS kredensil yang valid. 2. Jalankan skrip shell untuk menyebarkan tic-tac-toe game yang berfungsi penuh ke file. AWS Cloud	Administrator AWS, AWS DevOps, arsitek Cloud, Pengembang aplikasi

Pemecahan masalah

Isu	Solusi

Build tidak membuat aplikasi satu halaman atau mengekspornya ke folder keluaran.	Lihatlah isi `next.config.mjs` file.
	
	Jika kode memiliki konfigurasi default berikut:
	
	`const nextConfig = {};`
	
	memodifikasinya sebagai berikut:
	
	`const nextConfig = {`
	`  output: 'export',`
	`  distDir: 'out',`
	`};`

Sumber daya terkait

- Membuat proyek React baru (dokumentasi React)
- Ikhtisar Pengembang Amazon Q (AWS dokumentasi)
- Praktik terbaik Pengembang Amazon Q (Panduan AWS Preskriptif)
- Menginstal, mengonfigurasi, dan menggunakan Amazon Q Developer dengan JetBrains IDEs (YouTube video)
- Menginstal Amazon Q untuk baris perintah (AWS dokumentasi)

Gunakan SageMaker Processing untuk rekayasa fitur terdistribusi dari kumpulan data ML skala terabyte

Chris Boomhower, Amazon Web Services

Ringkasan

Banyak kumpulan data skala terabyte atau lebih besar sering terdiri dari struktur folder hierarkis, dan file dalam kumpulan data terkadang berbagi saling ketergantungan. Untuk alasan ini, insinyur pembelajaran mesin (ML) dan ilmuwan data harus membuat keputusan desain yang bijaksana untuk menyiapkan data tersebut untuk pelatihan model dan inferensi. Pola ini menunjukkan bagaimana Anda dapat menggunakan teknik macrosharding dan microsharding manual dalam kombinasi dengan Amazon SageMaker Processing dan paralelisasi CPU virtual (vCPU) untuk menskalakan proses rekayasa fitur secara efisien untuk kumpulan data BIG data MLyang rumit.

Pola ini mendefinisikan macrosharding sebagai pemisahan direktori data di beberapa mesin untuk diproses, dan microsharding sebagai pemisahan data pada setiap mesin di beberapa thread pemrosesan. Pola menunjukkan teknik ini dengan menggunakan Amazon SageMaker dengan contoh catatan bentuk gelombang deret waktu dari kumpulan data MIMIC-III. PhysioNet Dengan menerapkan teknik dalam pola ini, Anda dapat meminimalkan waktu pemrosesan dan biaya untuk rekayasa fitur sambil memaksimalkan pemanfaatan sumber daya dan efisiensi throughput. Pengoptimalan ini bergantung pada SageMaker Pemrosesan terdistribusi pada instans Amazon Elastic Compute Cloud (Amazon EC2) dan v CPUs untuk kumpulan data besar yang serupa, terlepas dari tipe datanya.

Prasyarat dan batasan

Prasyarat

- Akses ke instance SageMaker notebook atau SageMaker Studio, jika Anda ingin menerapkan pola ini untuk kumpulan data Anda sendiri. Jika Anda menggunakan Amazon SageMaker untuk pertama kalinya, lihat Memulai Amazon SageMaker di dokumentasi AWS.
- SageMaker Studio, jika Anda ingin menerapkan pola ini dengan data sampel PhysioNet MIMIC-III.
- Pola menggunakan SageMaker Processing, tetapi tidak memerlukan pengalaman menjalankan pekerjaan SageMaker Processing.

Batasan

- Pola ini sangat cocok untuk kumpulan data ML yang menyertakan file yang saling bergantung. Interdependensi ini paling diuntungkan dari macrosharding manual dan menjalankan beberapa pekerjaan Pemrosesan instance SageMaker tunggal secara paralel. Untuk kumpulan data di mana saling ketergantungan seperti itu tidak ada, `ShardedByS3Key` fitur dalam SageMaker Processing mungkin merupakan alternatif yang lebih baik untuk macrosharding, karena mengirimkan data sharded ke beberapa instance yang dikelola oleh pekerjaan Pemrosesan yang sama. Namun, Anda dapat menerapkan strategi microsharding pola ini di kedua skenario untuk memanfaatkan instance v. CPUs

Versi produk

- Amazon SageMaker Python SDK versi 2

Arsitektur

Tumpukan teknologi target

- Amazon Simple Storage Service (Amazon S3)
- Amazon SageMaker

Arsitektur target

Macrosharding dan instance terdistribusi EC2

10 proses paralel yang diwakili dalam arsitektur ini mencerminkan struktur dataset MIMIC-III. (Proses diwakili oleh elips untuk penyederhanaan diagram.) Arsitektur serupa berlaku untuk kumpulan data apa pun saat Anda menggunakan macrosharding manual. Dalam kasus MIMIC-III, Anda dapat menggunakan struktur mentah kumpulan data untuk keuntungan Anda dengan memproses setiap folder grup pasien secara terpisah, dengan sedikit usaha. Dalam diagram berikut, blok grup rekaman muncul di sebelah kiri (1). Mengingat sifat data yang terdistribusi, masuk akal untuk membelah oleh kelompok pasien.

Namun, sharding secara manual oleh kelompok pasien berarti bahwa pekerjaan Pemrosesan terpisah diperlukan untuk setiap folder grup pasien, seperti yang Anda lihat di bagian tengah diagram (2), alih-alih satu pekerjaan Pemrosesan dengan beberapa EC2 instance. Karena data MIMIC-III mencakup file bentuk gelombang biner dan file header berbasis teks yang cocok,

dan ada ketergantungan yang diperlukan pada pustaka wfdb untuk ekstraksi data biner, semua catatan untuk pasien tertentu harus tersedia pada contoh yang sama. Satu-satunya cara untuk memastikan bahwa setiap file header terkait bentuk gelombang biner juga ada adalah dengan menerapkan sharding manual untuk menjalankan setiap pecahan dalam pekerjaan Processing-nya sendiri, dan untuk menentukan `s3_data_distribution_type='FullyReplicated'` kapan Anda menentukan input pekerjaan Pemrosesan. Atau, jika semua data tersedia dalam satu direktori dan tidak ada dependensi di antara file, opsi yang lebih cocok mungkin meluncurkan pekerjaan Pemrosesan tunggal dengan beberapa EC2 instance dan ditentukan. `s3_data_distribution_type='ShardedByS3Key'` Menentukan `ShardedByS3Key` sebagai tipe distribusi data Amazon S3 SageMaker mengarahkan untuk mengelola pembagian data secara otomatis di seluruh instance.

Meluncurkan pekerjaan Pemrosesan untuk setiap folder adalah cara hemat biaya untuk memproses data sebelumnya, karena menjalankan beberapa instance secara bersamaan menghemat waktu. Untuk penghematan biaya dan waktu tambahan, Anda dapat menggunakan microsharding dalam setiap pekerjaan Pemrosesan.

Microsharding dan parallel v CPUs

Dalam setiap pekerjaan Pemrosesan, data yang dikelompokkan dibagi lagi untuk memaksimalkan penggunaan semua v yang tersedia CPUs pada EC2 instance yang dikelola SageMaker sepenuhnya. Blok di bagian tengah diagram (2) menggambarkan apa yang terjadi dalam setiap pekerjaan Pemrosesan utama. Isi folder catatan pasien diratakan dan dibagi rata berdasarkan jumlah v yang tersedia CPUs pada instance. Setelah isi folder dibagi, kumpulan file berukuran merata didistribusikan di semua v CPUs untuk diproses. Saat pemrosesan selesai, hasil dari setiap vCPU digabungkan menjadi satu file data untuk setiap pekerjaan Pemrosesan.

Dalam kode terlampir, konsep-konsep ini diwakili di bagian `src/feature-engineering-pass1/preprocessing.py` file berikut.

```
def chunks(lst, n):
    """
    Yield successive n-sized chunks from lst.

    :param lst: list of elements to be divided
    :param n: number of elements per chunk
    :type lst: list
    :type n: int
    :return: generator comprising evenly sized chunks
    :rtype: class 'generator'
```

```
    """
    for i in range(0, len(lst), n):
        yield lst[i:i + n]


# Generate list of data files on machine
data_dir = input_dir
d_subs = next(os.walk(os.path.join(data_dir, '.')))[1]
file_list = []
for ds in d_subs:
    file_list.extend(os.listdir(os.path.join(data_dir, ds, '.')))
dat_list = [os.path.join(re.split('_|\.', f)[0].replace('n', ''), f[:-4]) for f in
 file_list if f[-4:] == '.dat']

# Split list of files into sub-lists
cpu_count = multiprocessing.cpu_count()
splits = int(len(dat_list) / cpu_count)
if splits == 0: splits = 1
dat_chunks = list(chunks(dat_list, splits))

# Parallelize processing of sub-lists across CPUs
ws_df_list = Parallel(n_jobs=-1, verbose=0)(delayed(run_process)(dc) for dc in
 dat_chunks)

# Compile and pickle patient group dataframe
ws_df_group = pd.concat(ws_df_list)
ws_df_group = ws_df_group.reset_index().rename(columns={'index': 'signal'})
ws_df_group.to_json(os.path.join(output_dir, group_data_out))
```

Sebuah fungsi`chunks`, pertama kali didefinisikan untuk mengkonsumsi daftar yang diberikan dengan membaginya menjadi potongan-potongan panjang yang berukuran sama `n` dan dengan mengembalikan hasil ini sebagai generator. Selanjutnya, data diratakan di seluruh folder pasien dengan menyusun daftar semua file bentuk gelombang biner yang ada. Setelah ini selesai, jumlah `v` CPUs yang tersedia pada EC2 instance diperoleh. Daftar file bentuk gelombang biner dibagi secara merata di `v` ini CPUs dengan memanggil`chunks`, dan kemudian setiap sublis bentuk gelombang diproses pada vCPU-nya sendiri dengan menggunakan kelas Paralel joblib. Hasil secara otomatis digabungkan ke dalam satu daftar kerangka data oleh pekerjaan Pemrosesan, yang SageMaker kemudian diproses lebih lanjut sebelum menuliskannya ke Amazon S3 setelah pekerjaan selesai. Dalam contoh ini, ada 10 file yang ditulis ke Amazon S3 oleh pekerjaan Processing (satu untuk setiap pekerjaan).

Ketika semua pekerjaan Pemrosesan awal selesai, pekerjaan Pemrosesan sekunder, yang ditampilkan di blok di sebelah kanan diagram (3) menggabungkan file output yang dihasilkan oleh setiap pekerjaan Pemrosesan utama dan menulis output gabungan ke Amazon S3 (4).

Alat

Alat

- Python - Contoh kode yang digunakan untuk pola ini adalah Python (versi 3).
- SageMaker Studio — Amazon SageMaker Studio adalah lingkungan pengembangan terintegrasi (IDE) berbasis web untuk pembelajaran mesin yang memungkinkan Anda membuat, melatih, men-debug, menerapkan, dan memantau model pembelajaran mesin Anda. Anda menjalankan pekerjaan SageMaker Pemrosesan dengan menggunakan notebook Jupyter di dalam Studio. SageMaker
- SageMaker Pemrosesan - Amazon SageMaker Processing menyediakan cara yang disederhanakan untuk menjalankan beban kerja pemrosesan data Anda. Dalam pola ini, kode rekayasa fitur diimplementasikan dalam skala besar dengan menggunakan pekerjaan SageMaker Pemrosesan.

Kode

File zip terlampir menyediakan kode lengkap untuk pola ini. Bagian berikut menjelaskan langkah-langkah untuk membangun arsitektur untuk pola ini. Setiap langkah diilustrasikan dengan kode sampel dari lampiran.

Epik

Siapkan lingkungan SageMaker Studio Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses Amazon SageMaker Studio.	Onboard ke SageMaker Studio di akun AWS Anda dengan mengikuti petunjuk yang disediakan dalam SageMaker dokumentasi Amazon.	Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal utilitas wget.	Instal wget jika Anda onboard dengan konfigura si SageMaker Studio baru atau jika Anda belum pernah menggunakan utilitas ini di SageMaker Studio sebelumny a. Untuk menginstal, buka jendela terminal di konsol SageMaker Studio dan jalankan perintah berikut: `sudo yum install wget`	Ilmuwan data, insinyur ML

Unduh dan unzip kode sampel.	Unduh `attachments.zip` file di bagian Lampiran. Di jendela terminal, arahkan ke folder tempat Anda mengunduh file dan ekstrak isinya:
	
	`unzip attachment.zip`
	
	Arahkan ke folder tempat Anda mengekstrak file.zip, dan ekstrak konten file. `Scaled-Processing.zip`
	
	`unzip Scaled-Pr ocessing.zip`
		Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kumpulan data sampel dari physionet.org dan unggah ke Amazon S3.	Jalankan `get_data.ipynb` notebook Jupyter di dalam folder yang berisi file. `Scaled-Processing` Notebook ini mengunduh contoh kumpulan data MIMIC-III dari physionet.org dan mengunggahnya ke bucket sesi Studio Anda di Amazon S3. SageMaker	Ilmuwan data, insinyur ML

Konfigurasikan skrip preprocessing pertama

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ratakan hierarki file di semua subdirektori.	Dalam kumpulan data besar seperti MIMIC-III, file sering didistribusikan di beberapa subdirektori bahkan dalam kelompok induk logis. Skrip Anda harus dikonfigurasi untuk meratakan semua file grup di semua subdirektori, seperti yang ditunjukkan oleh kode berikut.
	`# Generate list of .dat files on machine`
	`data_dir = input_dir`
	`d_subs = next(os.walk(os.path.join(data_dir, '.')))[1]`
	`file_list = []`
	`for ds in d_subs:`
	`  file_list.extend(os.listdir(os.path.`
		Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`join(data_dir, ds, '.')))`
	`dat_list = [os.path.join(re.split('_|\.', f)[0].replace('n', ''), f[:-4]) for f in file_list if f[-4:] == '.dat']`
	
	Note
	Contoh cuplikan kode dalam epik ini berasal dari `src/feature-engineering-pass1/preprocessing.py` file, yang disediakan dalam lampiran.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bagilah file menjadi subkelompok berdasarkan jumlah vCPU.	File harus dibagi menjadi subkelompok berukuran merata, atau potongan, tergantung pada jumlah v yang CPUs ada pada instance yang menjalankan skrip. Untuk langkah ini, Anda dapat menerapkan kode yang mirip dengan berikut ini.
	`# Split list of files into sub-lists`
	`cpu_count = multiprocessing.cpu_count()`
	`splits = int(len(dat_list) / cpu_count)`
	`if splits == 0: splits = 1`
	`dat_chunks = list(chunks(dat_list, splits))`
		Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Paralelisasi pemrosesan subkelompok di seluruh v. CPUs	Logika skrip harus dikonfigu rasi untuk memproses semua subkelompok secara paralel. Untuk melakukan ini, gunakan `Parallel` kelas dan `delayed` metode perpustak aan Joblib sebagai berikut.
	`# Parallelize processin g of sub-lists across CPUs`
	`ws_df_list = Parallel( n_jobs=-1, verbose=0) (delayed(run_process) (dc) for dc in dat_chunk s)`
		Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Simpan output grup file tunggal ke Amazon S3.	Ketika pemrosesan vCPU paralel selesai, hasil dari setiap vCPU harus digabungkan dan diunggah ke jalur bucket S3 grup file. Untuk langkah ini, Anda dapat menggunakan kode yang mirip dengan berikut ini.
	`# Compile and pickle patient group dataframe`
	`ws_df_group = pd.concat(ws_df_list)`
	`ws_df_group = ws_df_group.reset_index().rename(columns={'index': 'signal'})`
	`ws_df_group.to_json(os.path.join(output_dir, group_data_out))`
		Ilmuwan data, insinyur ML

Konfigurasikan skrip preprocessing kedua

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gabungkan file data yang dihasilkan di semua pekerjaan Pemrosesan yang menjalankan skrip pertama.	Script sebelumnya mengeluarkan satu file untuk setiap pekerjaan SageMaker Processing yang memproses sekelompok file dari dataset. Selanjutnya, Anda perlu menggabungkan file output ini menjadi satu objek dan menulis satu set data output	Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	ke Amazon S3. Ini ditunjukkan dalam `src/feature-engineering-pass1p5/preprocessing.py` file, yang disediakan dalam lampiran, sebagai berikut.
	
	`def write_parquet(wavs_df, path):`
	`"""`
	`Write waveform summary dataframe to S3 in parquet format.`
	
	`:param wavs_df: waveform summary dataframe`
	`:param path: S3 directory prefix`
	`:type wavs_df: pandas dataframe`
	`:type path: str`
	`:return: None`
	`"""`
	`extra_args = {"ServerSideEncryption": "aws:kms"}`
	`wr.s3.to_parquet(`
	`df=wavs_df,`
	`path=path,`
	`compression='snappy',`
	`s3_additional_kwargs=extra_args)`
	
	`def combine_data():`
	`"""`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> Get combined data
	 and write to parquet.
	
	    :return: waveform
	 summary dataframe
	    :rtype: pandas
	 dataframe
	    """
	    wavs_df = get_data(
	)
	    wavs_df = normalize
	_signal_names(wavs
	_df)
	    write_parquet(wavs
	_df, "s3://{}/{}/
	{}".format(buck
	et_xform, dataset_p
	refix, pass1p5ou
	t_data))
	
	    return wavs_df
	
	wavs_df = combine_d
	ata()</pre>	

Jalankan pekerjaan Processing

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan pekerjaan Processing pertama.	Untuk melakukan macrosharding, jalankan pekerjaan Pemrosesan terpisah untuk setiap grup file. Microsharding dilakukan di dalam setiap pekerjaan Pemrosesan, karena setiap pekerjaan menjalankan skrip pertama	Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Anda. Kode berikut menunjukkan cara meluncurkan pekerjaan Processing untuk setiap direktori grup file dalam cuplikan berikut (termasuk dalam). `notebooks/FeatExtract_Pass1.ipynb`
	
	`pat_groups = list(range(30,40))`
	`ts = str(int(time.time()))`
	
	`for group in pat_groups:`
	`  sklearn_processor = SKLearnProcessor(framework_version='0.20.0',`
	
	`  role=role,`
	
	`  instance_type='ml.m5.4xlarge',`
	
	`  instance_count=1,`
	
	`  volume_size_in_gb=5)`
	`  sklearn_processor.run(`
	`  code='../src/feature-engineering-pass1/preprocessing.py',`
	`  job_name='-'.join(['scaled-`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`processing-p1', str(group), ts]), arguments=["input_path", "/opt/ml/processing/input", "output_path", "/opt/ml/processing/output", "group_data_out", "ws_df_group.json"], inputs= [ProcessingInput(source=f's3://{sess.default_bucket()}/data_inputs/{group}', destination='/opt/ml/processing/input', s3_data_distribution_type='FullyReplicated')], outputs= [ProcessingOutput(source='/opt/ml/processing/output', destination=f's3://{sess.default_bucket()}/data_outputs/{group}'`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>)
	        ],
	        wait=False
	    )</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan pekerjaan Pemrosesan kedua.	Untuk menggabungkan output yang dihasilkan oleh set pertama pekerjaan pemrosesan dan melakukan perhitungan tambahan untuk preprocessing, Anda menjalankan skrip kedua Anda dengan menggunakan satu pekerjaan Processing. SageMaker Kode berikut menunjukkan ini (termasuk dalam `notebooks/FeatExtract_Pass1p5.ipynb`).
	`ts = str(int(time.time()))`
	`bucket = sess.default_bucket()`
	
	`sklearn_processor = SKLearnProcessor(framework_version='0.20.0',`
	
	`role=role,`
	
	`instance_type='ml.t3.2xlarge',`
	
	`instance_count=1,`
	
	`volume_size_in_gb=5)`
	`sklearn_processor.run(`
	`code='../src/feature-engineering-pass1p5/preprocessing.py',`
		Ilmuwan data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>job_name='-'.join(
	['scaled-processing',
	'p1p5', ts]),
	arguments=['bucket
	', bucket,
	'pass1out
	_prefix', 'data_out
	puts',
	'pass1out
	_data', 'ws_df_gr
	oup.json',
	'pass1p5o
	ut_data', 'waveform
	_summary.parquet',
	'statsdat
	a_name', 'signal_s
	tats.csv'],
	wait=True
	)</pre>	

Sumber daya terkait

- Onboard ke Amazon SageMaker Studio Menggunakan Mulai Cepat (SageMaker dokumentasi)
- Data Proses (SageMaker dokumentasi)
- Pemrosesan Data dengan scikit-learn (dokumentasi) SageMaker
- Dokumentasi joblib.parallel
- Moody, B., Moody, G., Villarroel, M., Clifford, GD, & Silva, I. (2020). Database Bentuk Gelombang MIMIC-III (versi 1.0). PhysioNet.
- Johnson, A.E. W., Pollard, TJ, Shen, L., Lehman, LH, Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, LA, & Mark, RG (2016). MIMIC-III, database perawatan kritis yang dapat diakses secara bebas. Data Ilmiah, 3, 160035.
- Lisensi Database Bentuk Gelombang MIMIC-III

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Visualisasikan hasil AI/ML model menggunakan Flask dan AWS Elastic Beanstalk

Chris Caudill dan Durga Sury, Amazon Web Services

Ringkasan

Memvisualisasikan output dari layanan kecerdasan buatan dan pembelajaran mesin (AI/ML) seringkali membutuhkan panggilan API kompleks yang harus disesuaikan oleh pengembang dan teknisi Anda. Ini bisa menjadi kelemahan jika analis Anda ingin menjelajahi kumpulan data baru dengan cepat.

Anda dapat meningkatkan aksesibilitas layanan Anda dan memberikan bentuk analisis data yang lebih interaktif dengan menggunakan antarmuka pengguna berbasis web (UI) yang memungkinkan pengguna untuk mengunggah data mereka sendiri dan memvisualisasikan hasil model di dasbor.

Pola ini menggunakan Flask dan Plotly untuk mengintegrasikan Amazon Comprehend dengan aplikasi web khusus dan memvisualisasikan sentimen dan entitas dari data yang disediakan pengguna. Pola ini juga menyediakan langkah-langkah untuk menerapkan aplikasi dengan menggunakan AWS Elastic Beanstalk. Anda dapat menyesuaikan aplikasi dengan menggunakan layanan AI Amazon Web Services (AWS) atau dengan model terlatih khusus yang dihosting di titik akhir (misalnya, SageMaker titik akhir Amazon).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi pada mesin lokal Anda. Untuk informasi selengkapnya tentang ini, lihat Dasar-dasar konfigurasi dalam dokumentasi AWS CLI. Anda juga dapat menggunakan lingkungan pengembangan terintegrasi AWS Cloud9 (IDE); untuk informasi selengkapnya tentang ini, lihat tutorial Python untuk AWS Cloud9 dan Mempratinjau aplikasi yang sedang berjalan di AWS Cloud9 IDE dalam dokumentasi AWS Cloud9.

  Pemberitahuan: AWS Cloud9 tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS Cloud9 dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya
- Pemahaman tentang kerangka aplikasi web Flask. Untuk informasi selengkapnya tentang Flask, lihat Quickstart dalam dokumentasi Flask.

- Python versi 3.6 atau yang lebih baru, diinstal dan dikonfigurasi. Anda dapat menginstal Python dengan mengikuti petunjuk dari Menyiapkan lingkungan pengembangan Python Anda di dokumentasi AWS Elastic Beanstalk.
- Elastic Beanstalk Command Line Interface (EB CLI), diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang ini, lihat Menginstal CLI EB dan Mengonfigurasi CLI EB dari dokumentasi AWS Elastic Beanstalk.

Batasan

- Aplikasi Flask pola ini dirancang untuk bekerja dengan file.csv yang menggunakan kolom teks tunggal dan dibatasi hingga 200 baris. Kode aplikasi dapat disesuaikan untuk menangani jenis file dan volume data lainnya.
- Aplikasi tidak mempertimbangkan retensi data dan terus mengumpulkan file pengguna yang diunggah hingga dihapus secara manual. Anda dapat mengintegrasikan aplikasi dengan Amazon Simple Storage Service (Amazon S3) Simple Storage S3) untuk penyimpanan objek persisten atau menggunakan database seperti Amazon DynamoDB untuk penyimpanan nilai kunci tanpa server.
- Aplikasi hanya mempertimbangkan dokumen dalam bahasa Inggris. Namun, Anda dapat menggunakan Amazon Comprehend untuk mendeteksi bahasa utama dokumen. Untuk informasi selengkapnya tentang bahasa yang didukung untuk setiap tindakan, lihat referensi API di dokumentasi Amazon Comprehend.
- Daftar pemecahan masalah yang berisi kesalahan umum dan solusinya tersedia di bagian Informasi tambahan.

Arsitektur

Arsitektur aplikasi labu

Flask adalah kerangka kerja ringan untuk mengembangkan aplikasi web dengan Python. Ini dirancang untuk menggabungkan pemrosesan data Python yang kuat dengan UI web yang kaya. Aplikasi Flask pola menunjukkan cara membuat aplikasi web yang memungkinkan pengguna mengunggah data, mengirim data ke Amazon Comprehend untuk inferensi, dan kemudian memvisualisasikan hasilnya. Aplikasi ini memiliki struktur sebagai berikut:

- `static`— Berisi semua file statis yang mendukung UI web (misalnya JavaScript, CSS, dan gambar)
- `templates`- Berisi semua halaman HTML aplikasi

- `userData`— Menyimpan data pengguna yang diunggah
- `application.py`— File aplikasi Flask
- `comprehend_helper.py`— Fungsi untuk melakukan panggilan API ke Amazon Comprehend
- `config.py`— File konfigurasi aplikasi
- `requirements.txt`- Dependensi Python yang dibutuhkan oleh aplikasi

`application.py`Script berisi fungsionalitas inti aplikasi web, yang terdiri dari empat rute Flask. Diagram berikut menunjukkan rute Flask ini.

- `/`adalah root aplikasi dan mengarahkan pengguna ke `upload.html` halaman (disimpan dalam `templates` direktori).
- `/saveFile`adalah rute yang dipanggil setelah pengguna mengunggah file. Rute ini menerima `POST` permintaan melalui formulir HTML, yang berisi file yang diunggah oleh pengguna. File disimpan di `userData` direktori dan rute mengarahkan pengguna ke rute. `/dashboard`
- `/dashboard`mengirim pengguna ke `dashboard.html` halaman. Dalam HTML halaman ini, ia menjalankan JavaScript kode `static/js/core.js` yang membaca data dari `/data` rute dan kemudian membangun visualisasi untuk halaman tersebut.
- `/data`adalah JSON API yang menyajikan data untuk divisualisasikan di dasbor. Rute ini membaca data yang disediakan pengguna dan menggunakan fungsi `comprehend_helper.py` untuk mengirim data pengguna ke Amazon Comprehend untuk analisis sentimen dan pengenalan entitas bernama (NER). Respons Amazon Comprehend diformat dan dikembalikan sebagai objek JSON.

Arsitektur penyebaran

Pertimbangan desain

Untuk informasi selengkapnya tentang pertimbangan desain untuk aplikasi yang digunakan menggunakan Elastic Beanstalk di AWS Cloud, lihat di dokumentasi AWS Elastic Beanstalk.

Tumpukan teknologi

- Amazon Comprehend
- Elastic Beanstalk

- Flask

Otomatisasi dan skala

Penerapan Elastic Beanstalk secara otomatis diatur dengan penyeimbang beban dan grup penskalaan otomatis. Untuk opsi konfigurasi lainnya, lihat Mengonfigurasi lingkungan Elastic Beanstalk dalam dokumentasi AWS Elastic Beanstalk.

Alat

- AWS Command Line Interface (AWS CLI) adalah alat terpadu yang menyediakan antarmuka yang konsisten untuk berinteraksi dengan semua bagian AWS.
- Amazon Comprehend menggunakan Natural Language Processing (NLP) untuk mengekstrak wawasan tentang isi dokumen tanpa memerlukan preprocessing khusus.
- AWS Elastic Beanstalk membantu Anda menerapkan dan mengelola aplikasi dengan cepat di AWS Cloud tanpa harus mempelajari infrastruktur yang menjalankan aplikasi tersebut.
- Elastic Beanstalk CLI (EB CLI) adalah antarmuka baris perintah untuk AWS Elastic Beanstalk yang menyediakan perintah interaktif untuk menyederhanakan pembuatan, pembaruan, dan pemantauan lingkungan dari repositori lokal.
- Framework Flask melakukan pemrosesan data dan panggilan API menggunakan Python dan menawarkan visualisasi web interaktif dengan Plotly.

Kode

Kode untuk pola ini tersedia dalam hasil AI/ML model GitHub Visualize menggunakan repositori Flask dan AWS Elastic Beanstalk.

Epik

Siapkan aplikasi Flask

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Tarik kode aplikasi dari hasil AI/ML model GitHub Visualize menggunakan repositori Flask dan AWS Elastic Beanstalk	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan menjalankan perintah berikut: `git clone git@github.com:aws-samples/aws-comprehend-elasticbeanstalk-for-flask.git` **Note** Pastikan Anda mengonfigurasi kunci SSH Anda dengan GitHub.	

Instal modul Python.	Setelah Anda mengkloning repositori, `aws-comprehend-elasticbeanstalk-for-flask` direktori lokal baru dibuat. Dalam direktori itu, `requirements.txt` file berisi modul Python dan versi yang menjalankan aplikasi. Gunakan perintah berikut untuk menginstal modul:
	
	`cd aws-comprehend-elasticbeanstalk-for-flask`
	
	`pip install -r requirements.txt`
		Pengembang Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji aplikasi secara lokal.	Mulai server Flask dengan menjalankan perintah berikut:
	
	`python application.py`
	
	Ini mengembalikan informasi tentang server yang sedang berjalan. Anda harus dapat mengakses aplikasi dengan membuka browser dan mengunjungi http://localhost:5000
	
	Note
	Jika Anda menjalankan aplikasi di AWS Cloud9 IDE, Anda perlu mengganti `application.run()` perintah dalam file dengan baris `application.py` berikut:
	
	`application.run(host=os.getenv('IP', '0.0.0.0'),port=int(os.getenv('PORT', 8080)))`
	
	Anda harus mengembalikan perubahan ini sebelum penerapan.
		Pengembang Python

Menyebarkan aplikasi ke Elastic Beanstalk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan aplikasi Elastic Beanstalk.	Untuk meluncurkan proyek Anda sebagai aplikasi Elastic Beanstalk, jalankan perintah berikut dari direktori root aplikasi Anda: `eb init -p python-3.6 comprehend_flask --region us-east-1` ⚠ Important • `comprehend_flask` adalah nama aplikasi Elastic Beanstalk dan dapat diubah sesuai dengan kebutuhan Anda. • Anda dapat mengganti Wilayah AWS dengan Wilayah pilihan Anda. Wilayah default di AWS CLI digunakan jika Anda tidak menentukan Wilayah. • Aplikasi ini dibangun dengan Python versi 3.6. Anda mungkin mengalami kesalahan jika Anda menggunakan versi Python lainnya.	Arsitek, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jalankan `eb init -i` perintah untuk opsi konfigurasi penerapan lainnya.	
Menyebarkan lingkungan Elastic Beanstalk.	Jalankan perintah berikut dari direktori root aplikasi: `eb create comprehend-flask-env` **Note** `comprehend-flask-env` adalah nama lingkungan Elastic Beanstalk dan dapat diubah sesuai dengan kebutuhan Anda. Nama hanya dapat berisi huruf, angka, dan tanda hubung.	Arsitek, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Otorisasi penerapan Anda untuk menggunakan Amazon Comprehend.	Meskipun aplikasi Anda mungkin berhasil di-deploy, Anda juga harus menyediakan penerapan Anda dengan akses ke Amazon Comprehend. `ComprehendFullAccess` adalah kebijakan terkelola AWS yang menyediakan aplikasi yang diterapkan dengan izin untuk melakukan panggilan API ke Amazon Comprehend. Lampirkan `ComprehendFullAccess` kebijakan ke `aws-elasticbeanstalk-ec2-role` (peran ini dibuat secara otomatis untuk instance Amazon Elastic Compute Cloud EC2 (Amazon) penerapan Anda) dengan menjalankan perintah berikut: `aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/ComprehendFullAccess --role-name aws-elasticbeanstalk-ec2-role`	Pengembang, Arsitek keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important `aws-elasticbeanstalk-ec2-role` dibuat saat aplikasi Anda di-deploy. Anda harus menyelesaikan proses penerapan sebelum dapat melampirkan kebijakan AWS Identity and Access Management (IAM).	
Kunjungi aplikasi yang Anda gunakan.	Setelah aplikasi Anda berhasil menyebarkan, Anda dapat mengunjunginya dengan menjalankan `eb open` perintah. Anda juga dapat menjalankan `eb status` perintah untuk menerima detail tentang penerapan Anda. URL penyebaran tercantum di bawah`CNAME`.	Arsitek, Pengembang

(Opsional) Sesuaikan aplikasi ke model ML Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Otorisasi Elastic Beanstalk untuk mengakses model baru.	Pastikan Elastic Beanstalk memiliki izin akses yang diperlukan untuk titik akhir	Pengembang, Arsitek keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	model baru Anda. Misalnya, jika Anda menggunakan SageMaker titik akhir Amazon, penerapan Anda harus memiliki izin untuk memanggil titik akhir. Untuk informasi selengkapnya tentang ini, lihat InvokeEndpointdi SageMaker dokumentasi Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim data pengguna ke model baru.	Untuk mengubah model HTML yang mendasari dalam aplikasi ini, Anda harus mengubah file berikut: • `comprehend_helper.py` — Ini adalah skrip Python yang terhubung dengan Amazon Comprehend, memproses respons, dan mengembalikan hasil akhir ke aplikasi. Dalam skrip ini, Anda dapat merutekan data ke layanan AI lain di AWS Cloud atau Anda dapat mengirim data ke titik akhir model khusus. Kami menyarankan Anda juga memformat hasil dalam skrip ini untuk pemisahan logis dan kegunaan kembali pola ini. • `application.py` — Jika Anda mengubah nama `comprehend_helper.py` skrip atau fungsi, Anda perlu memperbarui `application.py` skrip aplikasi untuk mencerminkan perubahan tersebut.	Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui visualisasi dasbor.	Biasanya, menggabungkan model ML baru berarti visualisasi harus diperbarui untuk mencerminkan hasil baru. Perubahan ini dibuat dalam file-file berikut: • `templates/dashboard.html` — Aplikasi prebuilt hanya menyumbang dua visualisasi dasar. Seluruh tata letak halaman dapat disesuaikan dalam file ini. • `static/js/core.js` — Skrip ini menangkap output yang diformat dari `/data` rute server Flask dan menggunakan Plotly untuk membuat visualisasi. Anda dapat menambahkan atau memperbarui grafik halaman.	Pengembang web

(Opsional) Menyebarkan aplikasi yang diperbarui

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file persyaratan aplikasi Anda.	Sebelum mengirim perubahan ke Elastic Beanstalk, `requirements.txt` perbarui file untuk mencerminkan modul Python baru dengan menjalankan perintah	Pengembang Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	berikut di direktori root aplikasi Anda: `pip freeze > requirements.txt`	
Pasang kembali lingkungan Elastic Beanstalk.	Untuk memastikan bahwa perubahan aplikasi Anda tercermin dalam penyebaran Elastic Beanstalk Anda, navigasikan ke direktori root aplikasi Anda dan jalankan perintah berikut: `eb deploy` Ini mengirimkan versi terbaru dari kode aplikasi ke penyebaran Elastic Beanstalk Anda yang ada.	Administrator sistem, Arsitek

Sumber daya terkait

- Panggil titik akhir SageMaker model Amazon menggunakan Amazon API Gateway dan AWS Lambda
- Menyebarkan aplikasi Flask ke Elastic Beanstalk
- Referensi perintah EB CLI
- Menyiapkan lingkungan pengembangan Python Anda

Informasi tambahan

Daftar pemecahan masalah

Berikut ini adalah enam kesalahan umum dan solusinya.

Kesalahan 1

```
Unable to assume role "arn:aws:iam::xxxxxxxxxx:role/aws-elasticbeanstalk-ec2-role".
 Verify that the role exists and is configured correctly.
```

Solusi: Jika kesalahan ini terjadi saat Anda menjalankan`eb create`, buat contoh aplikasi di konsol Elastic Beanstalk untuk membuat profil instance default. Untuk informasi selengkapnya tentang ini, lihat Membuat lingkungan Elastic Beanstalk di dokumentasi AWS Elastic Beanstalk.

Kesalahan 2

```
Your WSGIPath refers to a file that does not exist.
```

Solusi: Kesalahan ini terjadi di log penerapan karena Elastic Beanstalk mengharapkan kode Flask diberi nama. `application.py` Jika Anda memilih nama yang berbeda, jalankan `eb config` dan edit WSGIPath seperti yang ditunjukkan pada contoh kode berikut:

```
aws:elasticbeanstalk:container:python:
     NumProcesses: '1'
     NumThreads: '15'
     StaticFiles: /static/=static/
     WSGIPath: application.py
```

Pastikan Anda mengganti `application.py` dengan nama file Anda.

Anda juga dapat memanfaatkan Gunicorn dan Procfile. Untuk informasi selengkapnya tentang pendekatan ini, lihat Mengonfigurasi server WSGI dengan Procfile dalam dokumentasi AWS Elastic Beanstalk.

Kesalahan 3

```
Target WSGI script '/opt/python/current/app/application.py' does not contain WSGI
 application 'application'.
```

Solusi: Elastic Beanstalk mengharapkan variabel yang mewakili aplikasi Flask Anda diberi nama. `application` Pastikan bahwa `application.py` file menggunakan `application` sebagai nama variabel:

```
application = Flask(__name__)
```

Kesalahan 4

```
The EB CLI cannot find your SSH key file for keyname
```

Solusi: Gunakan EB CLI untuk menentukan key pair mana yang akan digunakan atau untuk membuat key pair untuk instance deployment EC2 Anda. Untuk mengatasi kesalahan, jalankan `eb init -i` dan salah satu opsi akan bertanya:

```
Do you want to set up SSH for your instances?
```

Tanggapi dengan Y membuat key pair atau menentukan key pair yang ada.

Kesalahan 5

Saya telah memperbarui kode saya dan menerapkan kembali tetapi penerapan saya tidak mencerminkan perubahan saya.

Solusi: Jika Anda menggunakan repositori Git dengan penerapan Anda, pastikan bahwa Anda menambahkan dan mengkomit perubahan Anda sebelum menerapkan ulang.

Kesalahan 6

Anda melihat pratinjau aplikasi Flask dari AWS Cloud9 IDE dan mengalami kesalahan.

Solusi: Untuk informasi selengkapnya tentang hal ini, lihat Mempratinjau aplikasi yang sedang berjalan di AWS Cloud9 IDE dalam dokumentasi AWS Cloud9.

Pemrosesan bahasa alami menggunakan Amazon Comprehend

Dengan memilih untuk menggunakan Amazon Comprehend, Anda dapat mendeteksi entitas kustom dalam dokumen teks individual dengan menjalankan analisis real-time atau pekerjaan batch asinkron. Amazon Comprehend juga memungkinkan Anda untuk melatih pengenalan entitas kustom dan model klasifikasi teks yang dapat digunakan secara real time dengan membuat titik akhir.

Pola ini menggunakan pekerjaan batch asinkron untuk mendeteksi sentimen dan entitas dari file input yang berisi beberapa dokumen. Contoh aplikasi yang disediakan oleh pola ini dirancang bagi pengguna untuk mengunggah file.csv yang berisi satu kolom dengan satu dokumen teks per baris. `comprehend_helper.py`File dalam hasil AI/ML model GitHub Visualize menggunakan

repositori Flask dan AWS Elastic Beanstalk membaca file input dan mengirimkan input ke Amazon Comprehend untuk diproses.

BatchDetectEntities

Amazon Comprehend memeriksa teks kumpulan dokumen untuk entitas bernama dan mengembalikan entitas yang terdeteksi, lokasi, jenis entitas, dan skor yang menunjukkan tingkat kepercayaan Amazon Comprehend. Maksimal 25 dokumen dapat dikirim dalam satu panggilan API, dengan setiap dokumen berukuran lebih kecil dari 5.000 byte. Anda dapat memfilter hasil untuk hanya menampilkan entitas tertentu berdasarkan kasus penggunaan. Misalnya, Anda dapat melewati tipe `'quantity'` entitas dan menetapkan skor ambang batas untuk entitas yang terdeteksi (misalnya, 0,75). Kami menyarankan Anda menjelajahi hasil untuk kasus penggunaan spesifik Anda sebelum memilih nilai ambang batas. Untuk informasi selengkapnya tentang ini, lihat BatchDetectEntitiesdi dokumentasi Amazon Comprehend.

BatchDetectSentiment

Amazon Comprehend memeriksa sekumpulan dokumen yang masuk dan mengembalikan sentimen yang berlaku untuk setiap dokumen (,,, atau). `POSITIVE` `NEUTRAL` `MIXED` `NEGATIVE` Maksimal 25 dokumen dapat dikirim dalam satu panggilan API, dengan setiap dokumen berukuran lebih kecil dari 5.000 byte. Menganalisis sentimen sangat mudah dan Anda memilih sentimen dengan skor tertinggi untuk ditampilkan dalam hasil akhir. Untuk informasi selengkapnya tentang ini, lihat BatchDetectSentimentdi dokumentasi Amazon Comprehend.

Penanganan konfigurasi labu

Server Flask menggunakan serangkaian variabel konfigurasi untuk mengontrol bagaimana server berjalan. Variabel ini dapat berisi output debug, token sesi, atau pengaturan aplikasi lainnya. Anda juga dapat menentukan variabel kustom yang dapat diakses saat aplikasi sedang berjalan. Ada beberapa pendekatan untuk mengatur variabel konfigurasi.

Dalam pola ini, konfigurasi didefinisikan `config.py` dan diwariskan di dalam`application.py`.

- **Note**

  `config.py`berisi variabel konfigurasi yang diatur pada startup aplikasi. Dalam aplikasi ini, `DEBUG` variabel didefinisikan untuk memberitahu aplikasi untuk menjalankan server

dalam modus debug. : Mode debug tidak boleh digunakan saat menjalankan aplikasi di lingkungan produksi. `UPLOAD_FOLDER`adalah variabel kustom yang didefinisikan untuk direferensikan nanti dalam aplikasi dan menginformasikannya di mana data pengguna yang diunggah harus disimpan.

- `application.py`memulai aplikasi Flask dan mewarisi pengaturan konfigurasi yang didefinisikan dalam. `config.py` Ini dilakukan oleh kode berikut:

```
application = Flask(__name__)
application.config.from_pyfile('config.py')
```

Lebih banyak pola

- Percepat MLOps dengan Backstage dan template Amazon SageMaker AI swalayan
- Mengotomatiskan operasi AWS infrastruktur dengan menggunakan Amazon Bedrock
- Terapkan sistem agen di Amazon Bedrock dengan framework CreWai dengan menggunakan Terraform
- Terapkan ChatOps solusi untuk mengelola hasil pemindaian SAST dengan menggunakan Amazon Q Developer dalam tindakan kustom aplikasi obrolan dan CloudFormation
- Hasilkan wawasan data dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight
- Hasilkan wawasan z/OS data Db2 dengan menggunakan dan AWS Mainframe Modernization Amazon Q di Quick Sight
- Berikan instans SageMaker notebook akses sementara ke CodeCommit repositori di akun AWS lain
- Mengelola AWS Organizations kebijakan sebagai kode dengan menggunakan AWS CodePipeline dan Amazon Bedrock
- Modernisasi aplikasi CardDemo mainframe dengan menggunakan AWS Transform
- Memodernisasi dan menyebarkan aplikasi mainframe menggunakan dan Terraform AWS Transform
- Lakukan analitik lanjutan menggunakan Amazon Redshift ML
- Merampingkan manajemen EC2 kepatuhan Amazon dengan agen Amazon Bedrock dan AWS Config
- Merampingkan pengembangan dan penerapan bot Amazon Lex dengan menggunakan alur kerja otomatis
- Memecahkan masalah status AWS Step Functions dengan menggunakan Amazon Bedrock

Analitik

Topik

- Menganalisis data Amazon Redshift di Layanan Analisis Microsoft SQL Server
- Analisis dan visualisasikan data JSON bersarang dengan Amazon Athena dan Amazon Quick Sight
- Otomatiskan konsumsi data dari AWS Data Exchange Amazon S3
- Otomatiskan penegakan enkripsi di AWS Glue menggunakan templat AWS CloudFormation
- Membangun pipeline data untuk menyerap, mengubah, dan menganalisis data Google Analytics menggunakan AWS DataOps Development Kit
- Buat pipeline pemrosesan video dengan menggunakan Amazon Kinesis Video Streams dan AWS Fargate
- Buat pipeline layanan ETL untuk memuat data secara bertahap dari Amazon S3 ke Amazon Redshift menggunakan AWS Glue
- Membangun mesh data perusahaan dengan Amazon DataZone, AWS CDK, dan AWS CloudFormation
- Hitung value at risk (VaR) dengan menggunakan layanan AWS
- Konfigurasikan akses lintas akun ke Katalog Data AWS Glue bersama menggunakan Amazon Athena
- Mengonversi fitur temporal Teradata NORMALIZE ke Amazon Redshift SQL
- Mengonversi fitur Teradata RESET WHEN ke Amazon Redshift SQL
- Menerapkan dan mengelola data lake tanpa server di AWS Cloud dengan menggunakan infrastruktur sebagai kode
- Menegakkan penandaan kluster EMR Amazon saat diluncurkan
- Pastikan pencatatan EMR Amazon ke Amazon S3 diaktifkan saat peluncuran
- Hasilkan data pengujian menggunakan pekerjaan AWS Glue dan Python
- Mengkonsumsi data IoT secara hemat biaya langsung ke Amazon S3 menggunakan AWS IoT Greengrass
- Luncurkan pekerjaan Spark di cluster EMR sementara menggunakan fungsi Lambda
- Migrasikan beban kerja Apache Cassandra ke Amazon Keyspaces dengan menggunakan AWS Glue
- Migrasi data Hadoop ke Amazon S3 menggunakan Migrator WANdisco LiveData

- Migrasikan Oracle Business Intelligence 12c ke AWS Cloud dari server lokal
- Migrasi Tumpukan ELK ke Elastic Cloud di AWS
- Migrasi data ke AWS Cloud dengan menggunakan Starburst
- Optimalkan konsumsi ETL ukuran file input di AWS
- Mengatur pipeline ETL dengan validasi, transformasi, dan partisi menggunakan AWS Step Functions
- Lakukan analitik lanjutan menggunakan Amazon Redshift ML
- Kueri tabel Amazon DynamoDB dengan SQL dengan menggunakan Amazon Athena
- Akses, kueri, dan gabungkan tabel Amazon DynamoDB menggunakan Athena
- Siapkan ruang data minimum yang layak untuk berbagi data antar organisasi
- Siapkan penyortiran khusus bahasa untuk hasil kueri Amazon Redshift menggunakan UDF Python skalar
- Berlangganan fungsi Lambda ke notifikasi acara dari bucket S3 di Wilayah AWS yang berbeda
- Tiga jenis pekerjaan AWS Glue ETL untuk mengonversi data ke Apache Parquet
- Visualisasikan log audit Amazon Redshift menggunakan Amazon Athena dan Amazon QuickSight
- Visualisasikan laporan kredensi IAM untuk semua akun AWS menggunakan Amazon Quick Sight
- Lebih banyak pola

Menganalisis data Amazon Redshift di Layanan Analisis Microsoft SQL Server

Sunil Vora, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menghubungkan dan menganalisis data Amazon Redshift di Microsoft SQL Server Analysis Services, dengan menggunakan Penyedia Intellisoft OLE DB atau CData Penyedia ADO.NET untuk akses database.

Amazon Redshift adalah layanan gudang data dengan skala petabyte yang terkelola penuh di cloud. SQL Server Analysis Services adalah alat pemrosesan analitik online (OLAP) yang dapat Anda gunakan untuk menganalisis data dari data mart dan gudang data seperti Amazon Redshift. Anda dapat menggunakan SQL Server Analysis Services untuk membuat kubus OLAP dari data Anda untuk analisis data yang cepat dan canggih.

Prasyarat dan batasan

Asumsi

- Pola ini menjelaskan cara menyiapkan SQL Server Analysis Services dan Intellisoft OLE DB Provider atau CData ADO.NET Provider untuk Amazon Redshift pada instans Amazon Elastic Compute Cloud (Amazon). EC2 Atau, Anda dapat menginstal keduanya pada host di pusat data perusahaan Anda.

Prasyarat

- Akun AWS yang aktif
- Cluster Amazon Redshift dengan kredensil

Arsitektur

Tumpukan teknologi sumber

- Klaster Amazon Redshift

Tumpukan teknologi target

- Layanan Analisis Server Microsoft SQL

Arsitektur sumber dan target

Alat

- Microsoft Visual Studio 2019 (Edisi Komunitas)
- Penyedia Intellisoft OLE DB untuk Amazon Redshift (Uji Coba) atau CData Penyedia ADO.NET untuk Amazon Redshift (Uji Coba)

Epik

Menganalisis tabel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menganalisis tabel dan data yang akan diimpor.	Identifikasi tabel Amazon Redshift yang akan diimpor dan ukurannya.	DBA

Siapkan EC2 instans dan instal alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan sebuah EC2 instance.	Di akun AWS Anda, buat EC2 instance di subnet pribadi atau publik.	Administrator sistem
Instal alat untuk akses database.	Unduh dan instal Penyedia Intellisoft OLE DB untuk Amazon Redshift (atau CData Penyedia ADO.NET untuk Amazon Redshift).	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Visual Studio.	Unduh dan instal Visual Studio 2019 (Edisi Komunitas).	Administrator sistem
Instal ekstensi.	Instal ekstensi Microsoft Analysis Services Projects di Visual Studio.	Administrator sistem
Buat proyek.	Buat proyek model tabular baru di Visual Studio untuk menyimpan data Amazon Redshift Anda. Di Visual Studio, pilih opsi Analysis Services Tabular Project saat membuat proyek Anda.	DBA

Buat sumber data dan impor tabel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sumber data Amazon Redshift.	Buat sumber data Amazon Redshift dengan menggunakan Penyedia DB Intellisoft OLE untuk Amazon Redshift (atau Penyedia ADO.NET CData untuk Amazon Redshift) dan kredensi Amazon Redshift Anda.	Pergeseran Merah Amazon, DBA
Impor tabel.	Pilih dan impor tabel dan tampilan dari Amazon Redshift ke proyek SQL Server Analysis Services Anda.	Pergeseran Merah Amazon, DBA

Bersihkan setelah migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus EC2 instance.	Hapus EC2 instance yang Anda luncurkan sebelumnya.	Administrator sistem

Sumber daya terkait

- Amazon Redshift (dokumentasi AWS)
- Instal Layanan Analisis SQL Server (dokumentasi Microsoft)
- Desainer Model Tabular (dokumentasi Microsoft)
- Ikhtisar kubus OLAP untuk analitik lanjutan (dokumentasi Microsoft)
- Microsoft Visual Studio 2019 (Edisi Komunitas)
- Penyedia Intellisoft OLE DB untuk Amazon Redshift (Uji Coba)
- CData Penyedia ADO.NET untuk Amazon Redshift (Uji Coba)

Analisis dan visualisasikan data JSON bersarang dengan Amazon Athena dan Amazon Quick Sight

Anoop Singh, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerjemahkan struktur data berformat JSON yang bersarang ke dalam tampilan tabel dengan menggunakan Amazon Athena, dan kemudian memvisualisasikan data di Amazon Quick Sight.

Anda dapat menggunakan data berformat JSON untuk umpan data yang didukung API dari sistem operasional untuk membuat produk data. Data ini juga dapat membantu Anda memahami pelanggan Anda dan interaksi mereka dengan produk Anda dengan lebih baik, sehingga Anda dapat menyesuaikan pengalaman pengguna dan memprediksi hasil.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- File JSON yang mewakili struktur data bersarang (pola ini menyediakan file sampel)

Keterbatasan:

- Fitur JSON terintegrasi dengan baik dengan fungsi berorientasi SQL yang ada di Athena. Namun, mereka tidak kompatibel dengan ANSI SQL, dan file JSON diharapkan membawa setiap catatan pada baris terpisah. Anda mungkin perlu menggunakan `ignore.malformed.json` properti di Athena untuk menunjukkan apakah catatan JSON yang salah bentuk harus diubah menjadi karakter nol atau menghasilkan kesalahan. Untuk informasi selengkapnya, lihat Praktik terbaik untuk membaca data JSON di dokumentasi Athena.
- Pola ini hanya mempertimbangkan data berformat JSON dalam jumlah sederhana dan kecil. Jika Anda ingin menggunakan konsep-konsep ini dalam skala besar, pertimbangkan untuk menerapkan partisi data dan mengkonsolidasikan data Anda ke dalam file yang lebih besar.

Arsitektur

Diagram berikut menunjukkan arsitektur dan alur kerja untuk pola ini. Struktur data bersarang disimpan di Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dalam format JSON. Di Athena, data JSON dipetakan ke struktur data Athena. Anda kemudian membuat tampilan untuk menganalisis data, dan memvisualisasikan struktur data di Quick Sight.

Alat

Layanan AWS

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Pola ini menggunakan Amazon S3 untuk menyimpan file JSON.
- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar. Pola ini menggunakan Athena untuk query dan mengubah data JSON. Dengan beberapa tindakan di dalamnya Konsol Manajemen AWS, Anda dapat mengarahkan Athena ke data Anda di Amazon S3 dan menggunakan SQL standar untuk menjalankan kueri satu kali. Athena tanpa server, jadi tidak ada infrastruktur untuk mengatur atau mengelola, dan Anda hanya membayar untuk kueri yang Anda jalankan. Athena menskalakan secara otomatis dan menjalankan kueri secara paralel, sehingga hasilnya cepat, bahkan dengan kumpulan data besar dan kueri yang kompleks.
- Amazon Quick Sight adalah layanan intelijen bisnis skala cloud (BI) yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda di satu dasbor. Quick Sight memungkinkan Anda dengan mudah membuat dan mempublikasikan dasbor interaktif yang menyertakan wawasan pembelajaran mesin (ML). Anda dapat mengakses dasbor ini dari perangkat apa pun, dan menyematkannya ke aplikasi, portal, dan situs web Anda.

Contoh kode

File JSON berikut menyediakan struktur data bersarang yang dapat Anda gunakan dalam pola ini.

```
{
  "symbol": "AAPL",
  "financials": [
    {
      "reportDate": "2017-03-31",
```

```
      "grossProfit": 20591000000,
      "costOfRevenue": 32305000000,
      "operatingRevenue": 52896000000,
      "totalRevenue": 52896000000,
      "operatingIncome": 14097000000,
      "netIncome": 11029000000,
      "researchAndDevelopment": 2776000000,
      "operatingExpense": 6494000000,
      "currentAssets": 101990000000,
      "totalAssets": 334532000000,
      "totalLiabilities": 200450000000,
      "currentCash": 15157000000,
      "currentDebt": 13991000000,
      "totalCash": 67101000000,
      "totalDebt": 98522000000,
      "shareholderEquity": 134082000000,
      "cashChange": -1214000000,
      "cashFlow": 12523000000,
      "operatingGainsLosses": null
    }
  ]
}
```

Epik

Siapkan ember S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Untuk membuat bucket untuk menyimpan file JSON, masuk ke Konsol Manajemen AWS, buka konsol Amazon S3, lalu pilih Buat bucket. Untuk informasi selengkapnya, lihat Membuat bucket di dokumenta si Amazon S3.	Administrator sistem
Tambahkan data JSON bersarang.	Unggah file JSON Anda ke bucket S3. Untuk contoh file JSON, lihat bagian sebelumny	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	a. Untuk petunjuk, lihat Mengunggah objek dalam dokumentasi Amazon S3.	

Analisis data di Athena

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel untuk memetakan data JSON.	1. Buka konsol Athena. 2. Buat database dengan mengikuti petunjuk dalam dokumentasi Athena. 3. Dari menu Database, pilih database yang Anda buat. 4. Di editor kueri, masukkan `CREATE TABLE` pernyataan seperti berikut ini: `CREATE EXTERNAL TABLE financials_json ( symbol string, financials array< struct<reportdate: string, grossprofit: bigint, totalrevenue: bigint, totalcash: bigint, totaldebt: bigint, researchanddevelopment: bigint>> )`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>ROW FORMAT SERDE
	  'org.openx.data.js
	onserde.JsonSerDe'
	  LOCATION 's3://s3b
	ucket-for-athena/'</pre>
	dimana `LOCATION` menentukan lokasi bucket S3 yang berisi file JSON.
	5. Pilih Jalankan untuk membuat tabel.
	
	Untuk informasi selengkapnya tentang membuat tabel, lihat dokumentasi Athena.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tampilan untuk analisis data.	1. Buka konsol Athena. 2. Buat database dengan mengikuti petunjuk dalam dokumentasi Athena. 3. Dari menu Database, pilih database yang Anda buat. 4. Di editor kueri, masukkan `CREATE VIEW` pernyataan seperti berikut ini: `CREATE OR REPLACE VIEW financial_json_view AS SELECT symbol, financials[1].reportdate one_report_date, -- indexes start with 1 financials[1].totalrevenue one_total_revenue, financials[1].reportdate another_report_date, financials[1].totalrevenue another_total_revenue FROM financials_json where symbol='AAPL' ORDER BY 1` 5. Pilih Jalankan untuk membuat tampilan.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang membuat tampilan, lihat dokumentasi Athena.	
Menganalisis dan memvalidasi data.	1. Buka konsol Athena. 2. Di editor kueri, jalankan kueri dengan menggunakan tampilan yang Anda buat di langkah sebelumnya. 3. Validasi data terhadap file JSON, untuk mengonfirmasi bahwa nama kolom dan tipe data dipetakan dengan benar.	Developer

Visualisasikan data dalam Quick Sight

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Athena sebagai sumber data di Quick Sight.	1. Buka konsol Quick Suite. 2. Pilih Datasets, New Dataset. 3. Pilih Athena sebagai sumber data. 4. Pilih database yang menyertakan tampilan yang Anda buat. 5. Pilih tampilan yang ingin Anda buat dataset. 6. Pada halaman pembuatan kumpulan data Selesai, pilih Kueri data Anda secara langsung.	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Pilih Visualisasikan.	
Visualisasikan data dalam Quick Sight.	1. Setelah Anda memvisual isasikan kumpulan data, pilih visual dari panel kiri, dan pilih bidang untuk kumpulan data. Untuk informasi lebih lanjut, lihat tutorial di dokumentasi Quick Suite. 2. Simpan perubahan pada analisis. 3. Pilih Publikasikan dasbor untuk mempublikasikan visual yang Anda buat.	Analis data

Sumber daya terkait

- Dokumentasi Amazon Athena
- Tutorial Penglihatan Cepat Amazon
- Bekerja dengan JSON bersarang (posting blog)

Otomatiskan konsumsi data dari AWS Data Exchange Amazon S3

Adnan Alvee dan Manikanta Gona, Amazon Web Services

Ringkasan

Pola ini menyediakan CloudFormation template yang memungkinkan Anda untuk secara otomatis menyerap data dari AWS Data Exchange danau data Anda di Amazon Simple Storage Service (Amazon S3).

AWS Data Exchange adalah layanan yang memudahkan pertukaran kumpulan data berbasis file dengan aman di AWS Cloud. AWS Data Exchange kumpulan data berbasis langganan. Sebagai pelanggan, Anda juga dapat mengakses revisi kumpulan data saat penyedia mempublikasikan data baru.

CloudFormation Template membuat acara di Amazon CloudWatch Events dan sebuah AWS Lambda fungsi. Acara ini mengawasi pembaruan apa pun pada kumpulan data yang telah Anda langgani. Jika ada pembaruan, CloudWatch memulai fungsi Lambda, yang menyalin data ke bucket S3 yang Anda tentukan. Ketika data telah berhasil disalin, Lambda mengirimkan pemberitahuan Amazon Simple Notification Service (Amazon SNS) kepada Anda.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Berlangganan ke kumpulan data di AWS Data Exchange

Batasan

- CloudFormation Template harus digunakan secara terpisah untuk setiap set data berlangganan. AWS Data Exchange

Arsitektur

Tumpukan teknologi target

- AWS Lambda

- Amazon S3
- AWS Data Exchange
- Amazon CloudWatch
- Amazon SNS

Arsitektur target

Otomatisasi dan skala

Anda dapat menggunakan CloudFormation template beberapa kali untuk kumpulan data yang ingin Anda konsumsi ke dalam danau data.

Alat

- AWS Data Exchangememudahkan AWS pelanggan untuk bertukar kumpulan data berbasis file dengan aman di. AWS Cloud Sebagai pelanggan, Anda dapat menemukan dan berlangganan ratusan produk dari penyedia data yang memenuhi syarat. Kemudian, Anda dapat dengan cepat mengunduh kumpulan data atau menyalinnya ke Amazon S3 untuk digunakan di berbagai layanan AWS analisis dan pembelajaran mesin. Siapa pun yang memiliki Akun AWS dapat menjadi AWS Data Exchange pelanggan.
- AWS Lambda memungkinkan Anda menjalankan kode tanpa server provisioning atau pengelolaan. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda konsumsi; tidak ada biaya ketika kode Anda tidak berjalan. Dengan Lambda, Anda dapat menjalankan kode untuk hampir semua jenis aplikasi atau layanan backend tanpa administrasi. Lambda menjalankan kode Anda pada infrastruktur komputasi ketersediaan tinggi dan mengelola semua sumber daya komputasi, termasuk pemeliharaan server dan sistem operasi, penyediaan kapasitas dan penskalaan otomatis, pemantauan kode, dan pencatatan.
- Amazon S3 menyediakan penyimpanan untuk internet. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web.
- Amazon CloudWatch Events memberikan aliran peristiwa sistem yang mendekati real-time yang menjelaskan perubahan AWS sumber daya. Dengan menggunakan aturan sederhana yang dapat Anda atur dengan cepat, Anda dapat mencocokkan acara dan mengarahkannya ke satu atau lebih fungsi atau aliran target. CloudWatch Peristiwa menjadi sadar akan perubahan operasional saat terjadi. Ini menanggapi perubahan operasional ini dan mengambil tindakan korektif seperlunya,

dengan mengirim pesan untuk menanggapi lingkungan, mengaktifkan fungsi, membuat perubahan, dan menangkap informasi negara. Anda juga dapat menggunakan CloudWatch Acara untuk menjadwalkan tindakan otomatis yang dimulai sendiri pada waktu-waktu tertentu menggunakan ekspresi cron atau rate.

- Amazon Simple Notification Service (Amazon SNS) memungkinkan aplikasi, pengguna akhir, dan perangkat untuk langsung mengirim dan menerima notifikasi dari cloud. Amazon SNS menyediakan topik (saluran komunikasi) untuk pengiriman pesan dengan throughput tinggi, berbasis push. many-to-many Menggunakan topik Amazon SNS, penerbit dapat mendistribusikan pesan ke sejumlah besar pelanggan untuk pemrosesan paralel, termasuk antrian Amazon Simple Queue Service (Amazon SQS), fungsi Lambda, dan webhook. HTTP/S Anda juga dapat menggunakan Amazon SNS untuk mengirim notifikasi kepada pengguna akhir menggunakan push seluler, SMS, dan email.

Epik

Berlangganan kumpulan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berlangganan kumpulan data.	Di AWS Data Exchange konsol, berlangganan dataset. Untuk petunjuk, lihat Berlangganan produk data AWS Data Exchange dalam AWS dokumentasi.	AWS Umum
Perhatikan atribut kumpulan data.	Perhatikan Wilayah AWS, ID, dan ID revisi untuk kumpulan data. Anda akan membutuhkan ini untuk CloudFormation template di langkah berikutnya.	AWS Umum

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember dan folder S3.	Jika Anda sudah memiliki data lake di Amazon S3, buat folder untuk menyimpan data untuk dicerna. AWS Data Exchange Jika Anda menerapkan template untuk tujuan pengujian, buat bucket S3 baru, dan catat nama bucket dan awalan folder untuk langkah berikutnya.	AWS Umum
Menyebarkan CloudFormation template.	Terapkan CloudFormation template yang disediakan sebagai lampiran ke pola ini. Untuk instruksi, lihat CloudFormation dokumentasi. Konfigurasikan parameter berikut agar sesuai dengan pengaturan bucket Akun AWS, kumpulan data, dan S3 Anda: Dataset Wilayah AWS, ID Dataset, ID Revisi, Nama Bucket S3 (misalnya ,**DOC-EXAMPLE-BUCKET**), Awalan Folder (misalnya,), dan Email untuk Pemberita huan **myfolder/** SNS. Anda dapat mengatur parameter Nama Dataset ke nama apa pun. Saat Anda menerapkan template, ia menjalankan fungsi Lambda	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk secara otomatis menyerap kumpulan data pertama yang tersedia di kumpulan data. Konsumsi selanjutnya kemudian terjadi secara otomatis, karena data baru tiba di kumpulan data.	

Sumber daya terkait

- Berlangganan produk data pada AWS Data Exchange (AWS Data Exchange dokumentasi)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Otomatiskan penegakan enkripsi di AWS Glue menggunakan templat AWS CloudFormation

Diogo Guedes, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara mengatur dan mengotomatiskan penegakan enkripsi di AWS Glue dengan menggunakan CloudFormation templat AWS. Template membuat semua konfigurasi dan sumber daya yang diperlukan untuk menegakkan enkripsi. Sumber daya ini mencakup konfigurasi awal, kontrol pencegahan yang dibuat oleh EventBridge aturan Amazon, dan fungsi AWS Lambda.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Izin untuk menyebarkan CloudFormation template dan sumber dayanya

Batasan

Kontrol keamanan ini bersifat regional. Anda harus menerapkan kontrol keamanan di setiap Wilayah AWS tempat Anda ingin mengatur penegakan enkripsi di AWS Glue.

Arsitektur

Tumpukan teknologi target

- CloudWatch Log Amazon (dari AWS Lambda)
- EventBridge Aturan Amazon
- AWS CloudFormation tumpukan
- AWS CloudTrail
- Peran dan kebijakan yang dikelola AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)
- Alias AWS KMS

- Fungsi AWS Lambda
- AWS Systems Manager Parameter Store

Arsitektur target

Diagram berikut menunjukkan cara mengotomatiskan penegakan enkripsi di AWS Glue.

Diagram menunjukkan alur kerja berikut:

1. CloudFormation Template membuat semua sumber daya, termasuk konfigurasi awal dan kontrol detektif untuk penegakan enkripsi di AWS Glue.
2. EventBridge Aturan mendeteksi perubahan status dalam konfigurasi enkripsi.
3. Fungsi Lambda dipanggil untuk evaluasi dan pencatatan melalui Log. CloudWatch Untuk deteksi yang tidak sesuai, Parameter Store dipulihkan dengan Amazon Resource Name (ARN) untuk kunci AWS KMS. Layanan ini diperbaiki ke status yang sesuai dengan enkripsi diaktifkan.

Otomatisasi dan skala

Jika Anda menggunakan AWS Organizations, Anda dapat menggunakan AWS CloudFormation StackSets untuk menerapkan template ini di beberapa akun tempat Anda ingin mengaktifkan penegakan enkripsi di AWS Glue.

Alat

- Amazon CloudWatch membantu Anda memantau metrik sumber daya AWS Anda dan aplikasi yang Anda jalankan di AWS secara real time.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, fungsi Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun AWS lainnya.
- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS CloudTrail membantu Anda mengaktifkan audit operasional dan risiko, tata kelola, dan kepatuhan akun AWS Anda.

- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Systems Manager membantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan manajemen aplikasi dan sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola sumber daya AWS Anda dengan aman dalam skala besar.

Kode

Kode untuk pola ini tersedia di repositori GitHub aws-custom-guardrail-event-driven.

Praktik terbaik

AWS Glue mendukung enkripsi data saat istirahat untuk membuat pekerjaan di AWS Glue dan mengembangkan skrip menggunakan titik akhir pengembangan.

Pertimbangkan praktik terbaik berikut:

- Konfigurasikan pekerjaan ETL dan titik akhir pengembangan untuk menggunakan kunci AWS KMS untuk menulis data terenkripsi saat istirahat.
- Enkripsi metadata yang disimpan dalam AWS Glue Data Catalog dengan menggunakan kunci yang Anda kelola melalui AWS KMS.
- Gunakan kunci AWS KMS untuk mengenkripsi bookmark pekerjaan dan log yang dihasilkan oleh crawler dan pekerjaan ETL.

Epik

Luncurkan CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	Unduh `aws-custom-guardrail-event-driven.yaml` template dari GitHub repositori, lalu gunakan template. `CREATE_COMPLETE` Status menunjukkan bahwa template Anda berhasil di-deploy. **Note** Template tidak memerlukan parameter input.	Arsitek awan

Verifikasi pengaturan enkripsi di AWS Glue

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa konfigurasi kunci AWS KMS.	1. Masuk ke AWS Management Console, lalu buka konsol AWS Glue. 2. Di panel navigasi, di bawah Katalog Data, pilih Pengaturan katalog. 3. Verifikasi bahwa enkripsi Metadata dan enkripsi pengaturan kata sandi koneksi ditandai dan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dikonfigurasi untuk digunakan. `KMSKeyGlue`	

Uji penegakan enkripsi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi pengaturan enkripsi di CloudFormation.	1. Masuk ke AWS Management Console lalu buka CloudFormation konsol. 2. Di panel navigasi, pilih Stacks, lalu pilih tumpukan Anda. 3. Pilih tab Sumber Daya. 4. Dalam tabel Resources , temukan pengaturan enkripsi dengan Logical ID.	Arsitek awan
Alihkan infrastruktur yang disediakan ke status tidak patuh.	1. Masuk ke AWS Management Console, lalu buka konsol AWS Glue. 2. Di panel navigasi, di bawah Katalog Data, pilih Pengaturan katalog. 3. Kosongkan kotak centang enkripsi Metadata. 4. Kosongkan kotak centang Enkripsi kata sandi koneksi. 5. Pilih Simpan. 6. Segarkan konsol AWS Glue.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pagar pembatas mendeteksi status tidak sesuai di AWS Glue setelah Anda mengosongkan kotak centang, lalu memberlakukan kepatuhan dengan memulihkan kesalahan konfigurasi enkripsi secara otomatis. Akibatnya, kotak centang enkripsi harus dipilih kembali setelah Anda me-refresh halaman.	

Sumber daya terkait

- Membuat tumpukan di CloudFormation konsol AWS (CloudFormation dokumentasi AWS)
- Membuat aturan CloudWatch Peristiwa yang dipicu pada panggilan AWS API menggunakan AWS CloudTrail (CloudWatch dokumentasi Amazon)
- Menyiapkan enkripsi di AWS Glue (dokumentasi AWS Glue)

Membangun pipeline data untuk menyerap, mengubah, dan menganalisis data Google Analytics menggunakan AWS DataOps Development Kit

Anton Kukushkin dan Rudy Puig, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara membuat pipeline data untuk menyerap, mengubah, dan menganalisis data Google Analytics dengan menggunakan AWS DataOps Development Kit (AWS DDK) dan lainnya. Layanan AWS AWS DDK adalah kerangka kerja pengembangan sumber terbuka yang membantu Anda membangun alur kerja data dan arsitektur data modern. AWS Salah satu tujuan utama AWS DDK adalah untuk menghemat waktu dan upaya yang biasanya dikhususkan untuk tugas-tugas pipa data padat karya, seperti mengatur jaringan pipa, membangun infrastruktur, dan menciptakan infrastruktur di balik itu. DevOps Anda dapat menurunkan tugas-tugas padat karya ini ke AWS DDK sehingga Anda dapat fokus pada penulisan kode dan aktivitas bernilai tinggi lainnya.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AppFlow Konektor Amazon untuk Google Analytics, dikonfigurasi
- Python dan pip (manajer paket Python)
- Git, diinstal dan dikonfigurasi
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- AWS Cloud Development Kit (AWS CDK), dipasang

Versi produk

- Python 3.7 atau yang lebih baru
- pip 9.0.3 atau yang lebih baru

Arsitektur

Tumpukan teknologi

- Amazon AppFlow
- Amazon Athena
- Amazon CloudWatch
- Amazon EventBridge
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Queue Service (Amazon SQS)
- AWS DataOps Kit Pengembangan (AWS DDK)
- AWS Lambda

Arsitektur target

Diagram berikut menunjukkan proses berbasis peristiwa yang menyerap, mengubah, dan menganalisis data Google Analytics.

Diagram menunjukkan alur kerja berikut:

1. Aturan acara CloudWatch terjadwal Amazon memanggil Amazon AppFlow.
2. Amazon AppFlow menyerap data Google Analytics ke dalam bucket S3.
3. Setelah data dicerna oleh bucket S3, notifikasi peristiwa akan EventBridge dibuat, ditangkap oleh aturan CloudWatch Peristiwa, dan kemudian dimasukkan ke dalam antrean Amazon SQS.
4. Fungsi Lambda mengkonsumsi peristiwa dari antrian Amazon SQS, membaca objek S3 masing-masing, mengubah objek ke format Apache Parquet, menulis objek yang diubah ke bucket S3, dan kemudian membuat atau memperbarui definisi tabel. AWS Glue Data Catalog
5. Kueri Athena berjalan melawan tabel.

Alat

AWS alat

- Amazon AppFlow adalah layanan integrasi yang dikelola sepenuhnya yang memungkinkan Anda bertukar data dengan aman antara aplikasi perangkat lunak sebagai layanan (SaaS).
- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar.

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Cloud Development Kit (AWS CDK)adalah kerangka kerja untuk mendefinisikan infrastruktur cloud dalam kode dan menyediakannya. CloudFormation
- AWS DataOps Development Kit (AWS DDK) adalah kerangka kerja pengembangan sumber terbuka untuk membantu Anda membangun alur kerja data dan arsitektur data modern. AWS

Kode

Kode untuk pola ini tersedia di GitHub AWS DataOps Development Kit (AWS DDK) dan Menganalisis data Google Analytics dengan repositori Amazon AppFlow, Amazon Athena, AWS DataOps dan Development Kit.

Epik

Siapkan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning kode sumber.	Untuk mengkloning kode sumber, jalankan perintah berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/aws-samples/aws-ddk-examples.git`	
Buat lingkungan virtual.	Arahkan ke direktori kode sumber, lalu jalankan perintah berikut untuk membuat lingkungan virtual: `cd google-analytics-data-using-appflow/python && python3 -m venv .venv`	DevOps insinyur
Instal dependensi.	Untuk mengaktifkan lingkungan virtual dan menginstal dependensi, jalankan perintah berikut: `source .venv/bin/activate && pip install -r requirements.txt`	DevOps insinyur

Menerapkan aplikasi yang menggunakan pipeline data Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap lingkungan.	1. Konfirmasikan bahwa AWS CLI sudah diatur dengan kredensil yang valid untuk Anda. Akun AWS Untuk informasi selengkapnya, lihat Menggunakan profil	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bernama dalam AWS CLI dokumentasi. 2. Jalankan perintah `cdk bootstrap --profile [AWS_PROFILE]` .	
Menyebarkan data.	Untuk menyebarkan pipa data, jalankan `cdk deploy --profile [AWS_PROFILE]` perintah.	DevOps insinyur

Uji deployment

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi status tumpukan.	1. Buka konsol CloudFormation. 2. Pada halaman Stacks, konfirmasikan bahwa status tumpukan `DdkAppflowAthenaStack` adalah`CREATE_COMPLETE` .	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Penerapan gagal selama pembuatan `AWS::AppFlow::Flow` sumber daya dan Anda menerima kesalahan berikut: `Connector Profile with name ga-connection does not exist`	Konfirmasikan bahwa Anda membuat AppFlow konektor Amazon untuk Google Analytics dan menamakannya`ga-connection` .

Isu	Solusi
	Untuk petunjuk, lihat Google Analytics di AppFlow dokumentasi Amazon.

Sumber daya terkait

- AWS DataOps Kit Pengembangan (AWS DDK) (GitHub)
- AWS Contoh DDK () GitHub

Informasi tambahan

AWS Pipa data DDK terdiri dari satu atau banyak tahapan. Dalam contoh kode berikut, Anda gunakan `AppFlowIngestionStage` untuk menyerap data dari Google Analytics, `SqsToLambdaStage` menangani transformasi data, dan `AthenaSQLStage` menjalankan kueri Athena.

Pertama, transformasi data dan tahap konsumsi dibuat, seperti contoh kode berikut menunjukkan:

```
        appflow_stage = AppFlowIngestionStage(
            self,
            id="appflow-stage",
            flow_name=flow.flow_name,
        )
        sqs_lambda_stage = SqsToLambdaStage(
            self,
            id="lambda-stage",
            lambda_function_props={
                "code": Code.from_asset("./ddk_app/lambda_handlers"),
                "handler": "handler.lambda_handler",
                "layers": [
                    LayerVersion.from_layer_version_arn(
                        self,
                        id="layer",
                        layer_version_arn=f"arn:aws:lambda:
{self.region}:336392948345:layer:AWSDataWrangler-Python39:1",
                    )
                ],
                "runtime": Runtime.PYTHON_3_9,
            },
```

```
        )
        # Grant lambda function S3 read & write permissions
        bucket.grant_read_write(sqs_lambda_stage.function)
        # Grant Glue database & table permissions
        sqs_lambda_stage.function.add_to_role_policy(
            self._get_glue_db_iam_policy(database_name=database.database_name)
        )
        athena_stage = AthenaSQLStage(
            self,
            id="athena-sql",
            query_string=[
                (
                    "SELECT year, month, day, device, count(user_count) as cnt "
                    f"FROM {database.database_name}.ga_sample "
                    "GROUP BY year, month, day, device "
                    "ORDER BY cnt DESC "
                    "LIMIT 10; "
                )
            ],
            output_location=Location(
                bucket_name=bucket.bucket_name, object_key="query-results/"
            ),
            additional_role_policy_statements=[
                self._get_glue_db_iam_policy(database_name=database.database_name)
            ],
        )
```

Selanjutnya, `DataPipeline` konstruksi digunakan untuk “menghubungkan” tahapan bersama-sama dengan menggunakan EventBridge aturan, seperti contoh kode berikut menunjukkan:

```
        (
            DataPipeline(self, id="ingestion-pipeline")
            .add_stage(
                stage=appflow_stage,
                override_rule=Rule(
                    self,
                    "schedule-rule",
                    schedule=Schedule.rate(Duration.hours(1)),
                    targets=appflow_stage.targets,
                ),
            )
            .add_stage(
                stage=sqs_lambda_stage,
```

```
                # By default, AppFlowIngestionStage stage emits an event after the flow
run finishes successfully
                # Override rule below changes that behavior to call the the stage when
data lands in the bucket instead
                override_rule=Rule(
                    self,
                    "s3-object-created-rule",
                    event_pattern=EventPattern(
                        source=["aws.s3"],
                        detail={
                            "bucket": {"name": [bucket.bucket_name]},
                            "object": {"key": [{"prefix": "ga-data"}]},
                        },
                        detail_type=["Object Created"],
                    ),
                    targets=sqs_lambda_stage.targets,
                ),
            )
            .add_stage(stage=athena_stage)
        )
```

Untuk contoh kode lainnya, lihat GitHub Menganalisis data Google Analytics dengan Amazon AppFlow, Amazon Athena, dan repositori AWS DataOps Development Kit.

Buat pipeline pemrosesan video dengan menggunakan Amazon Kinesis Video Streams dan AWS Fargate

Piotr Chotkowski dan Pushparaju Thangavel, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menggunakan Amazon Kinesis Video Streams dan AWS Fargate untuk mengekstrak bingkai dari aliran video dan menyimpannya sebagai file gambar untuk diproses lebih lanjut di Amazon Simple Storage Service (Amazon S3).

Pola ini menyediakan contoh aplikasi dalam bentuk proyek Java Maven. Aplikasi ini mendefinisikan infrastruktur AWS dengan menggunakan AWS Cloud Development Kit (AWS CDK). Baik logika pemrosesan bingkai dan definisi infrastruktur ditulis dalam bahasa pemrograman Java. Anda dapat menggunakan aplikasi sampel ini sebagai dasar untuk mengembangkan pipa pemrosesan video real-time Anda sendiri atau untuk membangun langkah pra-pemrosesan video dari pipa pembelajaran mesin.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Kit Pengembangan Java SE (JDK) 11, diinstal
- Apache Maven, diinstal
- AWS Cloud Development Kit (AWS CDK), diinstal
- AWS Command Line Interface (AWS CLI) versi 2, diinstal
- Docker (diperlukan untuk membuat gambar Docker untuk digunakan dalam definisi tugas AWS Fargate), diinstal

Batasan

Pola ini dimaksudkan sebagai bukti konsep, atau sebagai dasar untuk pengembangan lebih lanjut. Seharusnya tidak digunakan dalam bentuknya saat ini dalam penyebaran produksi.

Versi produk

- Pola ini diuji dengan AWS CDK versi 1.77.0 (lihat versi AWS CDK)

- JDK 11
- AWS CLI versi 2

Arsitektur

Tumpukan teknologi target

- Amazon Kinesis Video Streams
- Tugas AWS Fargate
- Antrean Amazon Simple Queue Service (Amazon SQS)
- Buket Amazon S3

Arsitektur target

Pengguna membuat aliran video Kinesis, mengunggah video, dan mengirimkan pesan JSON yang berisi rincian tentang input Kinesis video stream dan output S3 bucket ke antrian SQS. AWS Fargate, yang menjalankan aplikasi utama dalam wadah, menarik pesan dari antrian SQS dan mulai mengekstrak bingkai. Setiap frame disimpan dalam file gambar dan disimpan di bucket S3 target.

Otomatisasi dan skala

Aplikasi sampel dapat menskalakan baik secara horizontal maupun vertikal dalam satu Wilayah AWS. Penskalaan horizontal dapat dicapai dengan meningkatkan jumlah tugas AWS Fargate yang diterapkan yang dibaca dari antrian SQS. Penskalaan vertikal dapat dicapai dengan meningkatkan jumlah utas pemecahan bingkai dan penerbitan gambar dalam aplikasi. Pengaturan ini diteruskan sebagai variabel lingkungan ke aplikasi dalam definisi QueueProcessingFargateServicesumber daya di AWS CDK. Karena sifat penerapan tumpukan AWS CDK, Anda dapat menerapkan aplikasi ini di beberapa Wilayah AWS dan akun tanpa upaya tambahan.

Alat

Alat

- AWS CDK adalah kerangka kerja pengembangan perangkat lunak untuk mendefinisikan infrastruktur dan sumber daya cloud Anda dengan menggunakan bahasa pemrograman seperti TypeScript,, JavaScript Python, Java, dan C #/.Net.

- Amazon Kinesis Video Streams adalah layanan AWS yang dikelola sepenuhnya yang dapat Anda gunakan untuk melakukan streaming video langsung dari perangkat ke AWS Cloud, atau membuat aplikasi untuk pemrosesan video real-time atau analitik video berorientasi batch.
- AWS Fargate adalah mesin komputasi tanpa server untuk kontainer. Fargate menghilangkan kebutuhan untuk menyediakan dan mengelola server, dan memungkinkan Anda fokus pada pengembangan aplikasi Anda.
- Amazon S3 adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja.
- Amazon SQS adalah layanan antrian pesan yang dikelola sepenuhnya yang memungkinkan Anda memisahkan dan menskalakan layanan mikro, sistem terdistribusi, dan aplikasi tanpa server.

Kode

- File.zip dari proyek aplikasi sampel (`frame-splitter-code.zip`) dilampirkan.

Epik

Menyebarkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai daemon Docker.	Mulai daemon Docker di sistem lokal Anda. AWS CDK menggunakan Docker untuk membuat image yang digunakan dalam tugas AWS Fargate. Anda harus menjalankan Docker sebelum melanjutkan ke langkah berikutnya.	Pengembang, DevOps insinyur
Bangun proyek.	Unduh aplikasi `frame-splitter-code` sampel (terlampir) dan ekstrak isinya ke dalam folder di mesin lokal Anda. Sebelum Anda dapat	Pengembang, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menyebarkan infrastruktur, Anda harus membangun proyek Java Maven. Pada prompt perintah, navigasikan ke direktori root proyek, dan buat proyek dengan menjalankan perintah: `mvn clean install`	
Bootstrap AWS CDK.	(Hanya pengguna AWS CDK pertama kali) Jika ini adalah pertama kalinya Anda menggunakan AWS CDK, Anda mungkin harus mem-bootstrap lingkungan dengan menjalankan perintah AWS CLI: `cdk bootstrap --profile "$AWS_PROFILE_NAME"` dimana `$AWS_PROFILE_NAME` memegang nama profil AWS dari kredensi AWS Anda. Atau, Anda dapat menghapus parameter ini untuk menggunakan profil default. Untuk informasi selengkapnya, lihat dokumentasi AWS CDK.	Pengembang, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan tumpukan AWS CDK.	Pada langkah ini, Anda membuat sumber daya infrastruktur yang diperluka n (antrean SQS, bucket S3, definisi tugas AWS Fargate) di akun AWS Anda, membuat image Docker yang diperlukan untuk tugas AWS Fargate, dan menerapkan aplikasi. Pada prompt perintah, arahkan ke direktori root proyek, dan jalankan perintah: `cdk deploy --profile "$AWS_PROFILE_NAME" --all` dimana `$AWS_PROF ILE_NAME` memegang nama profil AWS dari kredensi AWS Anda. Atau, Anda dapat menghapus parameter ini untuk menggunakan profil default. Konfirmasikan penyebaran. Perhatikan nilai QueueUrldan Bucket dari keluaran penyebaran CDK; Anda akan memerluka nnya di langkah selanjutn ya. AWS CDK membuat aset, mengunggahnya ke akun AWS Anda, dan membuat semua sumber daya infrastruktur. Anda dapat mengamati proses pembuatan	Pengembang, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber daya di CloudFormation konsol AWS. Untuk informasi selengkapnya, lihat CloudFormation dokumentasi AWS dan dokumentasi AWS CDK.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aliran video.	Pada langkah ini, Anda membuat aliran video Kinesis yang akan berfungsi sebagai aliran input untuk pemrosesan video. Pastikan AWS CLI telah diinstal dan dikonfigurasi. Di AWS CLI, jalankan: `aws kinesisvideo --profile "$AWS_PROFILE_NAME" create-stream --stream-name "$STREAM_NAME" --data-retention-in-hours "24"` dimana `$AWS_PROFILE_NAME` memegang nama profil AWS dari kredensyal AWS Anda (atau hapus parameter ini untuk menggunakan profil default) dan `$STREAM_NAME` merupakan nama aliran yang valid. Atau, Anda dapat membuat aliran video dengan menggunakan konsol Kinesis dengan mengikuti langkah-langkah dalam dokumentasi Kinesis Video Streams. Perhatikan AWS Resource Name (ARN) dari aliran yang	Pengembang, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dibuat; Anda akan membutuhk annya nanti.	

Jalankan contoh

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah video ke streaming.	Di folder proyek untuk `frame-splitter-code` aplikasi sampel, buka `ProcessingTaskTest.java` file di `src/test/java/amazon/awscdk/examples/splitter` folder. Ganti `streamName` variabel `profileName` dan dengan nilai yang Anda gunakan pada langkah sebelumnya. Untuk mengunggah contoh video ke aliran video Kinesis yang Anda buat pada langkah sebelumnya, jalankan: `amazon.awscdk.examples.splitter.ProcessingTaskTest#testExample test` Atau, Anda dapat mengunggah video Anda dengan menggunakan salah satu metode yang dijelaskan dalam dokumentasi Kinesis Video Streams.	Pengembang, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Memulai pemrosesan video.	Sekarang setelah Anda mengunggah video ke aliran video Kinesis, Anda dapat mulai memprosesnya. Untuk memulai logika pemrosesan, Anda harus mengirim pesan dengan detail ke antrian SQS yang dibuat AWS CDK selama penerapan. Untuk mengirim pesan menggunakan AWS CLI, jalankan:
	`aws sqs --profile "$AWS_PROFILE_NAME" send-message --queue-url QUEUE_URL --message-body MESSAGE`
	dimana `$AWS_PROFILE_NAME` memegang nama profil AWS dari kredensyal AWS Anda (hapus parameter ini untuk menggunakan profil default), `QUEUE_URL` adalah `QueueUrl`nilai dari output AWS CDK, dan `MESSAGE` merupakan string JSON dalam format berikut:
	`{ "streamARN": "STREAM_ARN", "bucket": "BUCKET_NAME", "s3Directory": "test-output" }`
		Pengembang, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di `STREAM_ARN` mana ARN dari aliran video yang Anda buat pada langkah sebelumnya dan `BUCKET_NAME` merupakan nilai Bucket dari output AWS CDK. Mengirim pesan ini memulai pemrosesan video. Atau, Anda dapat mengirim pesan menggunakan konsol Amazon SQS, seperti yang dijelaskan dalam dokumentasi Amazon SQS.	
Lihat gambar bingkai video.	Anda dapat melihat gambar yang dihasilkan di bucket `s3://BUCKET_NAME/test-output` keluaran S3 yang `BUCKET_NAME` merupakan nilai Bucket dari output AWS CDK.	Pengembang, DevOps insinyur

Sumber daya terkait

- Dokumentasi AWS CDK
- Referensi AWS CDK API
- Lokakarya pengantar AWS CDK
- Dokumentasi Amazon Kinesis Video Streams
- Contoh: Mengidentifikasi Objek dalam Streaming Video Menggunakan SageMaker
- Contoh: Mengurai dan Merender Fragmen Aliran Video Kinesis
- Analisis video langsung dalam skala secara real time menggunakan Amazon Kinesis Video Streams SageMaker dan Amazon (postingan blog AWS Machine Learning)

- AWS Fargate Memulai

Informasi tambahan

Memilih IDE

Kami menyarankan Anda menggunakan IDE Java favorit Anda untuk membangun dan menjelajahi proyek ini.

Membersihkan

Setelah Anda selesai menjalankan contoh ini, hapus semua sumber daya yang diterapkan untuk menghindari biaya infrastruktur AWS tambahan.

Untuk menghapus infrastruktur dan aliran video, gunakan dua perintah ini di AWS CLI:

```
cdk destroy --profile "$AWS_PROFILE_NAME" --all
```

```
aws kinesisvideo --profile "$AWS_PROFILE_NAME" delete-stream --stream-arn "$STREAM_ARN"
```

Atau, Anda dapat menghapus sumber daya secara manual dengan menggunakan CloudFormation konsol AWS untuk menghapus CloudFormation tumpukan AWS, dan konsol Kinesis untuk menghapus aliran video Kinesis. Perhatikan bahwa `cdk destroy` tidak menghapus bucket S3 keluaran atau gambar di repositori Amazon Elastic Container Registry (Amazon ECR) (). `aws-cdk/assets` Anda harus menghapusnya secara manual.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat pipeline layanan ETL untuk memuat data secara bertahap dari Amazon S3 ke Amazon Redshift menggunakan AWS Glue

Rohan Jamadagni dan Arunabha Datta, Amazon Web Services

Ringkasan

Pola ini memberikan panduan tentang cara mengonfigurasi Amazon Simple Storage Service (Amazon S3) untuk kinerja data lake yang optimal, lalu memuat perubahan data tambahan dari Amazon S3 ke Amazon Redshift dengan menggunakan AWS Glue, melakukan operasi ekstrak, transformasi, dan pemuatan (ETL).

File sumber di Amazon S3 dapat memiliki format yang berbeda, termasuk nilai yang dipisahkan koma (CSV), XML, dan file JSON. Pola ini menjelaskan bagaimana Anda dapat menggunakan AWS Glue untuk mengonversi file sumber menjadi format yang dioptimalkan biaya dan dioptimalkan kinerja seperti Apache Parquet. Anda dapat menanyakan file Parket langsung dari Amazon Athena dan Amazon Redshift Spectrum. Anda juga dapat memuat file Parket ke Amazon Redshift, menggabungkannya, dan membagikan data agregat dengan konsumen, atau memvisualisasikan data dengan menggunakan Amazon Quick Sight.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Bucket sumber S3 yang memiliki hak istimewa yang tepat dan berisi file CSV, XHTML, atau JSON.

Asumsi

- File sumber CSV, XHTML, atau JSON sudah dimuat ke Amazon S3 dan dapat diakses dari akun tempat AWS Glue dan Amazon Redshift dikonfigurasi.
- Praktik terbaik untuk memuat file, memisahkan file, kompresi, dan menggunakan manifes diikuti, seperti yang dibahas dalam dokumentasi Amazon Redshift.
- Struktur file sumber tidak berubah.
- Sistem sumber dapat menelan data ke Amazon S3 dengan mengikuti struktur folder yang ditentukan di Amazon S3.

- Cluster Amazon Redshift mencakup satu Availability Zone. (Arsitektur ini sesuai karena AWS Lambda, AWS Glue, dan Amazon Athena tidak memiliki server.) Untuk ketersediaan tinggi, snapshot cluster diambil pada frekuensi reguler.

Batasan

- Format file terbatas pada format yang saat ini didukung oleh AWS Glue.
- Pelaporan hilir real-time tidak didukung.

Arsitektur

Tumpukan teknologi sumber

- Bucket S3 dengan file CSV, XML, atau JSON

Tumpukan teknologi target

- Danau data S3 (dengan penyimpanan file Parket yang dipartisi)
- Amazon Redshift

Arsitektur target

Aliran data

Alat

- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel. Amazon S3 dapat digunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan data lake.
- AWS Lambda — AWS Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. AWS Lambda adalah layanan berbasis peristiwa; Anda dapat mengatur kode untuk memulai secara otomatis dari layanan AWS lainnya.

- Amazon Redshift — Amazon Redshift adalah layanan gudang data skala petabyte yang dikelola sepenuhnya. Dengan Amazon Redshift, Anda dapat menanyakan petabyte data terstruktur dan semi-terstruktur di seluruh gudang data dan data lake menggunakan SQL standar.
- AWS Glue — AWS Glue adalah layanan ETL yang dikelola sepenuhnya yang memudahkan penyiapan dan pemuatan data untuk analitik. AWS Glue menemukan data Anda dan menyimpan metadata terkait (misalnya, definisi tabel dan skema) di AWS Glue Data Catalog. Data katalog Anda segera dapat dicari, dapat ditanyakan, dan tersedia untuk ETL.
- AWS Secrets Manager — AWS Secrets Manager memfasilitasi perlindungan dan pengelolaan rahasia terpusat yang diperlukan untuk akses aplikasi atau layanan. Layanan ini menyimpan kredensi database, kunci API, dan rahasia lainnya, dan menghilangkan kebutuhan untuk hardcode informasi sensitif dalam format teks biasa. Secrets Manager juga menawarkan rotasi kunci untuk memenuhi kebutuhan keamanan dan kepatuhan. Ini memiliki integrasi bawaan untuk Amazon Redshift, Amazon Relational Database Service (Amazon RDS), dan Amazon DocumentDB. Anda dapat menyimpan dan mengelola rahasia secara terpusat dengan menggunakan konsol Secrets Manager, antarmuka baris perintah (CLI), atau Secrets Manager API dan. SDKs
- Amazon Athena - Amazon Athena adalah layanan kueri interaktif yang memudahkan untuk menganalisis data yang disimpan di Amazon S3. Athena tanpa server dan terintegrasi dengan AWS Glue, sehingga dapat langsung menanyakan data yang dikatalogkan menggunakan AWS Glue. Athena diskalakan secara elastis untuk memberikan kinerja kueri interaktif.

Epik

Buat bucket S3 dan struktur folder

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menganalisis sistem sumber untuk struktur dan atribut data.	Lakukan tugas ini untuk setiap sumber data yang berkontri busi ke danau data Amazon S3.	Insinyur data
Tentukan partisi dan strategi akses.	Strategi ini harus didasarkan pada frekuensi pengambilan data, pemrosesan delta, dan kebutuhan konsumsi. Pastikan bucket S3 tidak terbuka untuk	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	umum dan akses dikendali kan oleh kebijakan berbasis peran layanan tertentu saja. Untuk informasi lebih lanjut, lihat Dokumentasi Amazon S3.	
Buat bucket S3 terpisah untuk setiap tipe sumber data dan bucket S3 terpisah per sumber untuk data yang diproses (Parket).	Buat bucket terpisah untuk setiap sumber, lalu buat struktur folder yang didasarka n pada frekuensi penyerapan data sistem sumber; misalnya, . `s3://source-system-name/date/hour` Untuk file yang diproses (dikonversi ke format Parket), buat struktur yang serupa; misalnya,`s3://source-processed-bucket/date/hour` . Untuk informasi selengkapnya tentang membuat bucket S3, lihat dokumentasi Amazon S3.	Insinyur data

Buat gudang data di Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan klaster Amazon Redshift dengan grup parameter serta strategi pemeliharaan dan pencadang an yang sesuai.	Gunakan rahasia database Secrets Manager untuk kredensi pengguna admin saat membuat klaster Amazon Redshift. Untuk informasi tentang membuat dan mengukur cluster Amazon Redshift, lihat dokumenta si Amazon Redshift dan	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	whitepaper Sizing Cloud Data Warehouses.	
Buat dan lampirkan peran layanan IAM ke cluster Amazon Redshift.	Peran layanan AWS Identity and Access Managemen t (IAM) memastikan akses ke Secrets Manager dan bucket S3 sumber. Untuk informasi selengkapnya, lihat dokumentasi AWS tentang otorisasi dan penambahan peran.	Insinyur data
Buat skema database.	Ikuti praktik terbaik Amazon Redshift untuk desain tabel. Berdasarkan kasus penggunaan, pilih kunci pengurutan dan distribusi yang sesuai, dan pengkodea n kompresi terbaik. Untuk praktik terbaik, lihat dokumentasi AWS.	Insinyur data
Konfigurasikan manajemen beban kerja.	Konfigurasi antrian manajemen beban kerja (WLM), akselerasi kueri pendek (SQA), atau penskalaa n konkurensi, tergantung pada kebutuhan Anda. Untuk informasi selengkapnya, lihat Menerapkan manajemen beban kerja di dokumentasi Amazon Redshift.	Insinyur data

Buat rahasia di Secrets Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat rahasia baru untuk menyimpan kredenal masuk Amazon Redshift di Secrets Manager.	Rahasia ini menyimpan kredensil untuk pengguna admin serta pengguna layanan database individu. Untuk petunjuk, lihat dokumentasi Secrets Manager. Pilih Amazon Redshift Cluster sebagai tipe rahasia. Selain itu, pada halaman rotasi Rahasia, nyalakan rotasi. Ini akan membuat pengguna yang sesuai di cluster Amazon Redshift dan akan memutar rahasia kunci pada interval yang ditentukan.	Insinyur data
Buat kebijakan IAM untuk membatasi akses Secrets Manager.	Batasi akses Secrets Manager hanya ke administrator Amazon Redshift dan AWS Glue.	Insinyur data

Konfigurasikan AWS Glue

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Di AWS Glue Data Catalog, tambahkan koneksi untuk Amazon Redshift.	Untuk petunjuk, lihat dokumentasi AWS Glue.	Insinyur data
Buat dan lampirkan peran layanan IAM untuk AWS Glue untuk mengakses Secrets	Untuk informasi selengkapnya, lihat dokumentasi AWS Glue.	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Manager, Amazon Redshift, dan bucket S3.		
Tentukan AWS Glue Data Catalog untuk sumbernya.	Langkah ini melibatkan pembuatan database dan tabel yang diperlukan di AWS Glue Data Catalog. Anda dapat menggunakan crawler untuk membuat katalog tabel di database AWS Glue, atau mendefinisikannya sebagai tabel eksternal Amazon Athena. Anda juga dapat mengakses tabel eksternal yang ditentukan di Athena melalui AWS Glue Data Catalog. Lihat dokumenta si AWS untuk informasi selengkapnya tentang mendefinisikan Katalog Data dan membuat tabel eksternal di Athena.	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pekerjaan AWS Glue untuk memproses data sumber.	Pekerjaan AWS Glue dapat berupa shell Python atau PySpark untuk menstanda risasi, menghapus duplikasi , dan membersihkan file data sumber. Untuk mengoptim alkan kinerja dan menghinda ri keharusan menanyakan seluruh bucket sumber S3, partisi bucket S3 berdasark an tanggal, dipecah berdasark an tahun, bulan, hari, dan jam sebagai predikat pushdown untuk pekerjaan AWS Glue. Untuk informasi selengkap nya, lihat dokumentasi AWS Glue. Muat data yang diproses dan diubah ke partisi bucket S3 yang diproses dalam format Parket. Anda dapat menanyakan file Parket dari Athena.	Insinyur data
Buat pekerjaan AWS Glue untuk memuat data ke Amazon Redshift.	Pekerjaan AWS Glue dapat berupa shell Python atau PySpark memuat data dengan meningkatkan data, diikuti dengan penyegaran lengkap. Untuk detailnya, lihat dokumentasi AWS Glue dan bagian Informasi tambahan.	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Jadwalkan pekerjaan AWS Glue dengan menggunakan pemicu seperlunya.	Pemuatan data tambahan terutama didorong oleh peristiwa Amazon S3 yang menyebabkan fungsi AWS Lambda memanggil pekerjaan AWS Glue. Gunakan penjadwalan berbasis pemicu AWS Glue untuk setiap pemuatan data yang menuntut berbasis waktu, bukan penjadwalan berbasis peristiwa.	Insinyur data

Buat fungsi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan lampirkan peran terkait layanan IAM untuk AWS Lambda untuk mengakses bucket S3 dan pekerjaan AWS Glue.	Buat peran terkait layanan IAM untuk AWS Lambda dengan kebijakan untuk membaca objek dan bucket Amazon S3, serta kebijakan untuk mengakses AWS Glue API untuk memulai pekerjaan AWS Glue. Untuk informasi lebih lanjut, lihat Pusat Pengetahuan.	Insinyur data
Buat fungsi Lambda untuk menjalankan pekerjaan AWS Glue berdasarkan peristiwa Amazon S3 yang ditentukan.	Fungsi Lambda harus dimulai dengan pembuatan file manifes Amazon S3. Fungsi Lambda harus meneruska n lokasi folder Amazon S3 (misalnya, bucket/year/	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	month/date/hour source_) ke pekerjaan AWS Glue sebagai parameter. Pekerjaan AWS Glue akan menggunakan parameter ini sebagai predikat pushdown untuk mengoptim alkan akses file dan kinerja pemrosesan pekerjaan. Untuk informasi selengkapnya, lihat dokumentasi AWS Glue.	
Buat event objek PUT Amazon S3 untuk mendeteksi pembuatan objek, dan panggil fungsi Lambda masing-ma sing.	Peristiwa objek PUT Amazon S3 harus dimulai hanya dengan pembuatan file manifes. File manifes mengontrol fungsi Lambda dan konkurensi pekerjaan AWS Glue, dan memproses beban sebagai batch alih-alih memproses file individua l yang tiba di partisi spesifik bucket sumber S3. Untuk informasi selengkapnya, lihat dokumentasi Lambda.	Insinyur data

Sumber daya terkait

- Dokumentasi Amazon S3
- Dokumentasi AWS Glue
- Dokumentasi Amazon Redshift
- AWS Lambda
- Amazon Athena
- AWS Secrets Manager

Informasi tambahan

Pendekatan terperinci untuk upsert dan refresh lengkap

Upsert: Ini untuk kumpulan data yang memerlukan agregasi historis, tergantung pada kasus penggunaan bisnis. Ikuti salah satu pendekatan yang dijelaskan dalam Memperbarui dan menyisipkan data baru (dokumentasi Amazon Redshift) berdasarkan kebutuhan bisnis Anda.

Penyegaran lengkap: Ini untuk kumpulan data kecil yang tidak memerlukan agregasi historis. Ikuti salah satu pendekatan ini:

1. Memangkas tabel Amazon Redshift.
2. Muat partisi saat ini dari area pementasan

atau:

1. Buat tabel sementara dengan data partisi saat ini.
2. Jatuhkan tabel Amazon Redshift target.
3. Ganti nama tabel sementara menjadi tabel target.

Membangun mesh data perusahaan dengan Amazon DataZone, AWS CDK, dan AWS CloudFormation

Dhrubajyoti Mukherjey, Ravi Kumar, Weizhou Sun, dan Adjoa Taylor, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS), pelanggan memahami bahwa data adalah kunci untuk mempercepat inovasi dan mendorong nilai bisnis bagi perusahaan mereka. Untuk mengelola data besar ini, Anda dapat mengadopsi arsitektur terdesentralisasi seperti data mesh. Arsitektur data mesh memfasilitasi pemikiran produk, pola pikir yang memperhitungkan pelanggan, tujuan, dan pasar. Data mesh juga membantu membangun model tata kelola federasi yang menyediakan akses cepat dan aman ke data Anda.

Strategi untuk membangun solusi perusahaan berbasis data mesh AWS membahas bagaimana Anda dapat menggunakan Kerangka Strategi Data Mesh untuk merumuskan dan menerapkan strategi data mesh untuk organisasi Anda. Dengan menggunakan Kerangka Strategi Data Mesh, Anda dapat mengoptimalkan organisasi tim dan interaksinya untuk mempercepat perjalanan data mesh Anda.

Dokumen ini memberikan panduan tentang cara membangun mesh data perusahaan dengan Amazon DataZone. Amazon DataZone adalah layanan manajemen data untuk membuat katalog, menemukan, berbagi, dan mengatur data yang disimpan di seluruh, di tempat AWS, dan sumber pihak ketiga. Pola ini mencakup artefak kode yang membantu Anda menyebarkan infrastruktur solusi data berbasis mesh data menggunakan dan. AWS Cloud Development Kit (AWS CDK) AWS CloudFormation Pola ini ditujukan untuk arsitek dan DevOps insinyur cloud.

Untuk informasi tentang tujuan pola ini dan ruang lingkup solusi, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Minimal dua aktif Akun AWS: satu untuk akun tata kelola pusat dan satu lagi untuk akun anggota
- AWS kredensi administrator untuk akun tata kelola pusat di lingkungan pengembangan Anda
- AWS Command Line Interface (AWS CLI) diinstal untuk mengelola Anda Layanan AWS dari baris perintah
- Node.js dan Node Package Manager (npm) diinstal untuk mengelola aplikasi AWS CDK

- AWS CDK Toolkit diinstal secara global di lingkungan pengembangan Anda dengan menggunakan npm, untuk mensintesis dan menyebarkan aplikasi AWS CDK

```
npm install -g aws-cdk
```

- Python versi 3.12 diinstal di lingkungan pengembangan Anda
- TypeScript diinstal di lingkungan pengembangan Anda atau diinstal secara global dengan menggunakan kompiler npm:

```
npm install -g typescript
```

- Docker diinstal di lingkungan pengembangan Anda
- Sistem kontrol versi seperti Git untuk mempertahankan kode sumber solusi (disarankan)
- Lingkungan pengembangan terintegrasi (IDE) atau editor teks dengan dukungan untuk Python dan TypeScript (sangat disarankan)

Batasan

- Solusinya telah diuji hanya pada mesin yang menjalankan Linux atau macOS.
- Dalam versi saat ini, solusinya tidak mendukung integrasi Amazon DataZone dan secara AWS IAM Identity Center default. Namun, Anda dapat mengonfigurasinya untuk mendukung integrasi ini.

Versi produk

- Python versi 3.12

Arsitektur

Diagram berikut menunjukkan arsitektur referensi data mesh. Arsitekturnya didasarkan pada Amazon DataZone dan menggunakan Amazon Simple Storage Service (Amazon S3) AWS Glue Data Catalog dan sebagai sumber data. Layanan AWS Yang Anda gunakan dengan Amazon DataZone dalam implementasi mesh data Anda mungkin berbeda, berdasarkan persyaratan organisasi Anda.

1. Dalam akun produsen, data mentah cocok untuk konsumsi dalam bentuk saat ini atau diubah untuk konsumsi dengan menggunakan AWS Glue. Metadata teknis untuk data disimpan di Amazon S3 dan dievaluasi menggunakan perayap data. AWS Glue Kualitas data diukur dengan

menggunakan Kualitas AWS Glue Data. Database sumber dalam Katalog Data terdaftar sebagai aset dalam DataZone katalog Amazon. DataZone Katalog Amazon di-host di akun tata kelola pusat menggunakan pekerjaan sumber DataZone data Amazon.

2. Akun tata kelola pusat menampung DataZone domain Amazon dan portal DataZone data Amazon. Produsen data dan konsumen terkait dengan DataZone domain Amazon. Akun AWS DataZone Proyek Amazon dari produsen data dan konsumen diatur di bawah unit DataZone domain Amazon yang sesuai.
3. Pengguna akhir aset data masuk ke portal DataZone data Amazon dengan menggunakan kredensialnya AWS Identity and Access Management (IAM) atau sistem masuk tunggal (dengan integrasi melalui Pusat Identitas IAM). Mereka mencari, memfilter, dan melihat informasi aset (misalnya, informasi kualitas data atau metadata bisnis dan teknis) di katalog DataZone data Amazon.
4. Setelah pengguna akhir menemukan aset data yang mereka inginkan, mereka menggunakan fitur DataZone langganan Amazon untuk meminta akses. Pemilik data di tim produsen menerima pemberitahuan dan mengevaluasi permintaan berlangganan di portal DataZone data Amazon. Pemilik data menyetujui atau menolak permintaan berlangganan berdasarkan validitasnya.
5. Setelah permintaan berlangganan dikabulkan dan dipenuhi, aset diakses di akun konsumen untuk kegiatan berikut:
   - Pengembangan model AI/ML dengan menggunakan Amazon AI SageMaker
   - Analisis dan pelaporan menggunakan Amazon Athena dan Amazon Quick Suite

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon Simple Storage Service (Amazon S3) dengan menggunakan SQL standar.
- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- Amazon DataZone adalah layanan manajemen data yang membantu Anda membuat katalog, menemukan, berbagi, dan mengatur data yang disimpan di seluruh AWS, di tempat, dan di sumber pihak ketiga.

- Amazon Quick Suite adalah layanan intelijen bisnis skala cloud (BI) yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda dalam satu dasbor.
- Amazon SageMaker AI adalah layanan pembelajaran mesin terkelola (ML) yang membantu Anda membangun dan melatih model ML, lalu menerapkannya ke lingkungan host yang siap produksi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Solusinya tersedia di repositori GitHub data-mesh-datazone-cdk-cloudformation.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori, jalankan perintah berikut di lingkungan pengembangan lokal Anda (Linux atau macOS): `git clone https://github.com/aws-samples/data-mesh-datazone-cdk-cloudformation`	Arsitek awan, DevOps insinyur
Buat lingkungan.	Untuk membuat lingkungan virtual Python, jalankan perintah berikut:	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`python3 -m venv .venv`
	`source .venv/bin/activate`
	`pip install -r requirements.txt`	

Bootstrap akun.	Untuk mem-bootstrap akun tata kelola pusat dengan menggunakan AWS CDK, jalankan perintah berikut: `cdk bootstrap aws://<GOVERNANCE_ACCOUNT_ID>/<AWS_REGION>` Masuk ke Konsol Manajemen AWS, buka konsol akun tata kelola pusat, dan dapatkan Nama Sumber Daya Amazon (ARN) dari peran eksekusi AWS CDK .	Arsitek awan, DevOps insinyur
Membangun `DzDataMeshMemberStackSet.yaml` file.	Untuk membangun `DzDataMeshMemberStackSet.yaml` file, dari direktori root repositori, memulai skrip bash berikut: `./lib/scripts/create_dz_data_mesh_member_stack_set.sh`	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan pembuatan template.	Pastikan bahwa file CloudFormation template dibuat di `lib/cfn-templates/DzDataMeshMemberStackSet.yaml` lokasi.	Arsitek awan, DevOps insinyur

Menyebarkan sumber daya di akun tata kelola pusat

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ubah konfigurasi.	Dalam `config/Config.ts` file, ubah parameter berikut:
	`DZ_APPLICATION_NAME - Name of the application.`
	`DZ_STAGE_NAME - Name of the stage.`
	`DZ_DOMAIN_NAME - Name of the Amazon DataZone domain`
	`DZ_DOMAIN_DESCRIPTION - Description of the Amazon DataZone domain`
	`DZ_DOMAIN_TAG - Tag of the Amazon DataZone domain`
	`DZ_ADMIN_PROJECT_NAME - Name of the Amazon DataZone project for administrators`
	`DZ_ADMIN_PROJECT_DESCRIPTION - Description of the Amazon DataZone project for administrators`
		Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`CDK_EXEC_ROLE_ARN - ARN of the cdk execution role DZ_ADMIN_ROLE_ARN - ARN of the administrator role` Biarkan parameter yang tersisa kosong.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui konfigurasi DataZone glosarium Amazon.	Untuk memperbarui konfigurasi DataZone glosarium Amazon dalam `lib/utils/glossary_config.json` file, gunakan konfigurasi contoh berikut: `{ "GlossaryName": "PII Data", "GlossaryDescription": "If data source contains PII attributes", "GlossaryTerms": [{ "Name": "Yes", "ShortDescription": "Yes", "LongDescription": "Yes Glossary Term" }, { "Name": "No", "ShortDescription": "No", "LongDescription": "No Glossary Term" } ] }`	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui konfigurasi formulir DataZone metadata Amazon.	Untuk memperbarui konfigurasi formulir DataZone metadata Amazon di`lib/utils/metadata_form_config.json file` , gunakan konfigurasi contoh berikut: `{ "FormName": "ScheduleDataRefresh", "FormDescription": "Form for data refresh schedule", "FormSmithyModel": "@amazon.datazone#displayname(defaultName: \"Data Refresh Schedule\")\nstructure ScheduleDataRefresh {\n @documentation(\"Schedule of Data Refresh\")\n @required\n @amazon.datazone#searchable\n @amazon.datazone#displayname(defaultName: \"Data Refresh Schedule\")\n data_refresh_schedule: String\n}" }`	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ekspor AWS kredensialnya.	Untuk mengekspor AWS kredensyal ke lingkungan pengembangan Anda untuk peran IAM dengan izin administratif, gunakan format berikut:
	`export AWS_ACCESS_KEY_ID=`
	`export AWS_SECRET_ACCESS_KEY=`
	`export AWS_SESSION_TOKEN=`
		Arsitek awan, DevOps insinyur

Sintesis template.	Untuk mensintesis CloudFormation template, jalankan perintah berikut: `npx cdk synth`	Arsitek awan, DevOps insinyur
Menyebarkan solusinya.	Untuk menerapkan solusi, jalankan perintah berikut: `npx cdk deploy --all`	Arsitek awan, DevOps insinyur

Konfigurasikan akun anggota baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan template.	Terapkan CloudFormation template yang terletak `lib/cfn-templates/DzDataMeshCfnStackSetExecutionRole.y`	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aml` di akun anggota dengan parameter input berikut: • `GovernanceAccountID` - ID Akun dari akun tata kelola • `DataZoneKMSKeyID` ID kunci AWS Key Management Service (AWS KMS) yang mengenkripsi metadata Amazon DataZone • `NotificationQueueName` - Nama antrian notifikasi Amazon SQS di akun tata kelola	
Perbarui ARNs.	Untuk memperbarui daftar peran CloudFormation StackSet ARNs eksekusi akun anggota, gunakan kode berikut: `DZ_MEMBER_STACK_SET_EXEC_ROLE_LIST - List of Stack set execution role arns for the member accounts.`	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Sintesis dan terapkan.	Untuk mensintesis CloudFormation template dan menerapkan solusi, jalankan perintah berikut:
	`npx cdk synth`
	`npx cdk deploy --all`
		Arsitek awan, DevOps insinyur

Kaitkan akun anggota.	Untuk mengaitkan akun anggota dengan akun tata kelola pusat, lakukan hal berikut: 1. Masuk ke konsol akun tata kelola pusat, dan buka DataZone konsol Amazon di https://console.aws.amazon.com/datazone/. 2. Pilih domain yang Anda buat. 3. Gulir ke tab Akun terkait, dan pilih Permintaan Asosiasi. 4. Berikan ID akun AWS, dan pilih AWSRAMPermissionDataZonePortalReadWritesebagai kebijakan RAM. 5. Pilih Asosiasi Permintaan. 6. Tunggu hingga Anda menerima pemberitahuan email bahwa akun Anda berhasil di-bootstrap.	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui parameter.	Untuk memperbarui parameter spesifik akun anggota dalam file konfigurasi di`config/Config.ts` , gunakan format berikut: `export const DZ_MEMBER_ACCOUNT_CONFIG: memberAccountConfig = { '123456789012' : { PROJECT_NAME: 'TEST-PROJECT-123456789012', PROJECT_DESCRIPTION: 'TEST-PROJECT-123456789012', PROJECT_EMAIL: 'user@xyz.com' } }`	Arsitek awan, DevOps insinyur

Sintesis dan gunakan template.	Untuk mensintesis CloudFormation template dan menerapkan solusi, jalankan perintah berikut:
	`npx cdk synth`
	`npx cdk deploy --all`
		Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan akun anggota.	Untuk membuat dan mengonfigurasi akun anggota tambahan dalam solusi data, ulangi langkah sebelumnya untuk setiap akun anggota. Solusi ini tidak membedakan antara produsen data dan konsumen.	Arsitek awan, DevOps insinyur

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Putuskan hubungan akun anggota.	Untuk memisahkan akun, lakukan hal berikut: 1. Masuk ke konsol dan buka DataZone konsol Amazon. 2. Pilih Lihat Domain. 3. Pilih domain yang Anda buat. 4. Pilih tab Asosiasi akun. 5. Pilih akun anggota yang ingin Anda pisahkan. 6. Pilih Putuskan, dan masukkan `disassociate` untuk mengonfirmasi. 7. Ulangi langkah 3-6 untuk semua akun anggota.	Arsitek awan, DevOps insinyur
Hapus instance tumpukan.	Untuk menghapus instance CloudFormation tumpukan, lakukan hal berikut:	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation/. 2. Di panel navigasi, pilih StackSets. 3. Pilih set tumpukan bernama StackSet- DataZone - DataMesh -Member, dan pilih tab Stack instance. 4. Salin Akun AWS ID akun anggota yang ingin Anda hapus dari keanggotaan. 5. Pilih Tindakan, pilih Hapus tumpukan dari StackSet, dan pertahankan opsi default. 6. Di bidang Nomor akun, masukkan ID akun. 7. Dalam daftar tarik-turun Tentukan wilayah, pilih. Wilayah AWS 8. Pilih Berikutnya, lalu pilih Kirim. 9. Pada tab Operasi, konfirmasikan bahwa operasi telah berhasil. Penghapusan tumpukan mungkin memakan waktu. 10. Ulangi langkah 2-9 untuk semua akun anggota.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hancurkan semua sumber daya.	Untuk menghancurkan sumber daya, terapkan langkah-l angkah berikut di lingkunga n pengembangan lokal Anda (Linux atau macOS): 1. Arahkan ke direktori root repositori Anda. 2. Ekspor AWS kredensyal untuk peran IAM yang Anda gunakan untuk membuat tumpukan. AWS CDK 3. Untuk menghancurkan sumber daya cloud, jalankan perintah berikut: `npx cdk destroy --all`	Arsitek awan, DevOps insinyur

Sumber daya terkait

- Amazon DataZone mulai cepat dengan data AWS Glue
- Tutorial: Buat AWS CDK aplikasi pertama Anda
- Memulai dengan CloudFormation
- Strategi untuk membangun solusi perusahaan berbasis data mesh AWS

Informasi tambahan

Tujuan

Menerapkan pola ini mencapai hal berikut:

- Kepemilikan data yang terdesentralisasi - Alihkan kepemilikan data dari tim pusat ke tim yang mewakili sistem sumber, unit bisnis, atau kasus penggunaan organisasi Anda.

- Pemikiran produk — Memperkenalkan pola pikir berbasis produk yang mencakup pelanggan, pasar, dan faktor lain ketika mempertimbangkan aset data dalam organisasi Anda.
- Tata kelola federasi - Meningkatkan pagar pembatas keamanan, kontrol, dan kepatuhan di seluruh produk data organisasi Anda.
- Dukungan multi-akun dan multi-proyek - Mendukung berbagi data dan kolaborasi yang efisien dan aman di seluruh unit bisnis atau proyek organisasi Anda.
- Pemantauan dan pemberitahuan terpusat - Pantau sumber daya cloud mesh data Anda dengan menggunakan Amazon CloudWatch, dan beri tahu pengguna saat akun anggota baru dikaitkan.
- Skalabilitas dan ekstensibilitas - Tambahkan kasus penggunaan baru ke dalam mesh data saat organisasi Anda berkembang.

Lingkup solusi

Ketika Anda menggunakan solusi ini, Anda dapat memulai dari yang kecil dan skala saat Anda maju dalam perjalanan data mesh Anda. Seringkali, ketika akun anggota mengadopsi solusi data, itu berisi konfigurasi akun khusus untuk organisasi, proyek, atau unit bisnis. Solusi ini mengakomodasi beragam Akun AWS konfigurasi ini dengan mendukung fitur-fitur berikut:

- AWS Glue Data Catalog sebagai sumber data untuk Amazon DataZone
- Pengelolaan domain DataZone data Amazon dan portal data terkait
- Manajemen penambahan akun anggota dalam solusi data berbasis data mesh data
- Pengelolaan DataZone proyek dan lingkungan Amazon
- Pengelolaan DataZone glosarium Amazon dan formulir metadata
- Pengelolaan peran IAM yang sesuai dengan pengguna solusi data berbasis jaringan data
- Pemberitahuan pengguna solusi data berbasis data mesh
- Pemantauan infrastruktur cloud yang disediakan

  Solusi ini menggunakan AWS CDK dan CloudFormation menyebarkan infrastruktur cloud. Ini digunakan CloudFormation untuk melakukan hal berikut:

  - Tentukan dan terapkan sumber daya cloud pada tingkat abstraksi yang lebih rendah.
  - Menyebarkan sumber daya cloud dari. Konsol Manajemen AWS Dengan menggunakan pendekatan ini, Anda dapat menerapkan infrastruktur tanpa lingkungan pengembangan.

Solusi data mesh digunakan AWS CDK untuk mendefinisikan sumber daya pada tingkat abstraksi yang lebih tinggi. Hasilnya, solusi ini menyediakan pendekatan terpisah, modular, dan terukur dengan memilih alat yang relevan untuk menyebarkan sumber daya cloud.

Langkah selanjutnya

Anda dapat menghubungi AWSpara ahli untuk panduan membangun data mesh dengan Amazon DataZone.

Sifat modular dari solusi ini mendukung membangun solusi manajemen data dengan arsitektur yang berbeda, seperti data fabric dan data lake. Selain itu, berdasarkan persyaratan organisasi Anda, Anda dapat memperluas solusi ke sumber DataZone data Amazon lainnya.

Hitung value at risk (VaR) dengan menggunakan layanan AWS

Sumon Samanta, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerapkan sistem perhitungan value at risk (VaR) dengan menggunakan layanan AWS. Di lingkungan lokal, sebagian besar sistem VaR menggunakan infrastruktur khusus yang besar dan perangkat lunak penjadwalan jaringan internal atau komersial untuk menjalankan proses batch. Pola ini menyajikan arsitektur yang sederhana, andal, dan dapat diskalakan untuk menangani pemrosesan VaR di AWS Cloud. Ini membangun arsitektur tanpa server yang menggunakan Amazon Kinesis Data Streams sebagai layanan streaming, Amazon Simple Queue Service (Amazon SQS) sebagai layanan antrian terkelola, Amazon sebagai layanan caching, dan AWS Lambda untuk ElastiCache memproses pesanan dan menghitung risiko.

VaR adalah ukuran statistik yang digunakan pedagang dan manajer risiko untuk memperkirakan potensi kerugian dalam portofolio mereka di luar tingkat kepercayaan tertentu. Sebagian besar sistem VaR melibatkan menjalankan sejumlah besar perhitungan matematika dan statistik dan menyimpan hasilnya. Perhitungan ini membutuhkan sumber daya komputasi yang signifikan, sehingga proses batch VaR harus dipecah menjadi set tugas komputasi yang lebih kecil. Memisahkan batch besar menjadi tugas yang lebih kecil dimungkinkan karena tugas-tugas ini sebagian besar independen (yaitu, perhitungan untuk satu tugas tidak bergantung pada tugas lain).

Persyaratan penting lainnya untuk arsitektur VaR adalah skalabilitas komputasi. Pola ini menggunakan arsitektur tanpa server yang secara otomatis menskalakan masuk atau keluar berdasarkan beban komputasi. Karena permintaan komputasi batch atau online sulit diprediksi, penskalaan dinamis diperlukan untuk menyelesaikan proses dalam garis waktu yang ditentukan oleh perjanjian tingkat layanan (SLA). Selain itu, arsitektur yang dioptimalkan biaya harus dapat menurunkan setiap sumber daya komputasi segera setelah tugas pada sumber daya tersebut selesai.

Layanan AWS sangat cocok untuk perhitungan VaR karena menawarkan kapasitas komputasi dan penyimpanan yang dapat diskalakan, layanan analitik untuk diproses dengan cara yang dioptimalkan biaya, dan berbagai jenis penjadwal untuk menjalankan alur kerja manajemen risiko Anda. Selain itu, Anda hanya membayar untuk sumber daya komputasi dan penyimpanan yang Anda gunakan di AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Masukkan file, yang tergantung pada kebutuhan bisnis Anda. Kasus penggunaan tipikal melibatkan file input berikut:
  - File data pasar (masukan ke mesin perhitungan VaR)
  - File data perdagangan (kecuali data perdagangan datang melalui aliran).
  - File data konfigurasi (model dan data konfigurasi statis lainnya)
  - File model mesin perhitungan (pustaka kuantitatif)
  - File data deret waktu (untuk data historis seperti harga saham selama lima tahun terakhir)
- Jika data pasar atau input lain masuk melalui aliran, Amazon Kinesis Data Streams disiapkan, dan Amazon Identity and Access Management (IAM) and Access Management (IAM) dikonfigurasi untuk menulis ke aliran.

Pola ini membangun arsitektur di mana data perdagangan ditulis dari sistem perdagangan ke aliran data Kinesis. Alih-alih menggunakan layanan streaming, Anda dapat menyimpan data perdagangan Anda dalam file batch kecil, menyimpannya di bucket Amazon Simple Storage Service (Amazon S3), dan memanggil acara untuk mulai memproses data.

Batasan

- Pengurutan aliran data Kinesis dijamin pada setiap pecahan, sehingga pesanan perdagangan yang ditulis ke beberapa pecahan tidak dijamin akan dikirimkan dalam urutan yang sama dengan operasi penulisan.
- Batas runtime AWS Lambda saat ini adalah 15 menit. (Untuk informasi lebih lanjut, lihat FAQ Lambda.)

Arsitektur

Arsitektur target

Diagram arsitektur berikut menampilkan layanan AWS dan alur kerja untuk sistem penilaian risiko.

Diagram ini menggambarkan hal sebagai berikut:

1. Aliran perdagangan dari sistem manajemen pesanan.
2. Fungsi Lambda jaring posisi tiket memproses pesanan dan menulis pesan terkonsolidasi untuk setiap ticker ke antrian risiko di Amazon SQS.
3. Mesin penghitung risiko Fungsi Lambda memproses pesan dari Amazon SQS, melakukan perhitungan risiko, dan memperbarui informasi laba rugi (PnL) VaR di cache risiko di Amazon. ElastiCache
4. Fungsi Lambda ElastiCache data baca mengambil hasil risiko dari ElastiCache dan menyimpannya dalam database dan bucket S3.

Untuk informasi selengkapnya tentang layanan dan langkah-langkah ini, lihat bagian Epik.

Otomatisasi dan skala

Anda dapat menerapkan seluruh arsitektur dengan menggunakan AWS Cloud Development Kit (AWS CDK) atau templat CloudFormation AWS. Arsitektur dapat mendukung pemrosesan batch dan pemrosesan intraday (real-time).

Penskalaan dibangun ke dalam arsitektur. Karena lebih banyak perdagangan ditulis ke dalam aliran data Kinesis dan menunggu untuk diproses, fungsi Lambda tambahan dapat dipanggil untuk memproses perdagangan tersebut dan kemudian dapat menurunkan skala setelah pemrosesan selesai. Memproses melalui beberapa antrian perhitungan risiko Amazon SQS juga merupakan pilihan. Jika pemesanan atau konsolidasi yang ketat diperlukan di seluruh antrian, pemrosesan tidak dapat diparalelkan. Namun, untuk end-of-the-day batch atau batch intraday mini, fungsi Lambda dapat memproses secara paralel dan menyimpan hasil akhir. ElastiCache

Alat

Layanan AWS

- Amazon Aurora MySQL Compatible Edition adalah mesin database relasional MySQL yang dikelola sepenuhnya dan kompatibel dengan MySQL yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan MySQL. Pola ini menggunakan MySQL sebagai contoh, tetapi Anda dapat menggunakan sistem RDBMS apa pun untuk menyimpan data.
- Amazon ElastiCache membantu Anda mengatur, mengelola, dan menskalakan lingkungan cache dalam memori terdistribusi di AWS Cloud.
- Amazon Kinesis Data Streams membantu Anda mengumpulkan dan memproses aliran besar catatan data secara real time.

- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Kode

Pola ini memberikan contoh arsitektur untuk sistem VaR di AWS Cloud dan menjelaskan bagaimana Anda dapat menggunakan fungsi Lambda untuk perhitungan VaR. Untuk membuat fungsi Lambda Anda, lihat contoh kode dalam dokumentasi Lambda. Untuk bantuan, hubungi AWS Professional Services.

Praktik terbaik

- Pertahankan setiap tugas komputasi VaR sekecil dan seringan mungkin. Bereksperimenlah dengan jumlah perdagangan yang berbeda di setiap tugas komputasi untuk melihat mana yang paling dioptimalkan untuk waktu dan biaya komputasi.
- Simpan objek yang dapat digunakan kembali di Amazon ElastiCache. Gunakan kerangka kerja seperti Apache Arrow untuk mengurangi serialisasi dan deserialisasi.
- Pertimbangkan batasan waktu Lambda. Jika menurut Anda tugas komputasi Anda mungkin melebihi 15 menit, cobalah untuk memecahnya menjadi tugas yang lebih kecil untuk menghindari batas waktu Lambda. Jika ini tidak memungkinkan, Anda dapat mempertimbangkan solusi orkestrasi kontainer dengan AWS Fargate, Amazon Elastic Container Service (Amazon ECS), dan Amazon Elastic Kubernetes Service (Amazon EKS).

Epik

Aliran perdagangan ke sistem risiko

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai menulis perdagangan.	Perdagangan baru, diselesai kan, atau sebagian diselesai kan ditulis dari sistem manajemen pesanan ke aliran risiko. Pola ini menggunakan Amazon Kinesis sebagai layanan streaming terkelola. Hash ticker order perdagangan digunakan untuk menempatk an pesanan perdagangan di beberapa pecahan.	Amazon Kinesis

Jalankan fungsi Lambda untuk pemrosesan pesanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai pemrosesan risiko dengan Lambda.	Jalankan fungsi AWS Lambda untuk pesanan baru. Berdasarkan jumlah order perdagangan yang tertunda, Lambda akan secara otomatis menskalakan. Setiap instans Lambda memiliki satu atau lebih pesanan dan mengambil posisi terbaru untuk setiap ticker dari Amazon. ElastiCac he (Anda dapat menggunak an ID CUSIP, nama Kurva, atau nama indeks untuk	Amazon Kinesis, AWS Lambda, Amazon ElastiCache

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	produk derivatif keuangan lainnya sebagai kunci untuk menyimpan dan mengambil data dari.) ElasticCache Dalam ElastiCache, total posisi (kuantitas) dan pasangan kunci-nilai < ticker, posisi bersih>, di mana posisi bersih adalah faktor penskalaa n, diperbarui satu kali untuk setiap ticker.	

Menulis pesan untuk setiap ticker ke dalam antrian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tulis pesan konsolidasi ke antrian risiko.	Tulis pesan ke antrian. Pola ini menggunakan Amazon SQS sebagai layanan antrian terkelola. Satu instance Lambda mungkin mendapatk an batch mini pesanan perdagangan pada waktu tertentu, tetapi hanya akan menulis satu pesan untuk setiap ticker ke Amazon SQS. Faktor penskalaan dihitung: (posisi bersih lama+posisi saat ini) /posisi bersih lama.	Amazon SQS, AWS Lambda

Memohon mesin risiko

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai perhitungan risiko.	Fungsi Lambda untuk lambda mesin risiko dipanggil. Setiap posisi diproses oleh satu fungsi Lambda. Namun, untuk tujuan pengoptimalan, setiap fungsi Lambda dapat memproses beberapa pesan dari Amazon SQS.	Amazon SQS, AWS Lambda

Mengambil hasil risiko dari cache

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ambil dan perbarui cache risiko.	Lambda mengambil posisi bersih saat ini untuk setiap ticker dari. ElastiCache Ini juga mengambil array laba dan rugi (PnL) VaR untuk setiap ticker dari. ElastiCache Jika array PnL sudah ada, fungsi Lambda memperbar ui array dan VaR dengan skala, yang berasal dari pesan Amazon SQS yang ditulis oleh fungsi Lambda jaring. Jika array PnL tidak masuk ElasticCache, PnL dan VaR baru dihitung dengan menggunakan data seri harga ticker simulasi.	Amazon SQS, AWS Lambda, Amazon ElastiCache

Perbarui data di Elastic Cache dan simpan dalam database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyimpan hasil risiko.	Setelah nomor VaR dan PnL diperbarui ElastiCache, fungsi Lambda baru dipanggil setiap lima menit. Fungsi ini membaca semua data yang disimpan dari ElastiCache dan menyimpannya dalam database yang kompatibel dengan Aurora MySQL dan dalam ember S3.	AWS Lambda, Amazon ElastiCache

Sumber daya terkait

- Kerangka Basel VaR

Konfigurasikan akses lintas akun ke Katalog Data AWS Glue bersama menggunakan Amazon Athena

Denis Avdonin, Amazon Web Services

Ringkasan

Pola ini memberikan step-by-step petunjuk, termasuk contoh kebijakan AWS Identity and Access Management (IAM), untuk mengonfigurasi berbagi lintas akun dari kumpulan data yang disimpan dalam bucket Amazon Simple Storage Service (Amazon S3) dengan menggunakan AWS Glue Data Catalog. Anda dapat menyimpan dataset dalam bucket S3. Metadata dikumpulkan oleh crawler AWS Glue dan dimasukkan ke dalam AWS Glue Data Catalog. Bucket S3 dan AWS Glue Data Catalog berada di akun AWS yang disebut sebagai akun data. Anda dapat memberikan akses ke prinsipal IAM di akun AWS lain yang disebut sebagai akun konsumen. Pengguna dapat menanyakan data di akun konsumen dengan menggunakan mesin kueri tanpa server Amazon Athena.

Prasyarat dan batasan

Prasyarat

- Dua akun AWS aktif
- Bucket S3 di salah satu akun AWS
- Mesin Athena versi 2
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi (atau AWS untuk menjalankan perintah CloudShell AWS CLI)

Versi produk

Pola ini bekerja dengan mesin Athena versi 2 dan mesin Athena versi 3 saja. Kami menyarankan Anda meningkatkan ke mesin Athena versi 3. Jika Anda tidak dapat meningkatkan dari mesin Athena versi 1 ke mesin Athena versi 3, ikuti pendekatan yang dibahas dalam Akses Katalog Data AWS Glue Lintas akun dengan Amazon Athena di AWS Big Data Blog.

Arsitektur

Tumpukan teknologi target

- Amazon Athena
- Amazon Simple Storage Service (Amazon S3)
- AWS Glue
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)

Diagram berikut menunjukkan arsitektur yang menggunakan izin IAM untuk berbagi data dalam bucket S3 di satu akun AWS (akun data) dengan akun AWS lain (akun konsumen) melalui AWS Glue Data Catalog.

Diagram menunjukkan alur kerja berikut:

1. Kebijakan bucket S3 di akun data memberikan izin ke peran IAM di akun konsumen dan peran layanan crawler AWS Glue di akun data.
2. Kebijakan kunci AWS KMS di akun data memberikan izin untuk peran IAM di akun konsumen dan peran layanan crawler AWS Glue di akun data.
3. Crawler AWS Glue di akun data menemukan skema data yang disimpan di bucket S3.
4. Kebijakan sumber daya Katalog Data AWS Glue di akun data memberikan akses ke peran IAM di akun konsumen.
5. Pengguna membuat referensi katalog bernama di akun konsumen dengan menggunakan perintah AWS CLI.
6. Kebijakan IAM memberikan peran IAM dalam akses akun konsumen ke sumber daya di akun data. Kebijakan kepercayaan peran IAM memungkinkan pengguna di akun konsumen untuk mengambil peran IAM.
7. Seorang pengguna di akun konsumen mengasumsikan peran IAM dan mengakses objek dalam katalog data dengan menggunakan kueri SQL.
8. Mesin tanpa server Athena menjalankan kueri SQL.

> **Note**
>
> Praktik terbaik IAM merekomendasikan agar Anda memberikan izin untuk peran IAM dan menggunakan federasi identitas.

Alat

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk melindungi data Anda.

Epik

Mengatur izin di akun data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan akses ke data di bucket S3.	Buat kebijakan bucket S3 berdasarkan template berikut dan tetapkan kebijakan ke bucket tempat data disimpan. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": [ "arn:aws:iam::<con`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	sumer account id>:role/
	<role name>",
	
	"arn:aws:iam::<dat
	a account id>:role/
	service-role/AWSGl
	ueServiceRole-data-
	bucket-crawler"
	]
	},
	"Action":
	"s3:GetObject",
	"Resource":
	"arn:aws:s3:::data-
	bucket/*"
	},
	{
	"Effect":
	"Allow",
	"Principa
	l": {
	"AWS": [
	
	"arn:aws:iam::<con
	sumer account id>:role/
	<role name>",
	
	"arn:aws:iam::<dat
	a account id>:role/
	service-role/AWSGl
	ueServiceRole-data-
	bucket-crawler"
	]
	},
	"Action":
	"s3:ListBucket",
	"Resource":
	"arn:aws:s3:::data-
	bucket"
	}
	]	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>}</pre> Kebijakan bucket memberikan izin untuk peran IAM di akun konsumen dan peran layanan crawler AWS Glue di akun data.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Jika diperlukan) Berikan akses ke kunci enkripsi data.	Jika bucket S3 dienkripsi oleh kunci AWS KMS, berikan `kms:Decrypt` izin pada kunci peran IAM di akun konsumen dan peran layanan crawler AWS Glue di akun data. Perbarui kebijakan kunci dengan pernyataan berikut: `{ "Effect": "Allow", "Principal": { "AWS": [ "arn:aws:iam::<consumer account id>:role/<role name>", "arn:aws:iam::<data account id>:role/service-role/AWSGlueServiceRole-data-bucket-crawler" ] }, "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" }`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan akses crawler ke data.	Lampirkan kebijakan IAM berikut ke peran layanan crawler: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data-bucket/*" }, { "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::data-bucket" } ] }`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Jika diperlukan) Berikan akses crawler ke kunci enkripsi data.	Jika bucket S3 dienkripsi oleh kunci AWS KMS, berikan `kms:Decrypt` izin pada kunci tersebut ke peran layanan crawler dengan melampirkan kebijakan berikut padanya: `{ "Effect": "Allow", "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" }`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan peran IAM di akun konsumen dan akses crawler ke katalog data.	1. Masuk ke AWS Management Console dan buka konsol AWS Glue. 2. Di panel navigasi, di bawah Katalog Data, pilih Pengaturan. 3. Di bagian Izin, tambahkan pernyataan berikut, lalu pilih Simpan. `{ "Version" : "2012-10-17", "Statement" : [ { "Effect" : "Allow", "Principal" : { "AWS" : [ "arn:aws:iam::<consumer account id>:role/<role name>", "arn:aws:iam::<data account id>:role/service-role/AWSGlueServiceRole-data-bucket-crawler" ] }, "Action" : "glue:*", "Resource" : [`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"arn:aws:glue:<region>:<data account id>:catalog",
	
	"arn:aws:glue:<region>:<data account id>:database/*",
	
	"arn:aws:glue:<region>:<data account id>:table/*"
	]
	}
	]
	}</pre>
	Kebijakan ini memungkinkan semua tindakan AWS Glue pada semua database dan tabel di akun data. Anda dapat menyesuaikan kebijakan agar hanya memberikan izin yang diperlukan kepada prinsipal konsumen. Misalnya, Anda dapat memberikan akses hanya-baca ke tabel atau tampilan tertentu dalam database.	

Akses data dari akun konsumen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat referensi bernama untuk katalog data.	Untuk membuat referensi katalog data bernama, gunakan CloudShellatau AWS	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CLI yang diinstal secara lokal untuk menjalankan perintah berikut: `aws athena create-data-catalog --name <shared catalog name> --type GLUE --parameters catalog-id=<data account id>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan peran IAM dalam akses akun konsumen ke data.	Lampirkan kebijakan berikut ke peran IAM di akun konsumen untuk memberikan akses lintas akun peran ke data: `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data-bucket/*" }, { "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::data-bucket" }, { "Effect": "Allow", "Action": "glue:*", "Resource": ["arn:aws:glue:<region>:<data account id>:catalog",`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"arn:aws:glue:<region>:<data account id>:database/*",
	
	"arn:aws:glue:<region>:<data account id>:table/*"
	]
	}
	]
	}</pre>
	Selanjutnya, gunakan templat berikut untuk menentukan apa yang pengguna dapat menerima peran IAM dalam kebijakan kepercayaannya:
	
	<pre>{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Principal": {
	"AWS": "arn:aws:iam::<consumer account id>:user/<IAM user>"
	},
	"Action": "sts:AssumeRole"
	}
	]
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Terakhir, berikan izin pengguna untuk mengambil peran IAM dengan melampirkan kebijakan yang sama ke grup pengguna tempat mereka berada.	
(Jika diperlukan) Berikan peran IAM dalam akses akun konsumen ke kunci enkripsi data.	Jika bucket S3 dienkripsi oleh kunci AWS KMS, berikan `kms:Decrypt` izin pada kunci peran IAM di akun konsumen dengan melampirkan kebijakan berikut padanya: `{ "Effect": "Allow", "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" }`	Administrator awan
Beralih ke peran IAM di akun konsumen untuk mengakses data.	Sebagai konsumen data, beralihlah ke peran IAM untuk mengakses data di akun data.	Konsumen data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Akses datanya.	Kueri data menggunakan Athena. Misalnya, buka editor kueri Athena dan jalankan kueri berikut:
	`SELECT * FROM <shared catalog name>.<database name>.<table name>`
	Alih-alih menggunakan referensi katalog bernama, Anda juga dapat merujuk ke katalog dengan Nama Sumber Daya Amazon (ARN).
	Note
	Jika Anda menggunakan referensi katalog dinamis dalam kueri atau tampilan, kelilingi referensi dengan tanda kutip ganda yang diloloskan (\”). Contoh:
	`SELECT * FROM \"glue:arn:aws:glue:<region>:<data account id>:catalog\".<database name>.<table name>`
		Konsumen data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkap nya, lihat Akses lintas akun ke katalog data AWS Glue di Panduan Pengguna Amazon Athena.	

Sumber daya terkait

- Akses lintas akun ke katalog data AWS Glue (dokumentasi Athena)
- (AWS CLI) (Referensi create-data-catalog Perintah AWS CLI)
- Akses Katalog Data AWS Glue lintas akun dengan Amazon Athena (AWS Big Data Blog)
- Praktik terbaik keamanan dalam IAM (dokumentasi IAM)

Informasi tambahan

Menggunakan Lake Formation sebagai alternatif untuk berbagi lintas akun

Anda juga dapat menggunakan AWS Lake Formation untuk berbagi akses ke objek katalog AWS Glue di seluruh akun. Lake Formation menyediakan kontrol akses berbutir halus pada tingkat kolom dan baris, kontrol akses berbasis tag, tabel yang diatur untuk transaksi ACID, dan fungsionalitas lainnya. Meskipun Lake Formation terintegrasi dengan baik dengan Athena, itu memang memerlukan konfigurasi tambahan dibandingkan dengan pendekatan IAM-only pola ini. Kami menyarankan Anda mempertimbangkan keputusan untuk menggunakan Lake Formation atau kontrol akses khusus IAM dalam konteks yang lebih luas dari keseluruhan arsitektur solusi Anda. Pertimbangan mencakup layanan lain apa yang terlibat dan bagaimana mereka berintegrasi dengan kedua pendekatan tersebut.

Mengonversi fitur temporal Teradata NORMALIZE ke Amazon Redshift SQL

Po Hong, Amazon Web Services

Ringkasan

NORMALIZE adalah ekstensi Teradata ke standar ANSI SQL. Ketika tabel SQL menyertakan kolom yang memiliki tipe data PERIODE, NORMALIZE menggabungkan nilai yang bertemu atau tumpang tindih di kolom itu, untuk membentuk satu periode yang mengkonsolidasikan beberapa nilai periode individual. Untuk menggunakan NORMALIZE, setidaknya satu kolom dalam daftar SQL SELECT harus dari tipe data PERIODE temporal Teradata. Untuk informasi selengkapnya tentang NORMALIZE, lihat dokumentasi Teradata.

Amazon Redshift tidak mendukung NORMALIZE, tetapi Anda dapat menerapkan fungsi ini dengan menggunakan sintaks SQL asli dan fungsi jendela LAG di Amazon Redshift. Pola ini berfokus pada penggunaan ekstensi Teradata NORMALIZE dengan kondisi ON MEETS OR OVERLAPS, yang merupakan format paling populer. Ini menjelaskan bagaimana fitur ini bekerja di Teradata dan bagaimana hal itu dapat diubah menjadi sintaks SQL asli Amazon Redshift.

Prasyarat dan batasan

Prasyarat

- Pengetahuan dan pengalaman SQL Teradata dasar
- Pengetahuan dan pengalaman Amazon Redshift

Arsitektur

Tumpukan teknologi sumber

- Gudang data Teradata

Tumpukan teknologi target

- Amazon Redshift

Arsitektur target

Untuk arsitektur tingkat tinggi untuk memigrasikan database Teradata ke Amazon Redshift, lihat pola Memigrasikan database Teradata ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT. Migrasi tidak secara otomatis mengonversi frasa Teradata NORMALIZE ke Amazon Redshift SQL. Anda dapat mengonversi ekstensi Teradata ini dengan mengikuti pedoman dalam pola ini.

Alat

Kode

Untuk mengilustrasikan konsep dan fungsionalitas NORMALIZE, pertimbangkan definisi tabel berikut di Teradata:

```
CREATE TABLE systest.project
     (    emp_id        INTEGER,
          project_name  VARCHAR(20),
          dept_id       INTEGER,
          duration      PERIOD(DATE)
     );
```

Jalankan kode SQL berikut untuk memasukkan data sampel ke dalam tabel:

```
BEGIN TRANSACTION;

INSERT INTO systest.project VALUES (10, 'First Phase', 1000,  PERIOD(DATE '2010-01-10',
 DATE '2010-03-20') );
INSERT INTO systest.project VALUES (10, 'First Phase', 2000,  PERIOD(DATE '2010-03-20',
 DATE '2010-07-15') );

INSERT INTO systest.project VALUES (10, 'Second Phase', 2000,  PERIOD(DATE
 '2010-06-15', DATE '2010-08-18') );
INSERT INTO systest.project VALUES (20, 'First Phase', 2000,  PERIOD(DATE '2010-03-10',
 DATE '2010-07-20') );

INSERT INTO systest.project VALUES (20, 'Second Phase', 1000,  PERIOD(DATE
 '2020-05-10', DATE '2020-09-20') );

END TRANSACTION;
```

Hasil:

```
select * from systest.project order by 1,2,3;
```

```
 *** Query completed. 4 rows found. 4 columns returned.
 *** Total elapsed time was 1 second.

     emp_id  project_name              dept_id  duration
-----------  --------------------  -----------  ------------------------
         10  First Phase                  1000  ('10/01/10', '10/03/20')
         10  First Phase                  2000  ('10/03/20', '10/07/15')
         10  Second Phase                 2000  ('10/06/15', '10/08/18')
         20  First Phase                  2000  ('10/03/10', '10/07/20')
         20  Second Phase                 1000  ('20/05/10', '20/09/20')
```

Kasus penggunaan Teradata NORMALIZE

Sekarang tambahkan klausa Teradata NORMALIZE SQL ke pernyataan SELECT:

```
SELECT NORMALIZE ON MEETS OR OVERLAPS emp_id, duration
FROM systest.project
ORDER BY 1,2;
```

Operasi NORMALIZE ini dilakukan pada satu kolom (`emp_id`). Untuk`emp_id=10`, tiga nilai periode yang tumpang tindih dalam durasi menyatu menjadi nilai periode tunggal, sebagai berikut:

```
     emp_id  duration
-----------  ------------------------
         10  ('10/01/10', '10/08/18')
         20  ('10/03/10', '10/07/20')
         20  ('20/05/10', '20/09/20')
```

Pernyataan SELECT berikut melakukan operasi NORMALIZE pada `project_name` dan`dept_id`. Perhatikan bahwa daftar SELECT hanya berisi satu kolom PERIODE,`duration`.

```
SELECT NORMALIZE project_name, dept_id, duration
FROM systest.project;
```

Output:

```
project_name              dept_id  duration
--------------------  -----------  ------------------------
First Phase                  1000  ('10/01/10', '10/03/20')
Second Phase                 1000  ('20/05/10', '20/09/20')
```

```
First Phase                  2000  ('10/03/10', '10/07/20')
Second Phase                 2000  ('10/06/15', '10/08/18')
```

SQL setara Amazon Redshift

Amazon Redshift saat ini tidak mendukung tipe data PERIODE dalam tabel. Sebagai gantinya, Anda perlu membagi bidang data Periode Teradata menjadi dua bagian:`start_date, end_date`, sebagai berikut:

```
CREATE TABLE systest.project
     (    emp_id        INTEGER,
          project_name  VARCHAR(20),
          dept_id       INTEGER,
          start_date  DATE,
          end_date    DATE
     );
```

Masukkan data sampel ke dalam tabel:

```
BEGIN TRANSACTION;

INSERT INTO systest.project VALUES (10, 'First Phase', 1000,  DATE '2010-01-10', DATE
 '2010-03-20' );
INSERT INTO systest.project VALUES (10, 'First Phase', 2000,  DATE '2010-03-20', DATE
 '2010-07-15');

INSERT INTO systest.project VALUES (10, 'Second Phase', 2000,  DATE '2010-06-15', DATE
 '2010-08-18' );
INSERT INTO systest.project VALUES (20, 'First Phase', 2000,  DATE '2010-03-10', DATE
 '2010-07-20' );

INSERT INTO systest.project VALUES (20, 'Second Phase', 1000,  DATE '2020-05-10', DATE
 '2020-09-20' );

END TRANSACTION;
```

Output:

```
 emp_id | project_name | dept_id | start_date |  end_date
--------+--------------+---------+------------+------------
     10 | First Phase  |    1000 | 2010-01-10 | 2010-03-20
     10 | First Phase  |    2000 | 2010-03-20 | 2010-07-15
```

```
     10 | Second Phase |    2000 | 2010-06-15 | 2010-08-18
     20 | First Phase  |    2000 | 2010-03-10 | 2010-07-20
     20 | Second Phase |    1000 | 2020-05-10 | 2020-09-20
(5 rows)
```

Untuk menulis ulang klausa NORMALIZE Teradata, Anda dapat menggunakan fungsi jendela LAG di Amazon Redshift. Fungsi ini mengembalikan nilai-nilai untuk baris pada offset tertentu di atas (sebelum) baris saat ini di partisi.

Anda dapat menggunakan fungsi LAG untuk mengidentifikasi setiap baris yang memulai periode baru dengan menentukan apakah suatu periode bertemu atau tumpang tindih dengan periode sebelumnya (0 jika ya dan 1 jika tidak). Ketika bendera ini diringkas secara kumulatif, ia menyediakan pengidentifikasi grup yang dapat digunakan di klausa Grup By luar untuk sampai pada hasil yang diinginkan di Amazon Redshift.

Berikut contoh pernyataan Amazon Redshift SQL yang menggunakan LAG ():

```
SELECT emp_id, start_date, end_date,
            (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY emp_id ORDER BY
 start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project
ORDER BY 1,2;
```

Output:

```
 emp_id | start_date |  end_date  | groupstartflag
--------+------------+------------+----------------
     10 | 2010-01-10 | 2010-03-20 |              1
     10 | 2010-03-20 | 2010-07-15 |              0
     10 | 2010-06-15 | 2010-08-18 |              0
     20 | 2010-03-10 | 2010-07-20 |              1
     20 | 2020-05-10 | 2020-09-20 |              1
(5 rows)
```

Pernyataan Amazon Redshift SQL berikut hanya dinormalisasi pada kolom: `emp_id`

```
SELECT T2.emp_id, MIN(T2.start_date) as new_start_date, MAX(T2.end_date) as
 new_end_date
FROM
( SELECT T1.*, SUM(GroupStartFlag) OVER (PARTITION BY emp_id ORDER BY start_date ROWS
 UNBOUNDED PRECEDING) As GroupID
```

```
FROM ( SELECT emp_id, start_date, end_date,
            (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY emp_id ORDER BY
 start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project ) T1
) T2
GROUP BY T2.emp_id, T2.GroupID
ORDER BY 1,2;
```

Output:

```
 emp_id | new_start_date | new_end_date
--------+----------------+-------------------------------------
     10 | 2010-01-10     | 2010-08-18
     20 | 2010-03-10     | 2010-07-20
     20 | 2020-05-10     | 2020-09-20
(3 rows)
```

Pernyataan Amazon Redshift SQL berikut dinormalisasi pada kolom dan kolom: `project_name` `dept_id`

```
SELECT T2.project_name, T2.dept_id, MIN(T2.start_date) as new_start_date,
 MAX(T2.end_date) as new_end_date
FROM
( SELECT T1.*, SUM(GroupStartFlag) OVER (PARTITION BY project_name, dept_id ORDER BY
 start_date ROWS UNBOUNDED PRECEDING) As GroupID
FROM ( SELECT project_name, dept_id, start_date, end_date,
            (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY project_name,
 dept_id ORDER BY start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project ) T1
) T2
GROUP BY T2.project_name, T2.dept_id, T2.GroupID
ORDER BY 1,2,3;
```

Output:

```
 project_name | dept_id | new_start_date | new_end_date
--------------+---------+----------------+--------------
 First Phase  |    1000 | 2010-01-10     | 2010-03-20
 First Phase  |    2000 | 2010-03-10     | 2010-07-20
 Second Phase |    1000 | 2020-05-10     | 2020-09-20
```

```
 Second Phase |    2000 | 2010-06-15     | 2010-08-18
(4 rows)
```

Epik

Konversi NORMALIZE ke Amazon Redshift SQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kode SQL Teradata Anda.	Gunakan frasa NORMALIZE sesuai dengan kebutuhan Anda.	Developer SQL
Konversi kode ke Amazon Redshift SQL.	Untuk mengonversi kode Anda, ikuti panduan di bagian “Alat” dari pola ini.	Developer SQL
Jalankan kode di Amazon Redshift.	Buat tabel Anda, muat data ke dalam tabel, dan jalankan kode Anda di Amazon Redshift.	Developer SQL

Sumber daya terkait

Referensi

- Teradata NORMALIZE fitur temporal (dokumentasi Teradata)
- Fungsi jendela LAG (dokumentasi Amazon Redshift)
- Bermigrasi ke Amazon Redshift (situs web AWS)
- Migrasikan database Teradata ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT (AWS Prescriptive Guidance)
- Mengonversi fitur Teradata RESET WHEN ke Amazon Redshift SQL (AWS Prescriptive Guidance)

Alat

- Alat Konversi Skema AWS (AWS SCT)

Mitra

- Mitra Kompetensi Migrasi AWS

Mengonversi fitur Teradata RESET WHEN ke Amazon Redshift SQL

Po Hong, Amazon Web Services

Ringkasan

RESET WHEN adalah fitur Teradata yang digunakan dalam fungsi jendela analitik SQL. Ini adalah perpanjangan dari standar ANSI SQL. RESET WHEN menentukan partisi di mana fungsi jendela SQL beroperasi berdasarkan beberapa kondisi tertentu. Jika kondisi dievaluasi ke TRUE, sub-partisi dinamis baru dibuat di dalam partisi jendela yang ada. Untuk informasi selengkapnya tentang RESET KAPAN, lihat dokumentasi Teradata.

Amazon Redshift tidak mendukung RESET WHEN dalam fungsi jendela SQL. Untuk mengimplementasikan fungsionalitas ini, Anda harus mengonversi RESET WHEN ke sintaks SQL asli di Amazon Redshift, dan menggunakan beberapa fungsi bersarang. Pola ini menunjukkan bagaimana Anda dapat menggunakan fitur Teradata RESET WHEN dan bagaimana Anda dapat mengonversinya ke sintaks Amazon Redshift SQL.

Prasyarat dan batasan

Prasyarat

- Pengetahuan dasar gudang data Teradata dan sintaks SQL-nya
- Pemahaman yang baik tentang Amazon Redshift dan sintaks SQL-nya

Arsitektur

Tumpukan teknologi sumber

- Gudang data Teradata

Tumpukan teknologi target

- Amazon Redshift

Arsitektur

Untuk arsitektur tingkat tinggi untuk memigrasikan database Teradata ke Amazon Redshift, lihat pola Memigrasikan database Teradata ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT. Migrasi tidak secara otomatis mengonversi frasa Teradata RESET WHEN ke Amazon Redshift SQL. Anda dapat mengonversi ekstensi Teradata ini dengan mengikuti panduan di bagian berikutnya.

Alat

Kode

Untuk mengilustrasikan konsep RESET WHEN, pertimbangkan definisi tabel berikut di Teradata:

```
create table systest.f_account_balance
( account_id integer NOT NULL,
  month_id integer,
  balance integer )
unique primary index (account_id, month_id);
```

Jalankan kode SQL berikut untuk memasukkan data sampel ke dalam tabel:

```
BEGIN TRANSACTION;
Insert Into systest.f_account_balance values (1,1,60);
Insert Into systest.f_account_balance values (1,2,99);
Insert Into systest.f_account_balance values (1,3,94);
Insert Into systest.f_account_balance values (1,4,90);
Insert Into systest.f_account_balance values (1,5,80);
Insert Into systest.f_account_balance values (1,6,88);
Insert Into systest.f_account_balance values (1,7,90);
Insert Into systest.f_account_balance values (1,8,92);
Insert Into systest.f_account_balance values (1,9,10);
Insert Into systest.f_account_balance values (1,10,60);
Insert Into systest.f_account_balance values (1,11,80);
Insert Into systest.f_account_balance values (1,12,10);
END TRANSACTION;
```

Tabel sampel memiliki data berikut:

account_id	bulan_id	keseimbangan
1	1	60
1	2	99

1	3	94
1	4	90
1	5	80
1	6	88
1	7	90
1	8	92
1	9	10
1	10	60
1	11	80
1	12	10

Untuk setiap akun, katakanlah Anda ingin menganalisis urutan kenaikan saldo bulanan berturut-turut. Ketika saldo satu bulan kurang dari, atau sama dengan, saldo bulan sebelumnya, persyaratannya adalah mengatur ulang penghitung ke nol dan memulai kembali.

Teradata RESET SAAT menggunakan kasus

Untuk menganalisis data ini, Teradata SQL menggunakan fungsi jendela dengan agregat bersarang dan frase RESET WHEN, sebagai berikut:

```
SELECT account_id, month_id, balance,
 ( ROW_NUMBER() OVER (PARTITION BY account_id ORDER BY month_id
RESET WHEN balance <= SUM(balance) over (PARTITION BY account_id ORDER BY month_id ROWS
 BETWEEN 1 PRECEDING AND 1 PRECEDING) ) -1 ) as balance_increase
FROM systest.f_account_balance
ORDER BY 1,2;
```

Output:

account_id	bulan_id	keseimbangan	balance_increase

1	1	60	0
1	2	99	1
1	3	94	0
1	4	90	0
1	5	80	0
1	6	88	1
1	7	90	2
1	8	92	3
1	9	10	0
1	10	60	1
1	11	80	2
1	12	10	0

Kueri diproses sebagai berikut di Teradata:

1. Fungsi agregat SUM (saldo) menghitung jumlah semua saldo untuk akun tertentu dalam bulan tertentu.
2. Kami memeriksa untuk melihat apakah saldo dalam bulan tertentu (untuk akun tertentu) lebih besar dari saldo bulan sebelumnya.
3. Jika saldo meningkat, kami melacak nilai hitungan kumulatif. Jika kondisi RESET WHEN dievaluasi menjadi false, yang berarti saldo telah meningkat selama beberapa bulan berturut-turut, kami terus meningkatkan hitungan.
4. Fungsi analitis berurutan ROW_NUMBER () menghitung nilai hitungan. Ketika kita mencapai bulan yang saldo kurang dari, atau sama dengan, saldo bulan sebelumnya, kondisi RESET WHEN dievaluasi menjadi benar. Jika demikian, kita memulai partisi baru dan ROW_NUMBER () memulai ulang hitungan dari 1. Kami menggunakan ROWS BETWEEN 1 PRECEDING DAN 1 PRECEDING untuk mengakses nilai baris sebelumnya.
5. Kami mengurangi 1 untuk memastikan bahwa nilai hitungan dimulai dengan 0.

SQL setara Amazon Redshift

Amazon Redshift tidak mendukung frase RESET WHEN dalam fungsi jendela analitik SQL. Untuk menghasilkan hasil yang sama, Anda harus menulis ulang Teradata SQL menggunakan sintaks SQL asli Amazon Redshift dan sub-kueri bersarang, sebagai berikut:

```
SELECT account_id, month_id, balance,
   (ROW_NUMBER() OVER(PARTITION BY account_id, new_dynamic_part ORDER BY month_id) -1)
 as balance_increase
FROM
( SELECT account_id, month_id, balance, prev_balance,
SUM(dynamic_part) OVER (PARTITION BY account_id ORDER BY month_id ROWS BETWEEN
 UNBOUNDED PRECEDING AND CURRENT ROW) As new_dynamic_part
FROM ( SELECT account_id, month_id, balance,
SUM(balance) over (PARTITION BY account_id ORDER BY month_id ROWS BETWEEN 1 PRECEDING
 AND 1 PRECEDING) as prev_balance,
(CASE When balance <= prev_balance Then 1 Else 0 END) as dynamic_part
FROM systest.f_account_balance ) A
) B
ORDER BY 1,2;
```

Karena Amazon Redshift tidak mendukung fungsi jendela bersarang di klausa SELECT dari satu pernyataan SQL, Anda harus menggunakan dua sub-kueri bersarang.

- Dalam sub-query bagian dalam (alias A), indikator partisi dinamis (dynamic_part) dibuat dan diisi. dynamic_part diatur ke 1 jika saldo satu bulan kurang dari atau sama dengan saldo bulan sebelumnya; jika tidak, itu diatur ke 0.
- Di lapisan berikutnya (alias B), atribut new_dynamic_part dihasilkan sebagai hasil dari fungsi jendela SUM.
- Akhirnya, Anda menambahkan new_dynamic_part sebagai atribut partisi baru (partisi dinamis) ke atribut partisi yang ada (account_id) dan menerapkan fungsi jendela ROW_NUMBER () yang sama seperti di Teradata (dan minus satu).

Setelah perubahan ini, Amazon Redshift SQL menghasilkan output yang sama dengan Teradata.

Epik

Konversi RESET WHEN ke Amazon Redshift SQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi jendela Teradata Anda.	Gunakan agregat bersarang dan frase RESET WHEN sesuai dengan kebutuhan Anda.	Developer SQL
Konversi kode ke Amazon Redshift SQL.	Untuk mengonversi kode Anda, ikuti panduan di bagian “Alat” dari pola ini.	Developer SQL
Jalankan kode di Amazon Redshift.	Buat tabel Anda, muat data ke dalam tabel, dan jalankan kode Anda di Amazon Redshift.	Developer SQL

Sumber daya terkait

Referensi

- RESET WHEN Phrase (dokumentasi Teradata)
- RESET SAAT penjelasan (Stack Overflow)
- Bermigrasi ke Amazon Redshift (situs web AWS)
- Migrasikan database Teradata ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT (AWS Prescriptive Guidance)
- Mengonversi fitur temporal Teradata NORMALIZE ke Amazon Redshift SQL (AWS Prescriptive Guidance)

Alat

- Alat Konversi Skema AWS (AWS SCT)

Mitra

- Mitra Kompetensi Migrasi AWS

Menerapkan dan mengelola data lake tanpa server di AWS Cloud dengan menggunakan infrastruktur sebagai kode

Kirankumar Chandrashekar dan Abdel Jaidi, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan komputasi tanpa server dan infrastruktur sebagai kode (IAc) untuk mengimplementasikan dan mengelola data lake di Amazon Web Services (AWS) Cloud. Pola ini didasarkan pada lokakarya kerangka kerja data lake tanpa server (SDLF) yang dikembangkan oleh AWS.

SDLF adalah kumpulan sumber daya yang dapat digunakan kembali yang mempercepat pengiriman data lake perusahaan di AWS Cloud dan membantu penyebaran yang lebih cepat ke produksi. Ini digunakan untuk mengimplementasikan struktur dasar danau data dengan mengikuti praktik terbaik.

SDLF mengimplementasikan proses integrasi/penerapan berkelanjutan (CI/CD) di seluruh penerapan kode dan infrastruktur dengan menggunakan layanan AWS seperti AWS, AWS, dan AWS. CodePipeline CodeBuild CodeCommit

Pola ini menggunakan beberapa layanan tanpa server AWS untuk menyederhanakan pengelolaan data lake. Ini termasuk Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dan Amazon DynamoDB untuk penyimpanan, AWS Lambda dan AWS Glue untuk komputasi, CloudWatch dan Amazon Events, Amazon Simple Queue Service (Amazon SQS), dan AWS Step Functions untuk orkestrasi.

Layanan kode AWS CloudFormation dan AWS bertindak sebagai lapisan IAc untuk menyediakan penerapan yang dapat direproduksi dan cepat dengan pengoperasian dan administrasi yang mudah.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi.
- Klien Git, diinstal dan dikonfigurasi.
- Lokakarya SDLF, terbuka di jendela browser web dan siap digunakan.

Arsitektur

Diagram arsitektur menggambarkan proses yang digerakkan oleh peristiwa dengan langkah-langkah berikut.

1. Setelah file ditambahkan ke bucket S3 data mentah, pemberitahuan peristiwa Amazon S3 ditempatkan dalam antrean SQS. Setiap notifikasi dikirimkan sebagai file JSON, yang berisi metadata seperti nama bucket S3, kunci objek, atau stempel waktu.
2. Pemberitahuan ini digunakan oleh fungsi Lambda yang merutekan peristiwa ke proses ekstraksi, transformasi, dan pemuatan (ETL) yang benar berdasarkan metadata. Fungsi Lambda juga dapat menggunakan konfigurasi kontekstual yang disimpan dalam tabel Amazon DynamoDB. Langkah ini memungkinkan decoupling dan scaling ke beberapa aplikasi di data lake.
3. Acara ini diarahkan ke fungsi Lambda pertama dalam proses ETL, yang mengubah dan memindahkan data dari area data mentah ke area pementasan untuk danau data. Langkah pertama adalah memperbarui katalog komprehensif. Ini adalah tabel DynamoDB yang berisi semua metadata file dari data lake. Setiap baris dalam tabel ini menyimpan metadata operasional tentang satu objek yang disimpan di Amazon S3. Panggilan sinkron dibuat ke fungsi Lambda yang melakukan transformasi cahaya, yang merupakan operasi komputasi yang murah (seperti mengonversi file dari satu format ke format lainnya), pada objek S3. Karena objek baru telah ditambahkan ke bucket S3 pementasan, katalog komprehensif diperbarui dan pesan dikirim ke antrian SQS menunggu fase berikutnya di ETL.
4. Aturan CloudWatch Peristiwa memicu fungsi Lambda setiap 5 menit. Fungsi ini memeriksa apakah pesan dikirim ke antrian SQS dari fase ETL sebelumnya. Jika pesan terkirim, fungsi Lambda memulai fungsi kedua dari AWS Step Functions dalam proses ETL.
5. Transformasi berat kemudian diterapkan pada sekumpulan file. Transformasi berat ini adalah operasi yang mahal secara komputasi, seperti panggilan sinkron ke pekerjaan AWS Glue, tugas AWS Fargate, langkah EMR Amazon, atau notebook Amazon. SageMaker Metadata tabel diekstraksi dari file output dengan menggunakan crawler AWS Glue, yang memperbarui katalog AWS Glue. Metadata file juga ditambahkan ke tabel katalog komprehensif di DynamoDB. Akhirnya, langkah kualitas data yang memanfaatkan Deequ juga dijalankan.

Tumpukan teknologi

- CloudWatch Acara Amazon
- AWS CloudFormation

- AWS CodePipeline
- AWS CodeBuild
- AWS CodeCommit
- Amazon DynamoDB
- AWS Glue
- AWS Lambda
- Amazon S3
- Amazon SQS
- AWS Step Functions

Alat

- CloudWatch Acara Amazon — CloudWatch Acara menghadirkan aliran peristiwa sistem yang mendekati waktu nyata yang menjelaskan perubahan sumber daya AWS.
- AWS CloudFormation — CloudFormation membantu membuat dan menyediakan penerapan infrastruktur AWS secara dapat diprediksi dan berulang kali.
- AWS CodeBuild — CodeBuild adalah layanan build terkelola penuh yang mengompilasi kode sumber Anda, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit — CodeCommit adalah layanan kontrol versi yang dihosting oleh AWS yang dapat Anda gunakan untuk menyimpan dan mengelola aset secara pribadi (seperti kode sumber dan file biner).
- AWS CodePipeline — CodePipeline adalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak Anda secara terus menerus.
- Amazon DynamoDB - DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat dan dapat diprediksi dengan skalabilitas.
- AWS Glue — AWS Glue adalah layanan ETL yang dikelola sepenuhnya yang memudahkan penyiapan dan pemuatan data untuk analitik.
- AWS Lambda — Lambda mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.

- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel. Amazon S3 dapat digunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan data lake.
- AWS Step Functions - AWS Step Functions adalah orkestrator fungsi tanpa server yang memudahkan untuk mengurutkan fungsi AWS Lambda dan beberapa layanan AWS ke dalam aplikasi penting bisnis.
- Amazon SQS — Amazon Simple Queue Service (Amazon SQS) adalah layanan antrian pesan yang dikelola sepenuhnya yang membantu Anda memisahkan dan menskalakan layanan mikro, sistem terdistribusi, dan aplikasi tanpa server.
- Deequ - Deequ adalah alat yang membantu Anda menghitung metrik kualitas data untuk kumpulan data besar, menentukan dan memverifikasi kendala kualitas data, dan tetap mendapat informasi tentang perubahan dalam distribusi data.

Repositori kode

Kode sumber dan sumber daya untuk SDLF tersedia di repositori AWS Labs GitHub .

Epik

Siapkan CI/CD pipeline untuk menyediakan IAc

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan CI/CD pipeline untuk mengelola IAc untuk data lake.	Masuk ke AWS Management Console dan ikuti langkah-langkah dari bagian Penyiapan awal bengkel SDLF. Ini menciptakan CI/CD sumber daya awal, seperti CodeCommit repositori, CodeBuild lingkungan, dan CodePipeline saluran pipa yang menyediakan dan mengelola IAc untuk data lake.	DevOps insinyur

Kontrol versi IAc

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning CodeCommit repositori pada mesin lokal Anda.	Ikuti langkah-langkah dari bagian Deploying the foundation dari lokakarya SDLF. Ini membantu Anda mengkloning repositori Git yang menghosting IAc ke lingkungan lokal Anda. Untuk informasi selengkapnya, lihat Menghubungkan ke CodeCommit repositori dari dokumentasi. CodeCommit	DevOps insinyur
Ubah CloudFormation template.	Gunakan workstation lokal Anda dan editor kode untuk memodifikasi CloudFormation template sesuai dengan kasus penggunaan atau persyaratan Anda. Komit mereka ke repositori Git yang dikloning secara lokal. Untuk informasi selengkapnya, lihat Bekerja dengan CloudFormation templat AWS dari CloudFormation dokumentasi AWS.	DevOps insinyur
Dorong perubahan ke CodeCommit repositori.	Kode infrastruktur Anda sekarang berada di bawah kontrol versi dan modifikasi pada basis kode Anda dilacak. Saat Anda mendorong perubahan ke CodeCommi	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	t repositori, CodePipeline secara otomatis menerapka nnya ke infrastruktur Anda dan mengirimkannya ke. CodeBuild **Important** Jika Anda menggunak an AWS SAM CLI di CodeBuild, jalankan perintah `sam package` dan`sam deploy`. Jika Anda menggunakan AWS CLI, jalankan perintah `aws cloudform ation package` and`aws cloudform ation deploy .`	

Sumber daya terkait

Siapkan CI/CD pipeline untuk menyediakan IAc

- Lokakarya SDLF - Pengaturan awal

Kontrol versi IAc

- Lokakarya SDLF — Menyebarkan fondasi
- Menghubungkan ke CodeCommit repositori
- Bekerja dengan CloudFormation templat AWS

Sumber daya lainnya

- Arsitektur referensi saluran analisis data tanpa server AWS
- Dokumentasi SDLF

Menegakkan penandaan kluster EMR Amazon saat diluncurkan

Priyanka Chaudhary, Amazon Web Services

Ringkasan

Pola ini memberikan kontrol keamanan yang memastikan bahwa klaster EMR Amazon diberi tag saat dibuat.

Amazon EMR adalah layanan Amazon Web Services (AWS) untuk memproses dan menganalisis data dalam jumlah besar. Amazon EMR menawarkan layanan konfigurasi rendah yang dapat diperluas sebagai alternatif yang lebih mudah untuk menjalankan komputasi cluster internal. Anda dapat menggunakan penandaan untuk mengkategorikan sumber daya AWS dengan berbagai cara, seperti berdasarkan tujuan, pemilik, atau lingkungan. Misalnya, Anda dapat menandai kluster EMR Amazon Anda dengan menetapkan metadata khusus untuk setiap cluster. Tag terdiri dari kunci dan nilai yang Anda tentukan. Kami menyarankan Anda membuat kumpulan tag yang konsisten untuk memenuhi persyaratan organisasi Anda. Saat Anda menambahkan tag ke kluster EMR Amazon, tag tersebut juga disebarkan ke setiap instans Amazon Elastic Compute Cloud (Amazon EC2) aktif yang terkait dengan cluster. Demikian pula, saat Anda menghapus tag dari kluster EMR Amazon, tag tersebut juga akan dihapus dari setiap EC2 instance aktif yang terkait.

Kontrol detektif memantau panggilan API dan memulai CloudWatch acara Amazon Events untuk RunJobFlow,, AddTagsRemoveTags, dan. CreateTags APIs Acara ini memanggil AWS Lambda, yang menjalankan skrip Python. Fungsi Python mendapatkan ID cluster EMR Amazon dari input JSON dari acara dan melakukan pemeriksaan berikut:

- Periksa apakah klaster EMR Amazon dikonfigurasi dengan nama tag yang Anda tentukan.
- Jika tidak, kirim notifikasi Amazon Simple Notification Service (Amazon SNS) kepada pengguna dengan informasi yang relevan: nama klaster Amazon EMR, detail pelanggaran, Wilayah AWS, akun AWS, dan Nama Sumber Daya Amazon (ARN) untuk Lambda tempat notifikasi ini berasal.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif

- Bucket Amazon Simple Storage Service (Amazon S3) untuk mengunggah kode Lambda yang disediakan. Atau, Anda dapat membuat bucket S3 untuk tujuan ini, seperti yang dijelaskan di bagian Epik.
- Alamat email aktif tempat Anda ingin menerima pemberitahuan pelanggaran.
- Daftar tag wajib yang ingin Anda periksa.

Batasan

- Kontrol keamanan ini bersifat regional. Anda harus menerapkannya di setiap Wilayah AWS yang ingin Anda pantau.

Versi produk

- Amazon EMR merilis 4.8.0 dan yang lebih baru.

Arsitektur

Arsitektur alur kerja

Otomatisasi dan skala

- Jika Anda menggunakan AWS Organizations, Anda dapat menggunakan AWS Cloudformation StackSets untuk menerapkan template ini di beberapa akun yang ingin Anda pantau.

Alat

Layanan AWS

- AWS CloudFormation — AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Anda dapat mengelola dan menyediakan tumpukan di beberapa akun AWS dan Wilayah AWS.
- Amazon CloudWatch Events - Amazon CloudWatch Events menghadirkan aliran peristiwa sistem yang mendekati waktu nyata yang menjelaskan perubahan sumber daya AWS.

- Amazon EMR - Amazon EMR adalah layanan web yang menyederhanakan menjalankan kerangka kerja data besar dan memproses sejumlah besar data secara efisien.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah layanan penyimpanan objek. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) mengoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien, termasuk server web dan alamat email. Pelanggan menerima semua pesan yang dipublikasikan ke topik tempat mereka berlangganan, dan semua pelanggan ke suatu topik menerima pesan yang sama.

Kode

Pola ini mencakup lampiran berikut:

- `EMRTagValidation.zip`— Kode Lambda untuk kontrol keamanan.
- `EMRTagValidation.yml`— CloudFormation Template yang mengatur acara dan fungsi Lambda.

Epik

Siapkan ember S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan bucket S3.	Di konsol Amazon S3, pilih atau buat bucket S3 untuk meng-host file kode Lambda .zip. Bucket S3 ini harus berada di Wilayah AWS yang sama dengan cluster EMR Amazon yang ingin Anda pantau. Nama bucket Amazon S3 unik secara global, dan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	namespace-nya digunakan bersama oleh semua akun AWS. Nama bucket S3 tidak dapat menyertakan garis miring terdepan.	
Unggah kode Lambda.	Unggah file kode Lambda .zip yang disediakan di bagian Lampiran ke bucket S3.	Arsitek awan

Menerapkan template AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan CloudFormation template AWS.	Buka CloudFormation konsol AWS di Wilayah AWS yang sama dengan bucket S3 Anda dan terapkan template. Untuk informasi selengkap nya tentang penerapan CloudFormation templat AWS, lihat Membuat tumpukan di CloudFormation konsol AWS dalam CloudFormation dokumentasi.	Arsitek awan
Lengkapi parameter dalam template.	Ketika Anda meluncurkan template, Anda akan diminta untuk informasi berikut: • Bucket S3: Tentukan bucket yang Anda buat atau pilih di epik pertama. Di sinilah	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda mengunggah kode Lambda terlampir (file.zip). • Kunci S3 : Tentukan lokasi file.zip Lambda di bucket S3 Anda (misalnya, nama file.zip atau controls/ filename .zip). Jangan sertakan garis miring terdepan. • Email pemberitahuan: Berikan alamat email aktif tempat Anda ingin menerima notifikasi Amazon SNS. • Menandai nama kunci: Berikan tag yang ingin Anda periksa, dalam daftar yang dipisahkan koma (misalnya,,`Applicati onID` ,`Environment`). `0wner` CloudWatch Acara Acara memantau klaster untuk tag ini dan mengirimk an pemberitahuan jika tidak ditemukan. • Level logging Lambda: Tentukan level logging dan frekuensi untuk fungsi Lambda. Gunakan Info untuk mencatat pesan informasi terperinci tentang kemajuan, Kesalahan untuk peristiwa kesalahan yang masih memungkinkan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	penerapan dilanjutkan, dan Peringatan untuk situasi yang berpotensi berbahaya.	

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Ketika CloudFormation template berhasil digunakan , ia mengirimkan email berlangganan ke alamat email yang Anda berikan. Anda harus mengonfirmasi langganan email ini untuk mulai menerima pemberita huan pelanggaran.	Arsitek awan

Sumber daya terkait

- Panduan pengembang AWS Lambda
- Menandai cluster di Amazon EMR

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Pastikan pencatatan EMR Amazon ke Amazon S3 diaktifkan saat peluncuran

Priyanka Chaudhary, Amazon Web Services

Ringkasan

Pola ini menyediakan kontrol keamanan yang memantau konfigurasi logging untuk klaster EMR Amazon yang berjalan di Amazon Web Services (AWS).

Amazon EMR adalah alat AWS untuk pemrosesan dan analisis data besar. Amazon EMR menawarkan layanan konfigurasi rendah yang dapat diperluas sebagai alternatif untuk menjalankan komputasi cluster internal. Amazon EMR menyediakan dua jenis cluster EMR.

- Cluster EMR Amazon sementara: Kluster EMR Amazon sementara secara otomatis mati dan berhenti mengeluarkan biaya saat pemrosesan selesai.
- Cluster EMR Amazon yang persisten: Cluster EMR Amazon yang persisten terus berjalan setelah pekerjaan pemrosesan data selesai.

Amazon EMR dan Hadoop menghasilkan berkas log yang melaporkan status pada klaster. Secara default, ini ditulis ke node master di direktori mnt/var/log//. Bergantung pada bagaimana Anda mengonfigurasi cluster saat meluncurkannya, Anda juga dapat menyimpan log ini ke Amazon Simple Storage Service (Amazon S3) dan melihatnya melalui alat debugging grafis. Perhatikan bahwa pencatatan Amazon S3 hanya dapat ditentukan saat cluster diluncurkan. Dengan konfigurasi ini, log dikirim dari node utama ke lokasi Amazon S3 setiap 5 menit. Untuk cluster sementara, pencatatan Amazon S3 penting karena cluster menghilang saat pemrosesan selesai, dan file log ini dapat digunakan untuk men-debug pekerjaan yang gagal.

Pola ini menggunakan CloudFormation template AWS untuk menerapkan kontrol keamanan yang memantau panggilan API dan memulai CloudWatch Acara Amazon di "RunJobFlow.” Pemicunya memanggil AWS Lambda, yang menjalankan skrip Python. Fungsi Lambda mengambil ID cluster EMR dari input JSON peristiwa dan juga memeriksa URI log Amazon S3. Jika URI Amazon S3 tidak ditemukan, fungsi Lambda mengirimkan notifikasi Amazon Simple Notification Service (Amazon SNS) Simple Notification Service (Amazon SNS) yang merinci nama cluster EMR, detail pelanggaran, Wilayah AWS, akun AWS, dan Lambda Amazon Resource Name (ARN) tempat notifikasi tersebut bersumber.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Bucket S3 untuk file.zip kode Lambda
- Alamat email tempat Anda ingin menerima pemberitahuan pelanggaran

Batasan

- Kontrol detektif ini bersifat regional dan harus diterapkan di Wilayah AWS yang ingin Anda pantau.

Versi produk

- Amazon EMR merilis 4.8.0 dan yang lebih baru

Arsitektur

Tumpukan teknologi target

- CloudWatch Acara Amazon Events
- Amazon EMR
- Fungsi Lambda
- Bucket S3
- Amazon SNS

Arsitektur target

Otomatisasi dan skala

- Jika Anda menggunakan AWS Organizations, Anda dapat menggunakan AWS CloudFormation StackSets untuk menerapkan template ini di beberapa akun yang ingin Anda pantau.

Alat

Alat

- AWS CloudFormation — AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS menggunakan infrastruktur sebagai kode.
- AWS Cloudwatch Events — AWS CloudWatch Events menghadirkan aliran peristiwa sistem yang mendekati waktu nyata yang menjelaskan perubahan dalam sumber daya AWS.
- Amazon EMR - Amazon EMR adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar.
- AWS Lambda — AWS Lambda mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon S3 - Amazon S3 adalah antarmuka layanan web yang dapat Anda gunakan untuk menyimpan dan mengambil sejumlah data dari mana saja di web.
- Amazon SNS — Amazon SNS adalah layanan web yang mengoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien, termasuk server web dan alamat email.

Kode

- File.zip proyek tersedia sebagai lampiran.

Epik

Tentukan bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan bucket S3.	Untuk meng-host file kode Lambda .zip, pilih atau buat bucket S3 dengan nama unik yang tidak berisi garis miring di depan. Nama bucket S3 unik secara global, dan	Arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	namespace dibagikan oleh semua akun AWS. Bucket S3 Anda harus berada di Wilayah AWS yang sama dengan cluster EMR Amazon yang sedang dievaluasi.	

Unggah kode Lambda ke bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode Lambda ke bucket S3.	Unggah file kode Lambda .zip yang disediakan di bagian “Lampiran” ke bucket S3. Bucket S3 harus berada di Wilayah yang sama dengan cluster EMR Amazon yang sedang dievaluasi.	Arsitek Cloud

Menerapkan template AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan CloudFormation template AWS.	Di CloudFormation konsol AWS, di Wilayah yang sama dengan bucket S3 Anda, terapkan CloudFormation template AWS yang disediakan sebagai lampiran ke pola ini. Dalam epik berikutnya, berikan nilai untuk parameter. Untuk informasi selengkapnya tentang penerapan CloudForm	Arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ation templat AWS, lihat bagian “Sumber daya terkait”.	

Lengkapi parameter dalam CloudFormation template AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Beri nama ember S3.	Masukkan nama bucket S3 yang Anda buat di epik pertama.	Arsitek Cloud
Berikan kunci Amazon S3.	<directory><file-name>Berikan lokasi file kode Lambda .zip di bucket S3 Anda, tanpa garis miring (misalnya,/.zip).	Arsitek Cloud
Berikan alamat email.	Berikan alamat email aktif untuk menerima notifikasi Amazon SNS.	Arsitek Cloud
Tentukan tingkat logging.	Tentukan tingkat logging dan frekuensi untuk fungsi Lambda Anda. “Info” menunjuk pesan informasi terperinci tentang kemajuan aplikasi. “Kesalahan” menunjuk peristiwa kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. “Peringat an” menunjuk situasi yang berpotensi berbahaya.	Arsitek Cloud

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Ketika template berhasil digunakan, ia mengirimkan pesan email berlangganan ke alamat email yang disediaka n. Anda harus mengonfirmasi langganan email ini untuk menerima pemberitahuan pelanggaran.	Arsitek Cloud

Sumber daya terkait

- AWS Lambda
- Pencatatan EMR Amazon
- Menerapkan templat AWS CloudFormation

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Hasilkan data pengujian menggunakan pekerjaan AWS Glue dan Python

Moinul Al-Mamun, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara cepat dan mudah menghasilkan jutaan file sampel secara bersamaan dengan membuat pekerjaan AWS Glue yang ditulis dengan Python. File sampel disimpan dalam bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3). Kemampuan untuk menghasilkan sejumlah besar file sampel dengan cepat penting untuk menguji atau mengevaluasi layanan di AWS Cloud. Misalnya, Anda dapat menguji kinerja DataBrew pekerjaan AWS Glue Studio atau AWS Glue dengan melakukan analisis data pada jutaan file kecil dalam awalan Amazon S3.

Meskipun Anda dapat menggunakan layanan AWS lainnya untuk menghasilkan kumpulan data sampel, kami menyarankan Anda menggunakan AWS Glue. Anda tidak perlu mengelola infrastruktur apa pun karena AWS Glue adalah layanan pemrosesan data tanpa server. Anda bisa membawa kode Anda dan menjalankannya di klaster AWS Glue. Selain itu, AWS Glue menyediakan, mengonfigurasi, dan menskalakan sumber daya yang diperlukan untuk menjalankan pekerjaan Anda. Anda hanya membayar untuk sumber daya yang digunakan pekerjaan Anda saat menjalankan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi untuk bekerja dengan akun AWS

Versi produk

- Python 3.9
- AWS CLI versi 2

Batasan

Jumlah maksimum pekerjaan AWS Glue per pemicu adalah 50. Untuk informasi selengkapnya, lihat titik akhir dan kuota AWS Glue.

Arsitektur

Diagram berikut menggambarkan contoh arsitektur yang berpusat di sekitar pekerjaan AWS Glue yang menulis outputnya (yaitu, file sampel) ke bucket S3.

Diagram mencakup alur kerja berikut:

1. Anda menggunakan AWS CLI, AWS Management Console, atau API untuk memulai pekerjaan AWS Glue. AWS CLI atau API memungkinkan Anda mengotomatiskan paralelisasi pekerjaan yang dipanggil dan mengurangi runtime untuk menghasilkan file sampel.
2. Pekerjaan AWS Glue menghasilkan konten file secara acak, mengonversi konten menjadi format CSV, dan kemudian menyimpan konten sebagai objek Amazon S3 di bawah awalan umum. Setiap file kurang dari satu kilobyte. Pekerjaan AWS Glue menerima dua parameter pekerjaan yang ditentukan pengguna: dan. `START_RANGE` `END_RANGE` Anda dapat menggunakan parameter ini untuk mengatur nama file dan jumlah file yang dihasilkan di Amazon S3 oleh setiap pekerjaan yang dijalankan. Anda dapat menjalankan beberapa instance pekerjaan ini secara paralel (misalnya, 100 instance).

Alat

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Praktik terbaik

Pertimbangkan praktik terbaik AWS Glue berikut saat Anda menerapkan pola ini:

- Gunakan jenis pekerja AWS Glue yang tepat untuk mengurangi biaya. Kami menyarankan Anda memahami properti yang berbeda dari jenis pekerja, dan kemudian memilih jenis pekerja yang tepat untuk beban kerja Anda berdasarkan persyaratan CPU dan memori. Untuk pola ini, kami menyarankan Anda menggunakan pekerjaan shell Python sebagai jenis pekerjaan Anda untuk meminimalkan DPU dan mengurangi biaya. Untuk informasi selengkapnya, lihat Menambahkan lowongan di AWS Glue di Panduan Pengembang AWS Glue.
- Gunakan batas konkurensi yang tepat untuk skala pekerjaan Anda. Kami menyarankan Anda mendasarkan konkurensi maksimum pekerjaan AWS Glue Anda pada kebutuhan waktu Anda dan jumlah file yang diperlukan.
- Mulailah membuat sejumlah kecil file pada awalnya. Untuk mengurangi biaya dan menghemat waktu saat Anda membangun pekerjaan AWS Glue, mulailah dengan sejumlah kecil file (seperti 1.000). Ini dapat membuat pemecahan masalah lebih mudah. Jika menghasilkan sejumlah kecil file berhasil, maka Anda dapat menskalakan ke jumlah file yang lebih besar.
- Jalankan secara lokal terlebih dahulu. Untuk mengurangi biaya dan menghemat waktu saat Anda membangun pekerjaan AWS Glue, mulailah pengembangan secara lokal dan uji kode Anda. Untuk petunjuk tentang menyiapkan wadah Docker yang dapat membantu Anda menulis pekerjaan AWS Glue extract, transform, and load (ETL) baik di shell maupun di lingkungan pengembangan terintegrasi (IDE), lihat pekerjaan Mengembangkan AWS Glue ETL secara lokal menggunakan postingan kontainer di AWS Big Data Blog.

Untuk praktik terbaik AWS Glue lainnya, lihat Praktik terbaik dalam dokumentasi AWS Glue.

Epik

Buat bucket S3 tujuan dan peran IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 untuk menyimpan file.	Buat bucket S3 dan awalan di dalamnya.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Pola ini menggunakan `s3://{your-s3-bucket-name}/small-files/` lokasi untuk tujuan demonstrasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan peran IAM.	Anda harus membuat peran IAM yang dapat digunakan oleh pekerjaan AWS Glue Anda untuk menulis ke bucket S3 Anda. 1. Buat peran IAM (misalnya, disebut `"AWSGlueServiceRole-smallfiles"`). 2. Pilih AWS Glue sebagai entitas tepercaya kebijakan. 3. Lampirkan kebijakan terkelola AWS yang dipanggil `"AWSGlueServiceRole"` ke peran tersebut. 4. Buat kebijakan inline atau kebijakan terkelola pelanggan yang dipanggil `"s3-small-file-access"` berdasarkan konfigurasi berikut. Ganti `"{bucket}"` dengan nama bucket Anda. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"s3:GetObject",
	
	"s3:PutObject"
	],
	"Resource
	": [
	
	"arn:aws:s3:::{buc
	ket}/small-files/i
	nput/*"
	]
	}
	]
	}</pre> 5. Lampirkan `"s3-small-file-access"` kebijakan ke peran Anda.	

Membuat dan mengonfigurasi pekerjaan AWS Glue untuk menangani proses bersamaan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pekerjaan AWS Glue.	Anda harus membuat pekerjaan AWS Glue yang menghasilkan konten Anda dan menyimpannya di bucket S3. Buat pekerjaan AWS Glue, lalu konfigurasikan pekerjaan Anda dengan menyelesaikan langkah-langkah berikut: 1. Masuk ke AWS Management Console dan buka konsol AWS Glue.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di panel navigasi, di bawah Integrasi Data dan ETL, pilih Pekerjaan. 3. Di bagian Buat pekerjaan , pilih editor skrip Python Shell. 4. Di bagian Opsi, pilih Buat skrip baru dengan kode boilerplate, lalu pilih Buat. 5. Pilih Detail Pekerjaan. 6. Untuk Nama, masukkan create_small_files. 7. Untuk Peran IAM, pilih peran IAM yang Anda buat sebelumnya. 8. Di bagian Pekerjaan ini berjalan, pilih Skrip baru yang akan ditulis oleh Anda. 9. Perluas Properti lanjutan. 10. Untuk Konkurensi maksimum, masukkan 100 untuk tujuan demonstra si. Catatan: Konkurensi maksimum mendefinisikan berapa banyak contoh pekerjaan yang dapat Anda jalankan secara paralel. 11. Pilih Simpan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui kode pekerjaan.	1. Buka konsol AWS Glue.
	2. Di panel navigasi, pilih Tugas.
	3. Di bagian Pekerjaan Anda, pilih pekerjaan yang Anda buat sebelumnya.
	4. Pilih tab Script, dan kemudian perbarui skrip berdasarkan kode berikut. Perbarui`BUCKET_NAME` ,`PREFIX`, dan `text_str` variabel dengan nilai Anda.
	`from awsglue.utils import getResolvedOptions`
	`import sys`
	`import boto3`
	`from random import randrange`
	
	`# Two arguments`
	`args = getResolvedOptions(sys.argv, ['START_RANGE', 'END_RANGE'])`
	
	`START_RANGE = int(args['START_RANGE'])`
	`END_RANGE = int(args['END_RANGE'])`
	
	`BUCKET_NAME = '{BUCKET_NAME}'`
	`PREFIX = 'small-files/input/'`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>s3 = boto3.res
	ource('s3')
	for x in range(STA
	RT_RANGE, END_RANGE
	):
	    # generate file
	 name
	    file_name =
	 f"input_{x}.txt"
	    # generate text
	    text_str =
	 str(randrange(1000
	00))+","+str(randr
	ange(100000))+", "
	 + str(randrange(1000
	0000)) + "," +
	 str(randrange(1000
	0))
	    # write in s3
	s3.Object(BUCKE
	T_NAME, PREFIX +
	 file_name).put(Bod
	y=text_str)</pre>
	5. Pilih Simpan.	

Jalankan pekerjaan AWS Glue dari baris perintah atau konsol

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan pekerjaan AWS Glue dari baris perintah.	Untuk menjalankan tugas AWS Glue Anda dari AWS CLI, jalankan perintah berikut menggunakan nilai Anda:
	<pre>cmd:~$ aws glue start-
	job-run --job-name
	 create_small_files
	 --arguments '{"--STAR</pre>
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`T_RANGE":"0","--EN D_RANGE":"1000000"}' cmd:~$ aws glue start-job-run --job-name create_small_files --arguments '{"--STAR T_RANGE":"1000000" ,"--END_RANGE":"20 00000"}'` **Note** Untuk petunjuk tentang menjalank an tugas AWS Glue dari AWS Managemen t Console, lihat tugas Jalankan AWS Glue di cerita AWS Management Console dalam pola ini. **Tip** Sebaiknya gunakan AWS CLI untuk menjalankan pekerjaan AWS Glue jika Anda ingin menjalankan beberapa eksekusi sekaligus dengan parameter berbeda, seperti yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	ditunjukkan pada contoh di atas.
	
	Untuk menghasilkan semua perintah AWS CLI yang diperlukan untuk menghasilkan sejumlah file yang ditentukan menggunakan faktor paralelisasi tertentu, jalankan kode bash berikut (menggunakan nilai Anda):
	
	`# define parameters`
	`NUMBER_OF_FILES=`
	`10000000;`
	`PARALLELIZATION=50;`
	
	`# initialize`
	`_SB=0;`
	
	`# generate commands`
	`for i in $(seq 1`
	` $PARALLELIZATION);`
	`do`
	`  echo aws glue`
	` start-job-run --`
	`job-name create_sm`
	`all_files --argumen`
	`ts "'"'{"--START_RANG`
	`E":"'$(((NUMBER_OF`
	`_FILES/PARALLELIZA`
	`TION) * (i-1) +`
	` _SB))'","--END_RAN`
	`GE":"'$(((NUMBER_O`
	`F_FILES/PARALLELIZ`
	`ATION) * (i)))'"}'"'";`
	`  _SB=1;`
	`done`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jika Anda menggunakan skrip di atas, pertimbangkan hal berikut: • Skrip menyederhanakan pemanggilan dan pembuatan file kecil dalam skala besar. • Perbarui `NUMBER_OF_FILES` dan `PARALLELIZATION` dengan nilai-nilai Anda. • Script di atas mencetak daftar perintah yang harus Anda jalankan. Salin perintah output tersebut, lalu jalankan di terminal Anda. • Jika Anda ingin menjalankan perintah langsung dari dalam skrip, hapus `echo` pernyataan di baris 11. **Note** Untuk melihat contoh output dari skrip di atas, lihat Output skrip Shell di bagian Informasi tambahan dari pola ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan pekerjaan AWS Glue di AWS Management Console.	1. Masuk ke AWS Management Console dan buka konsol AWS Glue. 2. Di panel navigasi, di bawah Integrasi Data dan ETL, pilih Pekerjaan. 3. Di bagian Pekerjaan Anda, pilih pekerjaan Anda. 4. Di bagian Parameter (opsional), perbarui parameter Anda. 5. Pilih Tindakan, lalu pilih Jalankan pekerjaan. 6. Ulangi langkah 3-5 sebanyak yang Anda butuhkan. Misalnya, untuk membuat 10 juta file, ulangi proses ini 10 kali.	Pengembang aplikasi
Periksa status pekerjaan AWS Glue Anda.	1. Buka konsol AWS Glue. 2. Di panel navigasi, pilih Tugas. 3. Di bagian Pekerjaan Anda, pilih pekerjaan yang Anda buat sebelumnya (yaitu, `create_small_files`). 4. Untuk mengetahui kemajuan dan pembuatan file Anda, tinjau Run ID, Run Status, dan kolom lainnya.	Pengembang aplikasi

Sumber daya terkait

Referensi

- Registri Data Terbuka di AWS
- Kumpulan data untuk analitik
- Buka Data di AWS
- Menambahkan pekerjaan di AWS Glue
- Memulai AWS Glue

Panduan dan pola

- Praktik terbaik AWS Glue
- Aplikasi pengujian beban

Informasi tambahan

Tes benchmarking

Pola ini digunakan untuk menghasilkan 10 juta file menggunakan parameter paralelisasi yang berbeda sebagai bagian dari uji benchmarking. Tabel berikut menunjukkan output dari tes:

Paralelisasi	Jumlah file yang dihasilkan oleh pekerjaan yang dijalankan	Durasi Job	Kecepatan
10	1.000.000	6 jam, 40 menit	Sangat lambat
50	200.000	80 menit	Sedang
100	100.000	40 menit	Cepat

Jika Anda ingin membuat proses lebih cepat, Anda dapat mengonfigurasi lebih banyak proses bersamaan dalam konfigurasi pekerjaan Anda. Anda dapat dengan mudah menyesuaikan konfigurasi

pekerjaan berdasarkan kebutuhan Anda, tetapi perlu diingat bahwa ada batas kuota layanan AWS Glue. Untuk informasi selengkapnya, lihat titik akhir dan kuota AWS Glue.

Keluaran skrip shell

Contoh berikut menunjukkan output skrip shell dari pekerjaan Jalankan AWS Glue dari cerita baris perintah dalam pola ini.

```
user@MUC-1234567890 MINGW64 ~
  $ # define parameters
  NUMBER_OF_FILES=10000000;
  PARALLELIZATION=50;
  # initialize
  _SB=0;

  # generate commands
  for i in $(seq 1 $PARALLELIZATION);
   do
         echo aws glue start-job-run --job-name create_small_files --arguments
 "'"'{"--START_RANGE":"'$(((NUMBER_OF_FILES/PARALLELIZATION)  (i-1) + SB))'","--
ENDRANGE":"'$(((NUMBER_OF_FILES/PARALLELIZATION)  (i)))'"}'"'";
         _SB=1;
   done

  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"0","--END_RANGE":"200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"200001","--END_RANGE":"400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"400001","--END_RANGE":"600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"600001","--END_RANGE":"800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"800001","--END_RANGE":"1000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1000001","--END_RANGE":"1200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1200001","--END_RANGE":"1400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1400001","--END_RANGE":"1600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1600001","--END_RANGE":"1800000"}'
```

```
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1800001","--END_RANGE":"2000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2000001","--END_RANGE":"2200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2200001","--END_RANGE":"2400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2400001","--END_RANGE":"2600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2600001","--END_RANGE":"2800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2800001","--END_RANGE":"3000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3000001","--END_RANGE":"3200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3200001","--END_RANGE":"3400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3400001","--END_RANGE":"3600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3600001","--END_RANGE":"3800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3800001","--END_RANGE":"4000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4000001","--END_RANGE":"4200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4200001","--END_RANGE":"4400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4400001","--END_RANGE":"4600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4600001","--END_RANGE":"4800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4800001","--END_RANGE":"5000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5000001","--END_RANGE":"5200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5200001","--END_RANGE":"5400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5400001","--END_RANGE":"5600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5600001","--END_RANGE":"5800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5800001","--END_RANGE":"6000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6000001","--END_RANGE":"6200000"}'
```

```
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6200001","--END_RANGE":"6400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6400001","--END_RANGE":"6600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6600001","--END_RANGE":"6800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6800001","--END_RANGE":"7000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7000001","--END_RANGE":"7200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7200001","--END_RANGE":"7400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7400001","--END_RANGE":"7600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7600001","--END_RANGE":"7800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7800001","--END_RANGE":"8000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8000001","--END_RANGE":"8200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8200001","--END_RANGE":"8400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8400001","--END_RANGE":"8600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8600001","--END_RANGE":"8800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8800001","--END_RANGE":"9000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9000001","--END_RANGE":"9200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9200001","--END_RANGE":"9400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9400001","--END_RANGE":"9600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9600001","--END_RANGE":"9800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9800001","--END_RANGE":"10000000"}'

  user@MUC-1234567890 MINGW64 ~
```

FAQ

Berapa banyak pekerjaan yang berjalan bersamaan atau paralel yang harus saya gunakan?

Jumlah proses bersamaan dan pekerjaan paralel tergantung pada kebutuhan waktu Anda dan jumlah file pengujian yang diinginkan. Kami menyarankan Anda memeriksa ukuran file yang Anda buat. Pertama, periksa berapa lama waktu yang dibutuhkan pekerjaan AWS Glue untuk menghasilkan jumlah file yang Anda inginkan. Kemudian, gunakan jumlah lari bersamaan yang tepat untuk memenuhi tujuan Anda. Misalnya, jika Anda berasumsi bahwa 100.000 file membutuhkan waktu 40 menit untuk menyelesaikan proses tetapi waktu target Anda adalah 30 menit, maka Anda harus meningkatkan pengaturan konkurensi untuk pekerjaan AWS Glue Anda.

Jenis konten apa yang dapat saya buat menggunakan pola ini?

Anda dapat membuat semua jenis konten, seperti file teks dengan pembatas yang berbeda (misalnya, PIPE, JSON, atau CSV). Pola ini menggunakan Boto3 untuk menulis ke file dan kemudian menyimpan file dalam ember S3.

Tingkat izin IAM apa yang saya perlukan di bucket S3?

Anda harus memiliki kebijakan berbasis identitas yang memungkinkan `Write` akses ke objek di bucket S3 Anda. Untuk informasi selengkapnya, lihat Amazon S3: Mengizinkan akses baca dan tulis ke objek dalam bucket S3 dalam dokumentasi Amazon S3.

Mengkonsumsi data IoT secara hemat biaya langsung ke Amazon S3 menggunakan AWS IoT Greengrass

Sebastian Viviani dan Rizwan Syed, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara menyerap data Internet of Things (IoT) secara hemat biaya langsung ke bucket Amazon Simple Storage Service (Amazon S3) dengan menggunakan perangkat AWS IoT Greengrass Version 2. Perangkat menjalankan komponen khusus yang membaca data IoT dan menyimpan data dalam penyimpanan persisten (yaitu, disk atau volume lokal). Kemudian, perangkat memampatkan data IoT menjadi file Apache Parquet dan mengunggah data secara berkala ke bucket S3.

Jumlah dan kecepatan data IoT yang Anda konsumsi hanya dibatasi oleh kemampuan perangkat keras tepi dan bandwidth jaringan Anda. Anda dapat menggunakan Amazon Athena untuk menganalisis data yang Anda konsumsi secara hemat biaya. Athena mendukung file Apache Parquet terkompresi dan visualisasi data dengan menggunakan Amazon Managed Grafana.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Gateway edge yang berjalan pada AWS IoT Greengrass Versi 2 dan mengumpulkan data dari sensor (Sumber data dan proses pengumpulan data berada di luar cakupan pola ini, tetapi Anda dapat menggunakan hampir semua jenis data sensor. Pola ini menggunakan broker MQTT lokal dengan sensor atau gateway yang mempublikasikan data secara lokal.)
- Komponen AWS IoT Greengrass, peran, dan dependensi SDK
- Komponen pengelola aliran untuk mengunggah data ke bucket S3
- AWS SDK for Java, AWS SDK JavaScript for, atau AWS SDK for Python (Boto3) untuk menjalankan APIs

Batasan

- Data dalam pola ini tidak diunggah secara real time ke bucket S3. Ada periode penundaan, dan Anda dapat mengkonfigurasi periode penundaan. Data di-buffer sementara di perangkat edge dan kemudian diunggah setelah periode berakhir.
- SDK hanya tersedia di Java, Node.js, dan Python.

Arsitektur

Tumpukan teknologi target

- Amazon S3
- AWS IoT Greengrass
- Pialang MQTT
- Komponen manajer aliran

Arsitektur target

Diagram berikut menunjukkan arsitektur yang dirancang untuk menyerap data sensor IoT dan menyimpan data tersebut dalam bucket S3.

Diagram menunjukkan alur kerja berikut:

1. Beberapa sensor (misalnya, suhu dan katup) pembaruan dipublikasikan ke broker MQTT lokal.
2. Kompresor file Parket yang berlangganan sensor ini memperbarui topik dan menerima pembaruan ini.
3. Kompresor file Parquet menyimpan pembaruan secara lokal.
4. Setelah periode berlalu, file yang disimpan dikompresi menjadi file Parket dan diteruskan ke pengelola aliran untuk diunggah ke bucket S3 yang ditentukan.
5. Manajer aliran mengunggah file Parket ke bucket S3.

> **Note**
>
> Stream manager (`StreamManager`) adalah komponen terkelola. Untuk contoh cara mengekspor data ke Amazon S3, lihat Stream manager di dokumentasi AWS IoT

Greengrass. Anda dapat menggunakan broker MQTT lokal sebagai komponen atau broker lain seperti Eclipse Mosquitto.

Alat

Alat AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS IoT Greengrass adalah layanan runtime dan cloud IoT edge open source yang membantu Anda membangun, menerapkan, dan mengelola aplikasi IoT di perangkat Anda.

Alat-alat lainnya

- Apache Parquet adalah format file data berorientasi kolom sumber terbuka yang dirancang untuk penyimpanan dan pengambilan.
- MQTT (Message Queuing Telemetry Transport) adalah protokol pesan ringan yang dirancang untuk perangkat terbatas.

Praktik terbaik

Gunakan format partisi yang tepat untuk data yang diunggah

Tidak ada persyaratan khusus untuk nama awalan root di bucket S3 (misalnya, `"myAwesomeDataSet/"` atau`"dataFromSource"`), tetapi kami menyarankan Anda menggunakan partisi dan awalan yang bermakna sehingga mudah untuk memahami tujuan kumpulan data.

Kami juga menyarankan Anda menggunakan partisi yang tepat di Amazon S3 sehingga kueri berjalan secara optimal pada kumpulan data. Dalam contoh berikut, data dipartisi dalam format HIVE sehingga jumlah data yang dipindai oleh setiap kueri Athena dioptimalkan. Ini meningkatkan kinerja dan mengurangi biaya.

```
s3://<ingestionBucket>/<rootPrefix>/year=YY/month=MM/day=DD/
HHMM_<suffix>.parquet
```

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	1. Buat bucket S3 atau gunakan bucket yang sudah ada. 2. Buat awalan yang berarti untuk bucket S3 tempat Anda ingin menyerap data IoT (misalnya,). `s3:\\<bucket>\<prefix>` 3. Rekam awalan Anda untuk digunakan nanti.	Pengembang aplikasi
Tambahkan izin IAM ke bucket S3.	Untuk memberi pengguna akses tulis ke bucket S3 dan awalan yang Anda buat sebelumnya, tambahkan kebijakan IAM berikut ke peran AWS IoT Greengrass Anda: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "S3DataUpload", "Effect": "Allow", "Action": [ "s3:List*", "s3:Put*"`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>],
	"Resource":
	[
	
	"arn:aws:s3:::<ingestionBucket>",
	
	"arn:aws:s3:::<ingestionBucket>/<prefix>/*"
	]
	}
	]
	}</pre>
	Untuk informasi selengkapnya, lihat Membuat kebijakan IAM untuk mengakses sumber daya Amazon S3 dalam dokumentasi Aurora.
	
	Selanjutnya, perbarui kebijakan sumber daya (jika diperlukan) untuk bucket S3 agar memungkinkan akses tulis dengan prinsip AWS yang benar.	

Bangun dan terapkan komponen AWS IoT Greengrass

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui resep komponen.	Perbarui konfigurasi komponen saat Anda membuat penerapan berdasarkan contoh berikut:	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>{
	  "region": "<region>",
	  "parquet_period":
	 <period>,
	  "s3_bucket":
	 "<s3Bucket>",
	  "s3_key_prefix":
	 "<s3prefix>"
	}</pre>
	Ganti <region> dengan Wilayah AWS Anda, <period> dengan interval periodik Anda, <s3Bucket> dengan bucket S3 Anda, dan <s3prefix> dengan awalan Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat komponen.	Lakukan salah satu tindakan berikut: • Buat komponen. • Tambahkan komponen ke CI/CD pipeline (jika ada). Pastikan untuk menyalin artefak dari repositori artefak ke ember artefak AWS IoT Greengrass. Kemudian, buat atau perbarui komponen AWS IoT Greengrass Anda. • **Note** Tambahkan broker MQTT sebagai komponen atau tambahkan secara manual nanti. : Keputusan ini memengaruhi skema otentikasi yang dapat Anda gunakan dengan broker. Menambahkan broker secara manual memisahkan broker dari AWS IoT Greengrass dan memungkinkan skema otentikasi broker yang didukung.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Komponen broker yang disediaka n AWS memiliki skema otentikasi yang telah ditentuka n sebelumnya. Untuk informasi lebih lanjut, lihat broker MQTT 3.1.1 (Moquette) dan broker MQTT 5 (EMQX).	
Perbarui klien MQTT.	Kode sampel tidak menggunakan otentikasi karena komponen terhubung secara lokal ke broker. Jika skenario Anda berbeda, perbarui bagian klien MQTT sesuai kebutuhan. Selain itu, lakukan hal berikut: 1. Perbarui topik MQTT dalam langganan. 2. Perbarui parser pesan MQTT sesuai kebutuhan karena pesan dari setiap sumber mungkin berbeda.	Pengembang aplikasi

Tambahkan komponen ke perangkat inti AWS IoT Greengrass Versi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui penyebaran perangkat inti.	Jika penerapan perangkat inti AWS IoT Greengrass Versi 2 sudah ada, revisi penerapan nya. Jika penerapan tidak ada, buat penerapan baru. Untuk memberi komponen nama yang benar, perbarui konfigurasi pengelola log untuk komponen baru (jika diperlukan) berdasarkan hal berikut: `{ "logsUploaderConfiguration": { "systemLogsConfiguration": { ... }, "componentLogsConfigurationMap": { "<com.iot.ingest.parquet>": { "minimumLogLevel": "INFO", "diskSpaceLimit": "20", "diskSpaceLimitUnit": "MB", "deleteLogFileAfterCloudUpload": "false" } ... } }, ...`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"periodicUploadInt
	ervalSec": "300"
	}</pre>
	Terakhir, selesaikan revisi penerapan untuk perangkat inti AWS IoT Greengrass Anda.	

Verifikasi konsumsi data ke dalam bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa log untuk volume AWS IoT Greengrass.	Periksa hal berikut: • Klien MQTT berhasil terhubung ke broker MQTT lokal. • Klien MQTT berlangganan topik yang benar. • Pesan pembaruan sensor datang ke broker tentang topik MQTT. • Kompresi parket terjadi pada setiap interval periodik.	Pengembang aplikasi
Periksa ember S3.	Verifikasi apakah data sedang diunggah ke bucket S3. Anda dapat melihat file yang diunggah di setiap periode. Anda juga dapat memverifi kasi apakah data diunggah ke	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bucket S3 dengan menanyaka n data di bagian berikutnya.	

Mengatur pertanyaan dari Athena

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database dan tabel.	1. Buat database AWS Glue (jika diperlukan). 2. Buat tabel di AWS Glue secara manual atau dengan menjalankan crawler di AWS Glue.	Pengembang aplikasi
Berikan Athena akses ke data.	1. Perbarui izin untuk memungkinkan Athena mengakses bucket S3. Untuk informasi selengkap nya, lihat Akses halus ke database dan tabel di Katalog Data AWS Glue di dokumentasi Athena. 2. Kueri tabel di database Anda.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Klien MQTT gagal terhubung	• Validasi izin pada broker MQTT. Jika Anda memiliki broker MQTT dari AWS, lihat broker MQTT 3.1.1 (Moquette) dan broker MQTT 5 (EMQX).

Isu	Solusi
	• Validasi kredensil pada klien MQTT. Jika Anda memiliki broker MQTT dari AWS, lihat broker MQTT 3.1.1 (Moquette) dan broker MQTT 5 (EMQX).
Klien MQTT gagal berlangganan	Validasi izin pada broker MQTT. Jika Anda memiliki broker MQTT dari AWS, lihat broker MQTT 3.1.1 (Moquette) dan broker MQTT 5 (EMQX).
File parket tidak dibuat	• Verifikasi bahwa topik MQTT sudah benar. • Verifikasi bahwa pesan MQTT dari sensor dalam format yang benar.
Objek tidak diunggah ke bucket S3	• Pastikan Anda memiliki konektivitas internet dan konektivitas titik akhir. • Verifikasi bahwa kebijakan sumber daya untuk bucket S3 Anda sudah benar. • Verifikasi izin untuk peran perangkat inti AWS IoT Greengrass Versi 2.

Sumber daya terkait

- DataFrame(Dokumentasi Panda)
- Dokumentasi Parket Apache (Dokumentasi parket)
- Kembangkan komponen AWS IoT Greengrass (Panduan Pengembang AWS IoT Greengrass, Versi 2)
- Menerapkan komponen AWS IoT Greengrass ke perangkat (Panduan Pengembang AWS IoT Greengrass, Versi 2)
- Berinteraksi dengan perangkat IoT lokal (Panduan Pengembang AWS IoT Greengrass, Versi 2)
- Broker MQTT 3.1.1 (Moquette) (Panduan Pengembang AWS IoT Greengrass, Versi 2)
- Broker MQTT 5 (EMQX) (Panduan Pengembang AWS IoT Greengrass, Versi 2)

Informasi tambahan

Analisis biaya

Skenario analisis biaya berikut menunjukkan bagaimana pendekatan konsumsi data yang tercakup dalam pola ini dapat memengaruhi biaya konsumsi data di AWS Cloud. Contoh harga dalam skenario ini didasarkan pada harga pada saat publikasi. Harga dapat berubah sewaktu-waktu. Selain itu, biaya Anda dapat bervariasi tergantung pada Wilayah AWS, kuota layanan AWS, dan faktor lain yang terkait dengan lingkungan cloud Anda.

Set sinyal masukan

Analisis ini menggunakan rangkaian sinyal input berikut sebagai dasar untuk membandingkan biaya konsumsi IoT dengan alternatif lain yang tersedia.

Jumlah sinyal	Frekuensi	Data per sinyal
125	25 Hz	8 byte

Dalam skenario ini, sistem menerima 125 sinyal. Setiap sinyal adalah 8 byte dan terjadi setiap 40 milidetik (25 Hz). Sinyal-sinyal ini bisa datang secara individual atau dikelompokkan dalam muatan umum. Anda memiliki opsi untuk membagi dan mengemas sinyal-sinyal ini berdasarkan kebutuhan Anda. Anda juga dapat menentukan latensi. Latensi terdiri dari periode waktu untuk menerima, mengumpulkan, dan menelan data.

Untuk tujuan perbandingan, operasi konsumsi untuk skenario ini berbasis di Wilayah `us-east-1` AWS. Perbandingan biaya hanya berlaku untuk layanan AWS. Biaya lain, seperti perangkat keras atau konektivitas, tidak diperhitungkan dalam analisis.

Perbandingan biaya

Tabel berikut menunjukkan biaya bulanan dalam dolar AS (USD) untuk setiap metode konsumsi.

Metode	Biaya bulanan
AWS IoT* SiteWise	331.77 USD

AWS IoT SiteWise Edge dengan paket pemrosesan data (menjaga semua data tetap di tepi)	200 USD
Aturan AWS IoT Core dan Amazon S3 untuk mengakses data mentah	84.54 USD
Kompresi file parket di tepi dan mengunggah ke Amazon S3	0.5 USD

*Data harus di-downsampled untuk memenuhi kuota layanan. Ini berarti ada beberapa kehilangan data dengan metode ini.

Metode alternatif

Bagian ini menunjukkan biaya yang setara untuk metode alternatif berikut:

- AWS IoT SiteWise — Setiap sinyal harus diunggah dalam pesan individual. Oleh karena itu, jumlah total pesan per bulan adalah 125 × 25 × 3600 × 24 × 30, atau 8,1 miliar pesan per bulan. Namun, AWS IoT hanya SiteWise dapat menangani 10 titik data per detik per properti. Dengan asumsi data di-downsampled menjadi 10 Hz, jumlah pesan per bulan dikurangi menjadi 125 × 10 × 3600 × 24 × 30, atau 3,24 miliar. Jika Anda menggunakan komponen penerbit yang mengemas pengukuran dalam kelompok 10 (dengan 1 USD per juta pesan), maka Anda mendapatkan biaya bulanan sebesar 324 USD per bulan. Dengan asumsi bahwa setiap pesan adalah 8 byte (1 Kb/125), itu adalah 25,92 Gb penyimpanan data. Ini menambah biaya bulanan 7,77 USD per bulan. Total biaya untuk bulan pertama adalah 331,77 USD dan meningkat sebesar 7,77 USD setiap bulan.
- AWS IoT SiteWise Edge dengan paket pemrosesan data, termasuk semua model dan sinyal yang diproses sepenuhnya di edge (yaitu, tidak ada konsumsi cloud) — Anda dapat menggunakan paket pemrosesan data sebagai alternatif untuk mengurangi biaya dan mengonfigurasi semua model yang dihitung di tepi. Ini dapat bekerja hanya untuk penyimpanan dan visualisasi, bahkan jika tidak ada perhitungan nyata yang dilakukan. Dalam hal ini, perlu menggunakan perangkat keras yang kuat untuk gateway tepi. Ada biaya tetap 200 USD per bulan.
- Konsumsi langsung ke AWS IoT Core oleh MQTT dan aturan IoT untuk menyimpan data mentah di Amazon S3 — Dengan asumsi semua sinyal dipublikasikan dalam muatan umum, jumlah total pesan yang dipublikasikan ke AWS IoT Core adalah 25 × 3600 × 24 × 30, atau 64,8 juta per bulan. Dengan 1 USD per juta pesan, itu biaya bulanan 64,8 USD per bulan. Pada 0,15 USD per juta aktivasi aturan dan dengan satu aturan per pesan, yang menambahkan biaya bulanan

sebesar 19,44 USD per bulan. Dengan biaya 0,023 USD per Gb penyimpanan di Amazon S3, yang menambahkan 1,5 USD per bulan (meningkat setiap bulan untuk mencerminkan data baru). Total biaya untuk bulan pertama adalah 84,54 USD dan meningkat 1,5 USD setiap bulan.

- Mengompresi data di tepi dalam file Parket dan mengunggah ke Amazon S3 (metode yang diusulkan) - Rasio kompresi tergantung pada jenis data. Dengan data industri yang sama diuji untuk MQTT, total data output selama sebulan penuh adalah 1,2 Gb. Biayanya 0,03 USD per bulan. Rasio kompresi (menggunakan data acak) yang dijelaskan dalam tolok ukur lain berada di urutan 66 persen (lebih dekat ke skenario terburuk). Total data adalah 21 Gb dan biaya 0,5 USD per bulan.

Generator file parket

Contoh kode berikut menunjukkan struktur generator file Parket yang ditulis dengan Python. Contoh kode hanya untuk tujuan ilustrasi dan tidak akan berfungsi jika ditempelkan ke lingkungan Anda.

```
import queue
import paho.mqtt.client as mqtt
import pandas as pd

#queue for decoupling the MQTT thread
messageQueue = queue.Queue()
client = mqtt.Client()
streammanager = StreamManagerClient()

def feederListener(topic, message):
    payload = {
        "topic" : topic,
        "payload" : message,
    }
    messageQueue.put_nowait(payload)

def on_connect(client_instance, userdata, flags, rc):
    client.subscribe("#",qos=0)

def on_message(client, userdata, message):
    feederListener(topic=str(message.topic),
 message=str(message.payload.decode("utf-8")))


filename = "tempfile.parquet"
streamname = "mystream"
```

```
destination_bucket= "amzn-s3-demo-bucket"
keyname="mykey"
period= 60

client.on_connect = on_connect
client.on_message = on_message
streammanager.create_message_stream(
            MessageStreamDefinition(name=streamname,
 strategy_on_full=StrategyOnFull.OverwriteOldestData)
        )


while True:
   try:
       message = messageQueue.get(timeout=myArgs.mqtt_timeout)
   except (queue.Empty):
       logger.warning("MQTT message reception timed out")

   currentTimestamp = getCurrentTime()
   if currentTimestamp >= nextUploadTimestamp:
       df = pd.DataFrame.from_dict(accumulator)
       df.to_parquet(filename)
       s3_export_task_definition = S3ExportTaskDefinition(input_url=filename,
 bucket=destination_bucket, key=key_name)
       streammanager.append_message(streamname,
 Util.validate_and_serialize_to_json_bytes(s3_export_task_definition))
       accumulator = {}
       nextUploadTimestamp += period
   else:
        accumulator.append(message)
```

Luncurkan pekerjaan Spark di cluster EMR sementara menggunakan fungsi Lambda

Dhrubajyoti Mukherjeee, Amazon Web Services

Ringkasan

Pola ini menggunakan aksi Amazon EMR RunJobFlow API untuk meluncurkan klaster sementara untuk menjalankan pekerjaan Spark dari fungsi Lambda. Cluster EMR sementara dirancang untuk dihentikan segera setelah pekerjaan selesai atau jika terjadi kesalahan. Cluster transien memberikan penghematan biaya karena hanya berjalan selama waktu komputasi, dan memberikan skalabilitas dan fleksibilitas dalam lingkungan cloud.

Cluster EMR transien diluncurkan menggunakan API Boto3 dan bahasa pemrograman Python dalam fungsi Lambda. Fungsi Lambda, yang ditulis dengan Python, memberikan fleksibilitas tambahan untuk memulai cluster saat dibutuhkan.

Untuk mendemonstrasikan komputasi dan output batch sampel, pola ini akan meluncurkan pekerjaan Spark di cluster EMR dari fungsi Lambda dan menjalankan komputasi batch terhadap contoh data penjualan perusahaan fiksi. Output dari pekerjaan Spark akan menjadi file nilai yang dipisahkan koma (CSV) di Amazon Simple Storage Service (Amazon S3). File data input, file Spark .jar, cuplikan kode, dan CloudFormation template AWS untuk peran virtual private cloud (VPC) dan AWS Identity and Access Management (IAM) untuk menjalankan komputasi disediakan sebagai lampiran.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif

Batasan

- Hanya satu pekerjaan Spark yang dapat dimulai dari kode pada satu waktu.

Versi produk

- Diuji di Amazon EMR 6.0.0

Arsitektur

Tumpukan teknologi target

- Amazon EMR
- AWS Lambda
- Amazon S3
- Apache Spark

Arsitektur target

Otomatisasi dan skala

Untuk mengotomatiskan komputasi batch Spark-EMR, Anda dapat menggunakan salah satu opsi berikut.

- Menerapkan EventBridge aturan Amazon yang dapat memulai fungsi Lambda dalam jadwal cron. Untuk informasi selengkapnya, lihat Tutorial: Menjadwalkan fungsi AWS Lambda menggunakan. EventBridge
- Konfigurasikan pemberitahuan peristiwa Amazon S3 untuk memulai fungsi Lambda saat kedatangan file.
- Teruskan parameter input ke fungsi AWS Lambda melalui badan peristiwa dan variabel lingkungan Lambda.

Alat

Layanan AWS

- Amazon EMR adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar di AWS untuk memproses dan menganalisis data dalam jumlah besar.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- Apache Spark adalah mesin analisis multi-bahasa untuk pemrosesan data skala besar.

Epik

Buat peran Amazon EMR dan Lambda IAM dan VPC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM dan VPC.	Jika Anda sudah memiliki peran AWS Lambda dan Amazon EMR IAM serta VPC, Anda dapat melewati langkah ini. Untuk menjalank an kode, baik cluster EMR dan fungsi Lambda memerlukan peran IAM. Cluster EMR juga membutuhkan VPC dengan subnet publik atau subnet pribadi dengan gateway NAT. Untuk membuat semua peran IAM dan VPC secara otomatis, gunakan templat CloudForm ation AWS terlampir apa adanya, atau Anda dapat membuat peran dan VPC secara manual seperti yang ditentukan di bagian Informasi tambahan.	Arsitek awan
Perhatikan kunci keluaran CloudFormation template AWS.	Setelah CloudFormation template berhasil diterapkan, navigasikan ke tab Output di konsol CloudFormation AWS. Perhatikan lima tombol output:	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `S3Bucket` • `LambdaExecutionRole` • `ServiceRole` • `JobFlowRole` • `Ec2SubnetId` Anda akan menggunakan nilai-nilai dari tombol-tombol ini saat Anda membuat fungsi Lambda.	

Unggah file Spark .jar

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file Spark .jar.	Unggah file Spark .jar ke bucket S3 yang dibuat oleh CloudFormation AWS stack. Nama bucket sama dengan tombol output`S3Bucket`.	AWS Umum

Buat fungsi Lambda untuk meluncurkan cluster EMR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda.	Di konsol Lambda, buat fungsi Lambda Python 3.9+ dengan peran eksekusi. Kebijakan peran eksekusi harus mengizinkan Lambda meluncurkan klaster EMR.	Insinyur data, insinyur Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(Lihat CloudFormation template AWS terlampir.)	
Salin dan tempel kodenya.	Ganti kode dalam `lambda_function.py` file dengan kode dari bagian Informasi tambahan dari pola ini.	Insinyur data, insinyur Cloud
Ubah parameter dalam kode.	Ikuti komentar dalam kode untuk mengubah nilai parameter agar sesuai dengan akun AWS Anda.	Insinyur data, insinyur Cloud
Luncurkan fungsi untuk memulai cluster.	Luncurkan fungsi untuk memulai pembuatan cluster EMR sementara dengan file Spark .jar yang disediakan. Ini akan menjalankan pekerjaan Spark dan berakhir secara otomatis ketika pekerjaan selesai.	Insinyur data, insinyur Cloud
Periksa status cluster EMR.	Setelah kluster EMR dimulai, ia muncul di konsol EMR Amazon di bawah tab Clusters. Kesalahan apa pun saat meluncurkan cluster atau menjalankan pekerjaan dapat diperiksa sesuai dengan itu.	Insinyur data, insinyur Cloud

Siapkan dan jalankan demo sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file Spark .jar.	Unduh file Spark .jar dari bagian Lampiran dan unggah ke bucket S3.	Insinyur data, insinyur Cloud
Unggah dataset input.	Unggah `fake_sales_data.csv` file terlampir ke bucket S3.	Insinyur data, insinyur Cloud
Tempel kode Lambda dan ubah parameternya.	Salin kode dari bagian Alat, dan tempel kode dalam fungsi Lambda, ganti file kode`lambda_function.py`. Ubah nilai parameter agar sesuai dengan akun Anda.	Insinyur data, insinyur Cloud
Luncurkan fungsi dan verifikasi outputnya.	Setelah fungsi Lambda memulai cluster dengan pekerjaan Spark yang disediakan, ia menghasilkan file.csv di bucket S3.	Insinyur data, insinyur Cloud

Sumber daya terkait

- Membangun Spark
- Apache Spark dan Amazon EMR
- Boto3 Dokumentasi run_job_flow Dokumen
- Informasi dan dokumentasi Apache Spark

Informasi tambahan

Kode

```
"""
Copy paste the following code in your Lambda function. Make sure to change the
 following key parameters for the API as per your account

-Name (Name of Spark cluster)
-LogUri (S3 bucket to store EMR logs)
-Ec2SubnetId (The subnet to launch the cluster into)
-JobFlowRole (Service role for EC2)
-ServiceRole (Service role for Amazon EMR)

The following parameters are additional parameters for the Spark job itself. Change the
 bucket name and prefix for the Spark job (located at the bottom).

-s3://your-bucket-name/prefix/lambda-emr/SparkProfitCalc.jar (Spark jar file)
-s3://your-bucket-name/prefix/fake_sales_data.csv (Input data file in S3)
-s3://your-bucket-name/prefix/outputs/report_1/ (Output location in S3)
"""
import boto3

client = boto3.client('emr')


def lambda_handler(event, context):
    response = client.run_job_flow(
        Name='spark_job_cluster',
        LogUri='s3://your-bucket-name/prefix/logs',
        ReleaseLabel='emr-6.0.0',
        Instances={
            'MasterInstanceType': 'm5.xlarge',
            'SlaveInstanceType': 'm5.large',
            'InstanceCount': 1,
            'KeepJobFlowAliveWhenNoSteps': False,
            'TerminationProtected': False,
            'Ec2SubnetId': 'subnet-XXXXXXXXXXXXXX'
        },
        Applications=[{'Name': 'Spark'}],
        Configurations=[
            {'Classification': 'spark-hive-site',
             'Properties': {
                 'hive.metastore.client.factory.class':
 'com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalogHiveClientFactory'}
             }
        ],
```

```
        VisibleToAllUsers=True,
        JobFlowRole='EMRLambda-EMREC2InstanceProfile-XXXXXXXXX',
        ServiceRole='EMRLambda-EMRRole-XXXXXXXXX',
        Steps=[
            {
                'Name': 'flow-log-analysis',
                'ActionOnFailure': 'TERMINATE_CLUSTER',
                'HadoopJarStep': {
                    'Jar': 'command-runner.jar',
                    'Args': [
                        'spark-submit',
                        '--deploy-mode', 'cluster',
                        '--executor-memory', '6G',
                        '--num-executors', '1',
                        '--executor-cores', '2',
                        '--class', 'com.aws.emr.ProfitCalc',
                        's3://your-bucket-name/prefix/lambda-emr/SparkProfitCalc.jar',
                        's3://your-bucket-name/prefix/fake_sales_data.csv',
                        's3://your-bucket-name/prefix/outputs/report_1/'
                    ]
                }
            }
        ]
    )
```

Peran IAM dan pembuatan VPC

Untuk meluncurkan cluster EMR dalam fungsi Lambda, diperlukan peran VPC dan IAM. Anda dapat mengatur peran VPC dan IAM dengan menggunakan CloudFormation templat AWS di bagian Lampiran pola ini, atau Anda dapat membuatnya secara manual menggunakan tautan berikut.

Peran IAM berikut diperlukan untuk menjalankan Lambda dan Amazon EMR.

Peran eksekusi Lambda

Peran eksekusi fungsi Lambda memberinya izin untuk mengakses layanan dan sumber daya AWS.

Peran layanan untuk Amazon EMR

Peran Amazon EMR mendefinisikan tindakan yang diizinkan untuk Amazon EMR saat menyediakan sumber daya dan melakukan tugas tingkat layanan yang tidak dilakukan dalam konteks instans Amazon Elastic Compute Cloud (Amazon) yang berjalan dalam klaster. EC2 Misalnya, peran layanan digunakan untuk menyediakan EC2 instance saat klaster diluncurkan.

Peran layanan untuk EC2 instance

Peran layanan untuk EC2 instance klaster (juga disebut profil EC2 instans untuk Amazon EMR) adalah jenis peran layanan khusus yang ditetapkan ke EC2 setiap instance dalam klaster EMR Amazon saat instance diluncurkan. Proses aplikasi yang berjalan di atas Apache Hadoop mengambil peran ini sebagai izin untuk berinteraksi dengan layanan AWS lainnya.

Pembuatan VPC dan subnet

Anda dapat membuat VPC dari konsol VPC.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Migrasikan beban kerja Apache Cassandra ke Amazon Keyspaces dengan menggunakan AWS Glue

Nikolai Kolesnikov, Karthiga Priya Chandran, dan Samir Patel, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memigrasikan beban kerja Apache Cassandra yang ada ke Amazon Keyspaces (untuk Apache Cassandra) dengan menggunakan AWS Glue. CQLReplicator Anda dapat menggunakan CQLReplicator AWS Glue untuk meminimalkan kelambatan replikasi migrasi beban kerja Anda hingga hitungan menit. Anda juga mempelajari cara menggunakan bucket Amazon Simple Storage Service (Amazon S3) untuk menyimpan data yang diperlukan untuk migrasi, termasuk file Apache Parquet, file konfigurasi, dan skrip. Pola ini mengasumsikan bahwa beban kerja Cassandra Anda di-host di instans Amazon Elastic Compute Cloud (EC2Amazon) di cloud pribadi virtual (VPC).

Prasyarat dan batasan

Prasyarat

- Cluster Cassandra dengan tabel sumber
- Tabel target di Amazon Keyspaces untuk mereplikasi beban kerja
- Bucket S3 untuk menyimpan file Parket perantara yang berisi perubahan data tambahan
- Bucket S3 untuk menyimpan file dan skrip konfigurasi pekerjaan

Batasan

- CQLReplicator pada AWS Glue memerlukan waktu untuk menyediakan Data Processing Units (DPUs) untuk beban kerja Cassandra. Jeda replikasi antara cluster Cassandra dan ruang kunci target dan tabel di Amazon Keyspaces kemungkinan akan berlangsung hanya dalam hitungan menit.

Arsitektur

Tumpukan teknologi sumber

- Apache Cassandra

- DataStax Peladen
- ScyllaDB

Tumpukan teknologi target

- Amazon Keyspaces

Arsitektur migrasi

Diagram berikut menunjukkan contoh arsitektur di mana cluster Cassandra di-host pada EC2 instance dan tersebar di tiga Availability Zone. Node Cassandra di-host di subnet pribadi.

Diagram menunjukkan alur kerja berikut:

1. Peran layanan khusus menyediakan akses ke Amazon Keyspaces dan bucket S3.
2. Pekerjaan AWS Glue membaca konfigurasi pekerjaan dan skrip di bucket S3.
3. Pekerjaan AWS Glue terhubung melalui port 9042 untuk membaca data dari cluster Cassandra.
4. Pekerjaan AWS Glue terhubung melalui port 9142 untuk menulis data ke Amazon Keyspaces.

Alat

Layanan dan alat AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS CloudShell adalah shell berbasis browser yang dapat Anda gunakan untuk mengelola layanan AWS dengan menggunakan AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) dan berbagai alat pengembangan yang telah diinstal sebelumnya.
- AWS Glue adalah layanan ETL yang dikelola sepenuhnya yang membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- Amazon Keyspaces (untuk Apache Cassandra) adalah layanan database terkelola yang membantu Anda memigrasi, menjalankan, dan menskalakan beban kerja Cassandra Anda di AWS Cloud.

Kode

Kode untuk pola ini tersedia di GitHub CQLReplicatorrepositori.

Praktik terbaik

- Untuk menentukan sumber daya AWS Glue yang diperlukan untuk migrasi, perkirakan jumlah baris dalam tabel sumber Cassandra. Misalnya, 250 K baris per 0,25 DPU (2 vCPUs, memori 4 GB) dengan disk 84 GB.
- Tabel Amazon Keyspaces pra-hangat sebelum dijalankan. CQLReplicator Misalnya, delapan CQLReplicator ubin (AWS Glue jobs) dapat menulis hingga 22 K WCUs per detik, sehingga target harus dipanaskan terlebih dahulu hingga 25-30 K WCUs per detik.
- Untuk mengaktifkan komunikasi antara komponen AWS Glue, gunakan aturan masuk referensi mandiri untuk semua port TCP di grup keamanan Anda.
- Gunakan strategi lalu lintas tambahan untuk mendistribusikan beban kerja migrasi dari waktu ke waktu.

Epik

Menyebarkan CQLReplicator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat keyspace target dan tabel.	1. Buat keyspace dan tabel di Amazon Keyspaces. Untuk informasi selengkapnya tentang kapasitas tulis, lihat Menulis perhitungan unit di bagian Informasi tambahan dari pola ini. Anda juga dapat membuat keyspace dengan menggunakan Cassandra Query Language (CQL). Untuk informasi selengkapnya, lihat Membuat ruang kunci menggunakan CQL di	Pemilik aplikasi, administrator AWS, DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bagian Informasi tambahan dari pola ini. **Note** Setelah Anda membuat tabel, pertimbangkan untuk mengalihkan tabel ke mode kapasitas sesuai permintaan untuk menghindari biaya yang tidak perlu. 2. Untuk memperbarui ke mode throughput, jalankan skrip berikut: `ALTER TABLE target_keyspace.target_table WITH CUSTOM_PROPERTIES = { 'capacity_mode': { 'throughput_mode': 'PAY_PER_REQUEST'} }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan driver Cassandra untuk terhubung ke Cassandra.	Gunakan skrip konfigurasi berikut: `Datastax-java-driver { basic.request.consistency = "LOCAL_QUORUM" basic.contact-points = ["127.0.0.1:9042"] advanced.reconnect-on-init = true basic.load-balancing-policy { local-datacenter = "datacenter1" } advanced.auth-provider = { class = PlainTextAuthProvider username = "user-at-sample" password = "S@MPLE=PASSWORD=" } }` **Note** Skrip sebelumnya menggunakan Spark Cassandra Connector. Untuk informasi lebih lanjut, lihat konfigurasi referensi untuk Cassandra.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan driver Cassandra untuk terhubung ke Amazon Keyspaces.	Gunakan skrip konfigurasi berikut: `datastax-java-driver { basic { load-balancing-pol icy { local-datacenter = us-west-2 } contact-points = ["cassandr a.us-west-2.amazon aws.com:9142"] request { page-size = 2500 timeout = 360 seconds consistency = LOCAL_QUORUM } } advanced { control-connection { timeout = 360 seconds } session-leak.thres hold = 6 connection { connect-timeout = 360 seconds init-query-timeout = 360 seconds warn-on-init-error = false } auth-provider = { class = software. aws.mcs.auth.SigV4 AuthProvider` ...	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> aws-region = us-
	west-2
	 }
	
	 ssl-engine-factory {
	  class = DefaultSs
	lEngineFactory
	  }
	  }
	}</pre>
	Note
	Skrip sebelumnya menggunakan Spark Cassandra Connector . Untuk informasi lebih lanjut, lihat konfigura si referensi untuk Cassandra.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk pekerjaan AWS Glue.	Buat peran layanan AWS baru yang `glue-cassandra-migration` diberi nama AWS Glue sebagai entitas tepercaya. **Note** `glue-cassandra-migration` Seharusnya menyediakan akses baca dan tulis ke bucket S3 dan Amazon Keyspaces. Bucket S3 berisi file.jar, file konfigurasi untuk Amazon Keyspaces dan Cassandra, dan file Parket perantara. Misalnya, berisi `AWSGlueServiceRole` `AmazonS3FullAccess`, dan kebijakan `AmazonKeyspacesFullAccess` terkelola.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Unduh CQLReplicator di AWS CloudShell.	Unduh proyek ke folder rumah Anda dengan menjalankan perintah berikut:
	`git clone https://github.com/aws-samples/cql-replicator.git`
	`cd cql-replicator/glue`
	`# Only for AWS CloudShell, the bc package includes bc and dc. Bc is an arbitrary precision numeric processing arithmetic language`
	`sudo yum install bc -y`	

Ubah file konfigurasi referensi.	Salin `Cassandra Connector.conf` dan `KeyspacesConnector.conf` ke `../glue/conf` direktori di folder proyek.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Memulai proses migrasi.	Perintah berikut menginisi alisasi CQLReplicator lingkungan. Inisialisasi melibatkan penyalinan artefak.jar, dan membuat konektor AWS Glue, bucket S3, pekerjaan AWS Glue, keyspace, dan tabel: `migration ledger`
	
	`cd cql-replicator/glue/bin`
	`./cqlreplicator --state init --sg '"sg-1","sg-2"' \ --subnet "subnet-XXXXXXXXXXXX" \ --az us-west-2a --region us-west-2 \ --glue-iam-role glue-cassandra-migration \ --landing-zone s3://cql-replicator-1234567890-us-west-2`
	
	Skrip mencakup parameter berikut:
	
	• `--sg`— Grup keamanan yang mengizinkan akses ke cluster Cassandra dari AWS Glue dan menyertakan aturan masuk referensi
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mandiri untuk semua lalu lintas • `--subnet`— Subnet tempat cluster Cassandra berada • `--az`— Availability Zone dari subnet • `--region`— Wilayah AWS tempat cluster Cassandra digunakan • `--glue-iam-role` — Izin peran IAM yang dapat diasumsikan AWS Glue saat memanggil Amazon Keyspaces dan Amazon S3 atas nama Anda • `--landing zone`— Parameter opsional untuk menggunakan kembali bucket S3 (Jika Anda tidak memberikan nilai untuk `--landing zone` parameter, `init` proses akan mencoba membuat bucket baru untuk menyimpan file konfigura si, artefak .jar, dan file perantara.)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan.	Setelah Anda menjalank an perintah sebelumnya, akun AWS harus berisi yang berikut: • Pekerjaan CQLReplicator AWS Glue dan konektor AWS Glue di AWS Glue • Ember S3 yang menyimpan artefak • Ruang kunci target `migration` dan `ledger` tabel di Amazon Keyspaces	AWS DevOps

Jalankan CQLReplicator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai proses migrasi.	Untuk beroperasi CQLReplic ator pada AWS Glue, Anda perlu menggunakan `--state run` perintah, diikuti oleh serangkaian parameter . Konfigurasi yang tepat dari parameter ini terutama ditentukan oleh persyaratan migrasi unik Anda. Misalnya, pengaturan ini mungkin berbeda jika Anda memilih untuk mereplikasi nilai dan pembaruan time to live (TTL), atau Anda membongkar objek melebihi 1 MB ke Amazon S3.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk mereplikasi beban kerja dari cluster Cassandra ke Amazon Keyspaces, jalankan perintah berikut: `./cqlreplicator --state run --tiles 8 \ --landing-zone s3://cql-replicator-1234567890-us-west-2 \ --region us-west-2 \ --src-keyspace source_keyspace \ --src-table source_table \ --trg-keyspace taget_keyspace \ --writetime-column column_name \ --trg-table target_table --inc-traffic` Ruang kunci dan tabel sumber Anda ada `source_keyspace.source_table` di cluster Cassandra. Ruang kunci dan tabel target Anda ada `target_keyspace.target_table` di Amazon Keyspaces. Parameter ini `--inc-traffic`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membantu mencegah lalu lintas tambahan membebani cluster Cassandra dan Amazon Keyspaces dengan jumlah permintaan yang tinggi. Untuk mereplikasi pembaruan, tambahkan `--writetime-column regular_column_name` ke baris perintah Anda. Kolom reguler akan digunakan sebagai sumber stempel waktu tulis.	

Memantau proses migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi baris Cassandra yang dimigrasi selama fase migrasi historis.	Untuk mendapatkan jumlah baris yang direplikasi selama fase penimbunan ulang, jalankan perintah berikut: `./cqlreplicator --state stats \ --landing-zone s3://cql-replicator-1234567890-us-west-2 \ --src-keyspace source_keyspace --src-table source_table --region us-west-2`	AWS DevOps

Hentikan proses migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan `cqlreplicator` perintah atau konsol AWS Glue.	Untuk menghentikan proses migrasi dengan anggun, jalankan perintah berikut: `./cqlreplicator --state request-stop --tiles 8 \ --landing-zone s3://cql-replicator-1234567890-us-west-2 \ --region us-west-2 \ --src-keyspace source_keyspace --src-table source_table` Untuk segera menghenti kan proses migrasi, gunakan konsol AWS Glue.	AWS DevOps

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya yang digunakan.	Perintah berikut akan menghapus tabel AWS Glue job, konektor, bucket S3, dan `ledger` Keyspaces: `./cqlreplicator --state cleanup --landing-zone`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`s3://cql-replicator-1234567890-us-west-2`	

Pemecahan masalah

Isu	Solusi
Pekerjaan AWS Glue gagal dan mengembalikan kesalahan Kehabisan Memori (OOM).	1. Ubah jenis pekerja (naik skala). Misalnya, ubah `G0.25X` ke `G.1X` atau `G.1X` ke`G.2X`. Atau, tingkatkan jumlah DPUs per pekerjaan AWS Glue (skala keluar) di CQLReplicator. 2. Mulai proses migrasi dari titik di mana itu terputus. Untuk memulai ulang CQLReplicator pekerjaan yang gagal, jalankan kembali `--state run` perintah dengan parameter yang sama.

Sumber daya terkait

- CQLReplicator dengan AWS Glue README.MD
- Dokumentasi AWS Glue
- Dokumentasi Amazon Keyspaces
- Apache Cassandra

Informasi tambahan

Pertimbangan migrasi

Anda dapat menggunakan AWS Glue untuk memigrasikan beban kerja Cassandra Anda ke Amazon Keyspaces, sambil menjaga database sumber Cassandra Anda berfungsi sepenuhnya selama proses migrasi. Setelah replikasi selesai, Anda dapat memilih untuk memotong aplikasi Anda ke Amazon Keyspaces dengan jeda replikasi minimal (kurang dari menit) antara cluster Cassandra

dan Amazon Keyspaces. Untuk menjaga konsistensi data, Anda juga dapat menggunakan pipeline serupa untuk mereplikasi data kembali ke cluster Cassandra dari Amazon Keyspaces.

Tulis perhitungan unit

Sebagai contoh, pertimbangkan bahwa Anda bermaksud menulis 500.000.000 dengan ukuran baris 1 KiB selama satu jam. Jumlah total unit tulis Amazon Keyspaces (WCUs) yang Anda butuhkan didasarkan pada perhitungan ini:

```
(number of rows/60 mins 60s) 1 WCU per row = (500,000,000/(60*60s) * 1 WCU)
= 69,444 WCUs required
```

69.444 WCUs per detik adalah tarif untuk 1 jam, tetapi Anda dapat menambahkan beberapa bantal untuk overhead. Misalnya, `69,444 * 1.10 = 76,388 WCUs` memiliki overhead 10 persen.

Buat keyspace dengan menggunakan CQL

Untuk membuat keyspace dengan menggunakan CQL, jalankan perintah berikut:

```
CREATE KEYSPACE target_keyspace WITH replication = {'class': 'SingleRegionStrategy'}
CREATE TABLE target_keyspace.target_table ( userid uuid, level text, gameid int,
 description text, nickname text, zip text, email text, updatetime text, PRIMARY KEY
 (userid, level, gameid) ) WITH default_time_to_live = 0 AND CUSTOM_PROPERTIES =
 {'capacity_mode':{ 'throughput_mode':'PROVISIONED', 'write_capacity_units':76388,
 'read_capacity_units':3612 }} AND CLUSTERING ORDER BY (level ASC, gameid ASC)
```

Migrasi data Hadoop ke Amazon S3 menggunakan Migrator WANdisco LiveData

Tony Velcich, Amazon Web Services

Ringkasan

Migrasikan data Hadoop dari HDFS ke Amazon S3 menggunakan WANdisco LiveData Migrator untuk mengotomatiskan proses migrasi data.

Prasyarat dan batasan

Prasyarat

- Hadoop cluster edge node tempat LiveData Migrator akan diinstal. Node harus memenuhi persyaratan berikut:
  - Spesifikasi minimum: 4 CPUs, 16 GB RAM, penyimpanan 100 GB.
  - Jaringan minimum 2 Gbps.
  - Port 8081 dapat diakses di simpul tepi Anda untuk mengakses WANdisco UI.
  - Java 1.8 64-bit.
  - Pustaka klien Hadoop diinstal pada simpul tepi.
  - Kemampuan untuk mengautentikasi sebagai superuser HDFS (misalnya, “hdfs”).
  - Jika Kerberos diaktifkan di cluster Hadoop Anda, tab kunci yang valid yang berisi prinsipal yang sesuai untuk superuser HDFS harus tersedia di simpul tepi.
- Akun AWS aktif dengan akses ke bucket S3.
- Tautan AWS Direct Connect yang dibuat antara klaster Hadoop lokal Anda (khususnya simpul tepi) dan AWS.

Versi produk

- LiveData Migrator 1.8.6
- WANdisco UI (OneUI) 5.8.0

Arsitektur

Tumpukan teknologi sumber

- Cluster Hadoop lokal

Tumpukan teknologi target

- Amazon S3

Arsitektur

Diagram berikut menunjukkan arsitektur solusi LiveData Migrator.

Alur kerja terdiri dari empat komponen utama untuk migrasi data dari HDFS lokal ke Amazon S3.

- LiveData Migrator — Mengotomatiskan migrasi data dari HDFS ke Amazon S3, dan berada di simpul tepi cluster Hadoop.
- HDFS — Sistem file terdistribusi yang menyediakan akses throughput tinggi ke data aplikasi.
- Amazon S3 — Layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja.
- AWS Direct Connect — Layanan yang membuat koneksi jaringan khusus dari pusat data lokal Anda ke AWS.

Otomatisasi dan skala

Anda biasanya akan membuat beberapa migrasi sehingga Anda dapat memilih konten tertentu dari sistem file sumber Anda berdasarkan jalur atau direktori. Anda juga dapat memigrasikan data ke beberapa sistem file independen secara bersamaan dengan menentukan beberapa sumber daya migrasi.

Epik

Konfigurasikan penyimpanan Amazon S3 di akun AWS Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke akun AWS Anda.	Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/	Pengalaman AWS
Buat ember S3.	Jika Anda belum memiliki bucket S3 untuk digunakan sebagai penyimpanan target, pilih opsi “Buat bucket” di konsol Amazon S3, dan tentukan nama bucket, Wilayah AWS, dan pengaturan bucket untuk memblokir akses publik. AWS dan WANdisco merekomendasikan agar Anda mengaktifkan opsi blokir akses publik untuk bucket S3, dan menyiapkan kebijakan akses bucket dan izin pengguna untuk memenuhi persyaratan organisasi Anda. Contoh AWS disediakan di https://docs.aws.amazon.com/AmazonS3/latest/dev/example-walkthroughs-managing-access-example1.html.	Pengalaman AWS

Instal LiveData Migrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh penginstal LiveData Migrator.	Unduh penginstal LiveData Migrator dan unggah ke simpul tepi Hadoop. Anda dapat mengunduh uji coba gratis LiveData Migrator di https://www2.wandisco.com/ldm-trial. You can also obtain access to LiveData Migrator from AWS Marketplace, at https:/ /aws.amazon. com/ marketplace/pp/B07B8SZND9.	Administrator Hadoop, Pemilik aplikasi
Instal LiveData Migrator.	Gunakan installer yang diunduh dan instal LiveData Migrator sebagai superuser HDFS pada node tepi di cluster Hadoop Anda. Lihat bagian “Informasi tambahan” untuk perintah instalasi.	Administrator Hadoop, Pemilik aplikasi
Periksa status LiveData Migrator dan layanan lainnya.	Periksa status LiveData Migrator, Hive migrator, dan WANdisco UI dengan menggunakan perintah yang disediakan di bagian “Informasi tambahan”.	Administrator Hadoop, Pemilik aplikasi

Konfigurasikan penyimpanan melalui WANdisco UI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Daftarkan akun LiveData Migrator Anda.	Masuk ke WANdisco UI melalui browser web pada	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	port 8081 (pada simpul tepi Hadoop) dan berikan detail Anda untuk pendaftaran. Misalnya, jika Anda menjalankan LiveData Migrator pada host bernama myldmhost.example.com, URL-nya adalah: http://myldmhost.example.com:8081	
Konfigurasikan penyimpanan HDFS sumber Anda.	Berikan detail konfigurasi yang diperlukan untuk penyimpanan HDFS sumber Anda. Ini akan mencakup nilai “fs.defaultFS” dan nama penyimpanan yang ditentukan pengguna. Jika Kerberos diaktifkan, berikan lokasi utama dan tab tombol untuk digunakan LiveData Migrator. Jika NameNode HA diaktifkan di cluster, berikan jalur ke file core-site.xml dan hdfs-site.xml di simpul tepi.	Administrator Hadoop, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan penyimpanan Amazon S3 target Anda.	Tambahkan penyimpan an target Anda sebagai tipe S3a. Berikan nama penyimpanan yang ditentuka n pengguna dan nama bucket S3. Masukkan “org.apache.hadoop.fs.s3a.S imple AWSCredentials Provider” untuk opsi Penyedia Kredensial, dan berikan akses AWS dan kunci rahasia untuk bucket S3. Properti S3a tambahan juga akan dibutuhka n. Untuk detailnya, lihat bagian “Properti S3a” di dokumenta si LiveData Migrator di docs/ command-reference/# 3a. https://docs.wandisco.com/l ive-data-migrator/ filesystem-add-s	AWS, Pemilik aplikasi

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan pengecualian (jika diperlukan).	Jika Anda ingin mengecual ikan kumpulan data tertentu dari migrasi, tambahkan pengecualian untuk penyimpanan HDFS sumber. Pengecualian ini dapat didasarkan pada ukuran file, nama file (berdasarkan pola regex), dan tanggal modifikasi.	Administrator Hadoop, Pemilik aplikasi

Membuat dan memulai migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan migrasi.	Buat migrasi di dasbor WANdisco UI. Pilih sumber Anda (HDFS) dan target (ember S3). Tambahkan pengecualian baru yang telah Anda tentukan di langkah sebelumnya. Pilih opsi “Timpa” atau “Lewati jika Ukuran Cocokkan”. Buat migrasi saat semua bidang selesai.	Administrator Hadoop, Pemilik aplikasi
Mulai migrasi.	Di dasbor, pilih migrasi yang Anda buat. Klik untuk memulai migrasi. Anda juga dapat memulai migrasi secara otomatis dengan memilih opsi mulai otomatis saat Anda membuat migrasi.	Pemilik aplikasi

Kelola bandwidth (opsional)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan batas bandwidth jaringan antara sumber dan target.	Dalam daftar Penyimpanan di dasbor, pilih penyimpan an sumber Anda dan pilih “Manajemen Bandwidth” di daftar Pengelompokan. Hapus opsi tak terbatas, dan tentukan batas bandwidth maksimum dan unit. Pilih “Terapkan.”	Pemilik aplikasi, Jaringan

Pantau dan kelola migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Melihat informasi migrasi menggunakan WANdisco UI.	Gunakan WANdisco UI untuk melihat informasi lisensi, bandwidth, penyimpan an, dan migrasi. UI juga menyediakan sistem notifikas i sehingga Anda dapat menerima pemberitahuan tentang kesalahan, peringata n, atau tonggak penting dalam penggunaan Anda.	Administrator Hadoop, Pemilik aplikasi
Hentikan, lanjutkan, dan hapus migrasi.	Anda dapat menghentikan migrasi dari mentransfer konten ke targetnya dengan menempatkannya dalam status STOPTED. Migrasi yang dihentikan dapat dilanjutkan. Migrasi dalam status STOPTED juga dapat dihapus.	Administrator Hadoop, Pemilik aplikasi

Sumber daya terkait

- LiveData Dokumentasi migrator
- LiveData Migrator di AWS Marketplace

Informasi tambahan

Instalasi LiveData Migrator

Anda dapat menggunakan perintah berikut untuk menginstal LiveData Migrator, dengan asumsi bahwa installer ada di dalam direktori kerja Anda:

```
su - hdfs
chmod +x livedata-migrator.sh && sudo ./livedata-migrator.sh
```

Memeriksa status LiveData Migrator dan layanan lainnya setelah instalasi

Gunakan perintah berikut untuk memeriksa status LiveData Migrator, Hive migrator, dan UI: WANdisco

```
service livedata-migrator status
service hivemigrator status
service livedata-ui status
```

Migrasikan Oracle Business Intelligence 12c ke AWS Cloud dari server lokal

Pertunjukan Lanre (Lan-Ray) dan Patrick Huang, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memigrasikan Oracle Business Intelligence Enterprise Edition 12c dari server lokal ke AWS Cloud dengan menggunakan AWS. CloudFormation Ini juga menjelaskan bagaimana Anda dapat menggunakan layanan AWS lainnya untuk mengimplementasikan komponen Oracle BI 12c yang memberikan ketersediaan tinggi, keamanan, fleksibilitas, dan kemampuan untuk menskalakan secara dinamis.

Untuk daftar praktik terbaik terkait migrasi Oracle BI 12c ke AWS Cloud, lihat bagian Informasi tambahan dari pola ini.

> **Note**
>
> Ini adalah praktik terbaik untuk menjalankan beberapa migrasi pengujian sebelum mentransfer data Oracle BI 12c Anda yang ada ke cloud. Pengujian ini membantu Anda menyempurnakan pendekatan migrasi, mengidentifikasi dan memperbaiki potensi masalah, dan memperkirakan persyaratan waktu henti dengan lebih akurat.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Mengamankan konektivitas jaringan antara server lokal dan AWS melalui layanan AWS Virtual Private Network (AWS VPN) atau AWS Direct Connect
- Lisensi perangkat lunak untuk sistem operasi Oracle Anda, Oracle BI 12c, Oracle Database, Oracle Server, dan Oracle HTTP Server WebLogic

Batasan

Untuk informasi tentang batas ukuran penyimpanan, lihat dokumentasi Amazon Relational Database Service (Amazon RDS) untuk Oracle.

Versi produk

- Oracle Business Intelligence Enterprise Edisi 12c
- WebLogic Server Oracle 12c
- Oracle HTTP Server 12c
- Oracle Database 12c (atau yang lebih baru)
- Oracle Java SE 8

Arsitektur

Diagram berikut menunjukkan contoh arsitektur untuk menjalankan komponen Oracle BI 12c di AWS Cloud:

Diagram ini menunjukkan arsitektur berikut:

1. Amazon Route 53 menyediakan konfigurasi layanan nama domain (DNS).
2. Elastic Load Balancing (ELB) mendistribusikan lalu lintas jaringan untuk meningkatkan skalabilitas dan ketersediaan komponen Oracle BI 12c di beberapa Availability Zone.
3. Grup Auto Scaling Amazon Elastic Compute Cloud (Amazon EC2) meng-host Server HTTP Oracle, server Admin Weblogic, dan server BI terkelola di beberapa Availability Zone.
4. Amazon Relational Database Service (Amazon RDS) untuk database Oracle menyimpan metadata BI Server di beberapa Availability Zone.
5. Amazon Elastic File System (Amazon EFS) dipasang ke setiap komponen Oracle BI 12c untuk penyimpanan file bersama.

Tumpukan teknologi

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Elastic File System (Amazon EFS)
- Amazon RDS for Oracle
- AWS Certificate Manager (ACM)

- Elastic Load Balancing (ELB)
- Oracle BI 12c
- WebLogic Server Oracle 12c
- Server HTTP Oracle (OHS)

Alat

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS Certificate Manager (ACM) membantu Anda membuat, menyimpan, dan memperbarui sertifikat dan kunci SSL/TLS X.509 publik dan pribadi yang melindungi situs web dan aplikasi AWS Anda.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan, dan dengan cepat meningkatkannya ke atas atau ke bawah.
- EC2 Auto Scaling Amazon membantu Anda menjaga ketersediaan aplikasi dan memungkinkan Anda menambahkan atau menghapus EC2 instans Amazon secara otomatis sesuai dengan kondisi yang Anda tentukan.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di AWS Cloud.
- Elastic Load Balancing mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

- Oracle Data Pump membantu Anda memindahkan data dan metadata dari satu database ke database lainnya dengan kecepatan tinggi.
- Oracle Fusion Middleware adalah rangkaian alat pengembangan aplikasi dan solusi integrasi untuk manajemen identitas, kolaborasi, dan pelaporan intelijen bisnis.
- Oracle GoldenGate membantu Anda merancang, menjalankan, mengatur, dan memantau replikasi data dan mengalirkan solusi pemrosesan data di Oracle Cloud Infrastructure.
- Oracle WebLogic Scripting Tool (WLST) menyediakan antarmuka baris perintah yang membantu Anda secara horizontal skala cluster Anda. WebLogic

Epik

Menilai lingkungan sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan informasi inventaris perangkat lunak.	Identifikasi versi dan tingkat patch untuk setiap komponen perangkat lunak tumpukan teknologi sumber Anda, termasuk yang berikut ini: • Sistem operasi Oracle • Oracle Database • Oracle BI 12c • Server Oracle WebLogic • Server HTTP Oracle • Java	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI
Kumpulkan informasi inventaris komputasi dan penyimpanan.	Di lingkungan sumber Anda, tinjau metrik pemanfaatan saat ini dan historis untuk hal-hal berikut: • Penggunaan CPU • Penggunaan memori	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI, Administrator Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Penggunaan penyimpanan **Important** Pastikan Anda memperhitungkan lonjakan historis dalam penggunaan.	
Kumpulkan informasi tentang arsitektur lingkungan sumber dan persyaratannya.	Dapatkan pemahaman penuh tentang arsitektur lingkungan sumber Anda dan persyarat annya, termasuk pengetahuan tentang hal-hal berikut: • Konfigurasi domain Oracle WebLogic Server • Pengelompokan • Penyeimbangan beban • Konektivitas • Ketersediaan • Persyaratan pemulihan bencana	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI
Identifikasi sumber data Java Database Connectivity (JDBC)	Kumpulkan informasi tentang sumber data JDBC lingkungan sumber Anda dan driver untuk setiap mesin database yang digunakannya.	Arsitek Migrasi, Pemilik Aplikasi, Administrator Oracle BI, Insinyur Database atau Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan informasi tentang pengaturan khusus lingkungan.	Kumpulkan informasi tentang pengaturan dan konfigurasi yang khusus untuk lingkungan sumber Anda, termasuk yang berikut ini: • Skrip startup dan shutdown khusus • Java dan variabel lingkungan lainnya • Sertifikat	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI
Identifikasi dependensi apa pun pada aplikasi lain.	Kumpulkan informasi tentang integrasi di lingkungan sumber Anda yang membuat dependensi dengan aplikasi lain. **Important** Pastikan Anda mengidentifikasi integrasi Lightweight Directory Access Protocol (LDAP) dan persyaratan jaringan lainnya.	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI

Rancang lingkungan target Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dokumen desain tingkat tinggi.	Buat dokumen desain arsitektur target. Pastikan Anda menggunakan informasi yang Anda kumpulkan saat menilai lingkungan sumber Anda untuk menginformasikan dokumen desain.	Arsitek Solusi, Arsitek Aplikasi, Insinyur Basis Data, Arsitek Migrasi
Dapatkan persetujuan untuk dokumen desain.	Tinjau dokumen desain dengan pemangku kepentingan dan dapatkan persetujuan yang diperlukan.	Pemilik Aplikasi atau Layanan, Arsitek Solusi, Arsitek Aplikasi

Menyebarkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kode infrastruktur di CloudFormation.	Buat CloudFormation template untuk menyediakan infrastruktur Oracle BI 12c Anda di AWS Cloud. Untuk informasi selengkapnya, lihat Bekerja dengan CloudFormation templat AWS di Panduan CloudFormation Pengguna AWS. **Note** Ini adalah praktik terbaik untuk membuat CloudFormation	Achitect Infrastruktur Cloud, Arsitek Solusi, Arsitek Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	template modular untuk setiap tingkat Oracle BI 12c, daripada satu template besar untuk semua sumber daya Anda. Untuk informasi selengkapnya tentang praktik CloudForm ation terbaik, lihat 8 praktik terbaik saat mengotoma tiskan penerapan Anda dengan AWS CloudFormation di Blog AWS.	
Unduh perangkat lunak yang diperlukan.	DDownload perangkat lunak berikut bersama dengan versi dan tambalan yang diperlukan dari situs web Oracle: • Jawa JDK8 • WebLogic Server Oracle 12c • Oracle BI 12c	Arsitek Migrasi, Insinyur Basis Data, Arsitek Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan skrip instalasi.	Buat skrip instalasi perangkat lunak yang menjalankan instalasi diam. Skrip ini menyederhanakan otomatisasi penerapan. Untuk informasi lebih lanjut, lihat OBIEE 12c: Bagaimana Melakukan Instalasi Senyap? di situs Dukungan Oracle. Anda memerlukan akun Oracle Support untuk melihat dokumentasi.	Arsitek Migrasi, Insinyur Basis Data, Arsitek Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AMI Linux yang didukung Amazon EBS untuk tingkatan web dan aplikasi Anda.	1. Terapkan dan konfigurasikan EC2 instans Amazon untuk tingkatan web dan aplikasi Anda. Pastikan bahwa instance memenuhi prasyarat untuk menjalankan yang berikut ini: • Pengaturan lingkungan sistem operasi Oracle • Pengaturan akun pengguna sistem operasi Oracle • Instalasi perangkat lunak Java 2. Buat Amazon Machine Images (AMIs) dari instans dan simpan salinan untuk digunakan di masa mendatang. Untuk petunjuk, lihat Membuat AMI Linux yang didukung Amazon EBS di Panduan EC2 Pengguna Amazon untuk Instans Linux.	Arsitek Migrasi, Insinyur Basis Data, Arsitek Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan infrastruktur AWS Anda dengan menggunakan CloudFormation.	Terapkan tingkatan web dan aplikasi Oracle BI 12c Anda dalam modul dengan menggunakan CloudForm ation templat yang Anda buat. Untuk petunjuknya, lihat Memulai AWS CloudForm ation di Panduan CloudForm ation Pengguna AWS.	Arsitek Infrastruktur Cloud, Arsitek Solusi, Arsitek Aplikasi

Migrasikan Oracle BI 12c ke AWS dengan menggunakan instalasi baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tahap perangkat lunak yang diperlukan.	Tingkatkan perangkat lunak yang diperlukan di lokasi yang dapat diakses oleh EC2 instans Amazon. Misalnya, Anda dapat menampilkan perangkat lunak di Amazon S3 atau EC2 contoh Amazon lain yang dapat diakses oleh server web dan aplikasi Anda.	Arsitek Migrasi, Arsitek Oracle BI, Achitect Infrastruktur Cloud, Arsitek Solusi, Arsitek Aplikasi
Siapkan database repositori Anda untuk instalasi Oracle BI 12c.	Buat skema Oracle BI 12c dengan menjalankan Oracle Repository Creation Utility (RCU) terhadap instance database Amazon RDS for Oracle baru.	Achitect Infrastruktur Cloud, Arsitek Solusi, Arsitek Aplikasi, Arsitek Migrasi, Arsitek Oracle BI
Instal Oracle Fusion Middleware 12c dan Oracle BI 12c.	1. Dimulai dengan satu EC2 instans Amazon, instal infrastruktur Oracle Fusion	Arsitek Migrasi, Arsitek Oracle BI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Middleware 12c dan OBIEE 12c. Untuk informasi selengkapnya, lihat bagian berikut dari Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intellige nce: • Memulai pemasang infrastruktur pada BIHOST1 • Menginstal Oracle Business Intelligence dalam persiapan untuk penyebaran perusahaan **Note** Gunakan Amazon EFS untuk meng-host direktori yang akan dibagikan di antara node cluster Oracle BI 12c. 2. Terapkan tambalan yang diperlukan ke instalasi. 3. Buat AMIs instance dan simpan salinan untuk digunakan di masa mendatang.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan domain Oracle WebLogic Server Anda untuk Oracle BI 12c.	Konfigurasikan domain Oracle BI 12c Anda sebagai penerapan yang tidak berkerumun. Untuk informasi selengkapnya, lihat Mengonfigurasi Domain BI di Panduan Penyebaran Perusahaan Oracle Fusion Middleware untuk Oracle Business Intelligence.	Arsitek Migrasi, Arsitek Oracle BI
Lakukan skala horizontal dari Oracle BI 12c.	Secara horizontal skala node tunggal ke jumlah node yang diinginkan. Untuk informasi selengkap nya, lihat Menskalakan Oracle Business Intelligence di Oracle Fusion Middleware Enterpris e Deployment Guide untuk Oracle Business Intelligence.	Arsitek Migrasi, Arsitek Oracle BI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Oracle HTTP Server 12c.	1. Instal Oracle HTTP Server 12c di instans Amazon tingkat web Oracle. EC2 Untuk petunjuk lihat Instal Oracle HTTP Server 12c di Instal dan konfigurasi Oracle HTTP Server untuk Oracle Access Managemen t 12c. 2. Terapkan tambalan yang diperlukan ke instalasi. 3. Buat AMIs instance dan simpan salinan untuk digunakan di masa mendatang.	Arsitek Migrasi, Arsitek Oracle BI
Konfigurasikan penyeimba ng beban untuk penghentian SSL.	1. Buat atau impor sertifikat SSL di ACM. 2. Kaitkan sertifikat SSL dengan ELB.	Arsitek Infrastruktur Cloud, Arsitek Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan artefak metadata intelijen bisnis ke AWS.	1. Ekspor file Oracle Business Intelligence Application Archive (BAR) dari instalasi Oracle BI 12c lokal. Untuk mengekspor file BAR, gunakan WebLogic Scripting Tool (WLST) untuk menjalankan perintah. exportServiceInstance 2. Impor file BAR lokal ke instalasi AWS Oracle BI 12c. Untuk mengimpor file BAR, jalankan perintah importServiceInstanceWLST.	Arsitek Migrasi, Arsitek Oracle BI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan tugas pasca-migrasi.	Setelah mengimpor file BAR, lakukan hal berikut: • Konfigurasikan sumber data JDBC tambahan apa pun. • Instal driver untuk sumber data lain seperti PostgreSQL, atau Amazon Redshift. • Konfigurasikan Oracle LDAP, SSL, single sign-on (SSO), dan toko keamanan. WebLogic • Konfigurasikan kebijakan AWS Identity and Access Management (IAM). • Aktifkan pelacakan penggunaan. • Mengatur integrasi ke sistem lain. • Migrasikan skrip kustom apa pun.	Arsitek Migrasi, Arsitek Oracle BI

Uji lingkungan baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji lingkungan Oracle BI 12c yang baru.	Lakukan end-to-end pengujian pada lingkungan Oracle BI 12c yang baru. Gunakan otomatisasi sebanyak mungkin. Contoh kegiatan pengujian meliputi:	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Memvalidasi dasbor, laporan, dan URLs • Pengujian penerimaan pengguna (UAT) • Pengujian penerimaan operasional (OAT) **Note** Lakukan pengujian dan validasi tambahan sesuai kebutuhan.	

Potong ke lingkungan baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Putuskan sambungan lalu lintas ke lingkungan Oracle BI 12c lokal.	Pada jendela cutover yang ditunjuk, hentikan semua lalu lintas ke lingkungan Oracle BI 12c lokal.	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI
Sinkronisasi ulang database repositori Oracle BI 12c baru dengan database sumber.	Sinkronisasi ulang database repositori Amazon RDS Oracle Oracle BI 12c dengan database lokal. Untuk menyinkronkan database, Anda dapat menggunakan penyegaran Oracle Data Pump atau AWS DMS change data capture (CDC).	Administrator Oracle BI, Insinyur Databasa/Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan Oracle BI 12c Anda URLs untuk menunjuk ke lingkungan AWS baru.	Perbarui Oracle BI 12c URLs di server DNS internal Anda sehingga mengarah ke instalasi AWS baru.	Arsitek Migrasi, Arsitek Solusi, Pemilik Aplikasi, Administrator Oracle BI
Pantau lingkungan baru.	Pantau lingkungan Oracle BI 12c baru dengan menggunakan salah satu alat berikut: • Amazon CloudWatch • Wawasan Performa Amazon RDS • Manajer Perusahaan Oracle	Administrator Oracle BI, Insinyur Databasa/Administrator, Administrator Aplikasi
Dapatkan sign-off pada proyek.	Tinjau hasil pengujian dengan pemangku kepentingan dan dapatkan persetujuan yang diperlukan untuk menyelesaikan migrasi.	Pemilik Aplikasi, Pemilik Layanan, Arsitek Infrastruktur Cloud, Arsitek Migrasi, Arsitek Oracle BI

Sumber daya terkait

- Menggunakan Utilitas Pembuatan Repositori Oracle di RDS untuk Oracle (Panduan Pengguna Amazon RDS)
- Oracle di Amazon RDS (Panduan Pengguna Amazon RDS)
- Oracle WebLogic Server 12c di AWS (whitepaper AWS)
- Menyebarkan Oracle Business Intelligence untuk ketersediaan tinggi (Oracle Help Center)
- File Arsip Aplikasi Intelijen Bisnis Oracle (BAR) (Pusat Bantuan Oracle)
- Cara memigrasikan OBI 12c antar lingkungan (Oracle Support)

Informasi tambahan

Berikut ini adalah daftar praktik terbaik yang terkait dengan migrasi Oracle BI 12c ke AWS Cloud.

Database repositori

Ini adalah praktik terbaik untuk meng-host skema database Oracle BI 12c pada contoh Amazon RDS for Oracle. Jenis instans ini menyediakan kapasitas hemat biaya dan dapat diubah ukurannya saat mengotomatiskan tugas administrasi, seperti penyediaan perangkat keras, penyiapan basis data, penambalan, dan pencadangan.

Untuk informasi selengkapnya, lihat Menggunakan Utilitas Pembuatan Repositori Oracle di RDS untuk Oracle di Panduan Pengguna Amazon RDS.

Tingkatan web dan aplikasi

EC2 Instans Amazon yang dioptimalkan untuk memori seringkali cocok untuk server Oracle BI 12c. Apa pun jenis instans yang Anda pilih, pastikan bahwa instance yang Anda berikan memenuhi persyaratan penggunaan memori sistem Anda. Selain itu, pastikan Anda mengonfigurasi ukuran heap WebLogic Java Virtual Machine (JVM) yang memadai berdasarkan memori EC2 instans Amazon yang tersedia.

Penyimpanan lokal

I/O memainkan peran penting dalam kinerja keseluruhan aplikasi Oracle BI 12c Anda. Amazon Elastic Block Store (Amazon EBS) menawarkan kelas penyimpanan berbeda yang dioptimalkan untuk pola beban kerja yang berbeda. Pastikan Anda memilih jenis volume Amazon EBS yang sesuai dengan kasus penggunaan Anda.

Untuk informasi selengkapnya tentang jenis volume EBS, lihat fitur Amazon EBS di dokumentasi Amazon EBS.

Penyimpanan bersama

Domain Oracle BI 12c yang dikelompokkan memerlukan penyimpanan bersama untuk sumber daya berikut:

- File konfigurasi
- Direktori data tunggal Oracle BI 12c (SDD)
- Cache global Oracle
- Skrip Penjadwal Oracle BI
- Binari Oracle WebLogic Server

Anda dapat memenuhi persyaratan penyimpanan bersama ini dengan menggunakan Amazon EFS, yang menyediakan sistem file Network File System (NFS) elastis yang dapat diskalakan dan dikelola sepenuhnya.

Fine tuning kinerja penyimpanan bersama

Amazon EFS memiliki dua mode throughput: Provisioned dan Bursting. Layanan ini juga memiliki dua mode kinerja: Tujuan Umum dan Max I/O.

Untuk menyempurnakan kinerja, mulailah dengan menguji beban kerja Anda dalam mode kinerja Tujuan Umum dan mode throughput yang disediakan. Melakukan tes ini akan membantu Anda menentukan apakah mode dasar tersebut cukup untuk memenuhi tingkat layanan yang Anda inginkan.

Untuk informasi selengkapnya, lihat kinerja Amazon EFS di Panduan Pengguna Amazon EFS.

Ketersediaan dan pemulihan bencana

Merupakan praktik terbaik untuk menerapkan komponen Oracle BI 12c di beberapa Availability Zone untuk melindungi sumber daya tersebut jika terjadi kegagalan Availability Zone. Berikut ini adalah daftar praktik terbaik ketersediaan dan pemulihan bencana untuk sumber daya Oracle BI 12c tertentu yang dihosting di AWS Cloud:

- Database repositori Oracle BI 12c: Menerapkan instance database Amazon RDS Multi-AZ ke database Oracle BI 12crepository Anda. Dalam penerapan Multi-AZ, Amazon RDS secara otomatis menyediakan dan memelihara replika siaga sinkron di AZ yang berbeda. Menjalankan instance database repositori Oracle BI 12c di seluruh Availability Zones dapat meningkatkan ketersediaan selama pemeliharaan sistem yang direncanakan dan membantu melindungi database Anda dari kegagalan instans dan Availability Zone.
- Server Terkelola Oracle BI 12c: Untuk mencapai toleransi kesalahan, ini adalah praktik terbaik untuk menerapkan komponen sistem Oracle BI 12c pada Server Terkelola dalam Grup Auto Scaling EC2 Amazon yang dikonfigurasi untuk menjangkau beberapa Availability Zone. Auto Scaling menggantikan instans yang salah berdasarkan pemeriksaan kesehatan Amazon. EC2 Jika terjadi kegagalan Availability Zone, Oracle HTTP Server terus mengarahkan lalu lintas ke Managed Server di Availability Zone yang berfungsi. Kemudian, Auto Scaling meluncurkan instance untuk mengikuti persyaratan jumlah host Anda. Mengaktifkan replikasi status sesi HTTP disarankan untuk membantu memastikan bahwa ada failover yang lancar dari sesi yang ada ke Server Terkelola yang berfungsi.

- Server Administrasi Oracle BI 12c: Untuk memastikan bahwa Server Administrasi Anda memiliki ketersediaan tinggi, host dalam grup Auto EC2 Scaling Amazon yang dikonfigurasi untuk menjangkau beberapa Availability Zone. Kemudian, atur ukuran minimum dan maksimum grup yang disetel ke 1. Jika terjadi kegagalan Availability Zone, EC2 Auto Scaling Amazon memulai Server Administrasi pengganti di Availability Zone alternatif. Untuk memulihkan host dasar yang gagal dalam Availability Zone yang sama, Anda dapat mengaktifkan Amazon EC2 Auto Recovery.
- Server Oracle Web Tier: Ini adalah praktik terbaik untuk mengaitkan Server HTTP Oracle Anda dengan domain Oracle WebLogic Server Anda. Untuk ketersediaan tinggi, terapkan Server HTTP Oracle Anda di grup Auto EC2 Scaling Amazon yang dikonfigurasi untuk aspen beberapa Availability Zoness. Kemudian, tempatkan server di belakang penyeimbang beban elastis ELB. Untuk memberikan perlindungan tambahan terhadap kegagalan host, Anda dapat mengaktifkan Amazon EC2 Auto Recovery.

Skalabilitas

Elastisitas AWS Cloud membantu Anda menskalakan aplikasi baik secara horizontal maupun vertikal sebagai respons terhadap persyaratan beban kerja.

Penskalaan vertikal

Untuk menskalakan aplikasi secara vertikal, Anda dapat mengubah ukuran dan jenis EC2 instans Amazon yang menjalankan komponen Oracle BI 12c Anda. Anda tidak perlu menyediakan instance secara berlebihan pada awal penerapan dan menimbulkan biaya yang tidak perlu.

Penskalaan horisontal

EC2 Auto Scaling Amazon membantu Anda menskalakan aplikasi secara horizontal dengan menambahkan atau menghapus Server Terkelola secara otomatis berdasarkan persyaratan beban kerja.

> **Note**
>
> Penskalaan horizontal dengan Amazon EC2 Auto Scaling memerlukan keterampilan scripting dan pengujian menyeluruh untuk diterapkan.

Backup dan pemulihan

Berikut ini adalah daftar praktik terbaik pencadangan dan pemulihan untuk sumber daya Oracle BI 12c tertentu yang dihosting di AWS Cloud:

- Repositori metadata Oracle Business Intelligence: Amazon RDS secara otomatis membuat dan menyimpan cadangan instance database Anda. Cadangan ini disimpan untuk jangka waktu yang Anda tentukan. Pastikan Anda mengonfigurasi durasi pencadangan Amazon RDS dan pengaturan retensi berdasarkan persyaratan perlindungan data Anda. Untuk informasi selengkapnya, lihat pencadangan dan pemulihan Amazon RDS.
- Server Terkelola, Server Administrasi, dan server Tingkat Web: Pastikan Anda mengonfigurasi snapshot Amazon EBS berdasarkan persyaratan perlindungan dan retensi data Anda.
- Penyimpanan bersama: Anda dapat mengelola pencadangan dan pemulihan untuk file yang disimpan di Amazon EFS dengan menggunakan AWS Backup. Layanan AWS Backup juga dapat digunakan untuk mengelola pencadangan dan pemulihan layanan lain secara terpusat, termasuk Amazon, EC2 Amazon EBS, dan Amazon RDS. Untuk informasi selengkapnya, lihat Apa itu AWS Backup? Dalam Panduan Pengembang AWS Backup.

Keamanan dan kepatuhan

Berikut ini adalah daftar praktik terbaik keamanan dan layanan AWS yang dapat membantu Anda melindungi aplikasi Oracle BI 12c Anda di AWS Cloud:

- Enkripsi saat istirahat: Amazon RDS, Amazon EFS, dan Amazon EBS semuanya mendukung algoritme enkripsi standar industri. Anda dapat menggunakan AWS Key Management Service (AWS KMS) untuk membuat dan mengelola kunci kriptografi serta mengontrol penggunaannya di seluruh layanan AWS dan dalam aplikasi Anda. Anda juga dapat mengonfigurasi Oracle Transparent Data Encryption (TDE) pada instance database Amazon RDS for Oracle yang menghosting database repositori Oracle BI 12c Anda.
- Enkripsi dalam perjalanan: Ini adalah praktik terbaik untuk mengaktifkan protokol SSL atau TLS untuk melindungi data dalam perjalanan antara berbagai lapisan instalasi Oracle BI 12c Anda. Anda dapat menggunakan AWS Certificate Manager (ACM) untuk menyediakan, mengelola, dan menyebarkan sertifikat SSL dan TLS publik dan pribadi untuk sumber daya Oracle BI 12c Anda.
- Keamanan jaringan: Pastikan Anda menerapkan sumber daya Oracle BI 12c di VPC Amazon yang memiliki kontrol akses yang sesuai yang dikonfigurasi untuk kasus penggunaan Anda. Konfigurasikan grup keamanan Anda untuk memfilter lalu lintas masuk dan keluar dari EC2 instans Amazon yang menjalankan instalasi Anda. Selain itu, pastikan Anda mengonfigurasi Network

Access Control Lists (NACLs) yang mengizinkan atau menolak lalu lintas berdasarkan aturan yang ditetapkan.

- Pemantauan dan pencatatan: Anda dapat menggunakan AWS CloudTrail untuk melacak panggilan API ke infrastruktur AWS Anda, termasuk sumber daya Oracle BI 12c Anda. Fungsionalitas ini berguna saat melacak perubahan infrastruktur atau saat melakukan analisis keamanan. Anda juga dapat menggunakan Amazon CloudWatch untuk melihat data operasional yang dapat memberi Anda wawasan yang dapat ditindaklanjuti tentang kinerja dan kesehatan aplikasi Oracle BI 12c Anda. Anda juga dapat mengonfigurasi alarm dan mengambil tindakan otomatis berdasarkan alarm tersebut. Amazon RDS menyediakan alat pemantauan tambahan, termasuk Enhanced Monitoring dan Performance Insights.

Migrasi Tumpukan ELK ke Elastic Cloud di AWS

Battulga Purevragchaa dan Antony Prasad Thevaraj, Amazon Web Services

uday reddy, Tidak ada

Ringkasan

Elastic telah menyediakan layanan selama bertahun-tahun, dengan pengguna dan pelanggan mereka biasanya mengelola Elastic sendiri di tempat. Elastic Cloud, layanan Elasticsearch yang dikelola, menyediakan cara untuk menggunakan Elastic Stack (ELK Stack) dan solusi untuk pencarian perusahaan, observabilitas, dan keamanan. Anda dapat mengakses solusi Elastic dengan aplikasi seperti Log, Metrik, APM (pemantauan kinerja aplikasi), dan SIEM (informasi keamanan dan manajemen acara). Anda dapat menggunakan fitur terintegrasi seperti pembelajaran mesin, manajemen siklus hidup indeks, Kibana Lens (untuk visualisasi seret dan lepas).

Saat Anda beralih dari Elasticsearch yang dikelola sendiri ke Elastic Cloud, layanan Elasticsearch menangani hal-hal berikut:

- Menyediakan dan mengelola infrastruktur yang mendasarinya
- Membuat dan mengelola cluster Elasticsearch
- Menskalakan cluster ke atas dan ke bawah
- Upgrade, menambal, dan mengambil snapshot

Ini memberi Anda lebih banyak waktu untuk fokus pada pemecahan tantangan lain.

Pola ini menentukan cara memigrasikan Elasticsearch 7.13 lokal ke Elasticsearch di Elasticsearch di Elastic Cloud di Amazon Web Services (AWS). Versi lain mungkin memerlukan sedikit modifikasi pada proses yang dijelaskan dalam pola ini. Untuk informasi lebih lanjut, hubungi perwakilan Elastis Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan akses ke Amazon Simple Storage Service (Amazon S3) untuk snapshot
- Tautan pribadi dengan bandwidth yang aman dan cukup tinggi untuk menyalin file data snapshot ke Amazon S3

- Amazon S3 Transfer Acceleration
- Kebijakan Snapshot Elastis untuk memastikan bahwa konsumsi data diarsipkan secara teratur, baik ke penyimpanan data lokal yang cukup besar atau ke penyimpanan jarak jauh (Amazon S3)

Anda harus memahami seberapa besar snapshot dan kebijakan lifecyle untuk indeks yang menyertainya berada di lokasi sebelum memulai migrasi. Untuk informasi lebih lanjut, hubungi Elastic.

Peran dan keterampilan

Proses migrasi juga membutuhkan peran dan keahlian yang dijelaskan dalam tabel berikut.

Peran	Keahlian	Tanggung Jawab
Dukungan aplikasi	Keakraban dengan Awan Elastis dan Elastis di tempat	Semua tugas terkait Elastis
Administrator sistem atau DBA	Pengetahuan mendalam tentang lingkungan Elastis lokal dan konfigurasinya	Kemampuan untuk menyediakan penyimpanan, menginstal, dan menggunakan AWS Command Line Interface (AWS CLI) Interface (AWS CLI), dan mengidentifikasi semua sumber data yang memberi makan Elastic di lokasi
Administrator jaringan	Pengetahuan tentang konektivitas, keamanan, dan kinerja jaringan AWS lokal ke AWS	Pembentukan tautan jaringan dari tempat ke Amazon S3, dengan pemahaman tentang bandwidth konektivitas

Batasan

- Elasticsearch di Elastic Cloud hanya tersedia di Wilayah AWS yang didukung (September 2021).

Versi produk

- Elasticsearch 7.13

Arsitektur

Tumpukan teknologi sumber

Elasticsearch 7.13 di tempat atau versi lebih baru:

- Snapshot klaster
- Cuplikan indeks
- Konfigurasi Beats

Arsitektur teknologi sumber

Diagram berikut menunjukkan arsitektur lokal yang khas dengan metode konsumsi yang berbeda, tipe node, dan Kibana. Tipe node yang berbeda mencerminkan peran cluster, otentikasi, dan visualisasi Elasticsearch.

1. Tertelan dari Beats ke Logstash
2. Tertelan dari Beats ke antrian pesan Apache Kafka
3. Tertelan dari Filebeat ke Logstash
4. Tertelan dari antrian pesan Apache Kafka ke Logstash
5. Tertelan dari Logstash ke cluster Elasticsearch
6. Kluster Elasticsearch
7. Node otentikasi dan notifikasi
8. Kibana dan node gumpalan

Tumpukan teknologi target

Elastic Cloud diterapkan ke perangkat lunak Anda sebagai akun layanan (SaaS) di beberapa Wilayah AWS dengan replikasi lintas cluster.

- Snapshot klaster
- Cuplikan indeks

- Konfigurasi Beats
- Awan Elastis
- Penyeimbang Beban Jaringan
- Amazon Route 53
- Amazon S3

Arsitektur target

Infrastruktur Elastic Cloud yang dikelola adalah:

- Sangat tersedia, hadir di beberapa Availability Zone dan beberapa Wilayah AWS.
- Toleran kegagalan wilayah karena data (indeks dan snapshot) direplikasi menggunakan replikasi lintas cluster (CCR) Elastic Cloud
- Arsip, karena snapshot diarsipkan di Amazon S3
- Partisi jaringan toleran melalui kombinasi Network Load Balancers dan Route 53
- Konsumsi data yang berasal dari (tetapi tidak terbatas pada) APM Elastis, Beats, Logstash

Langkah migrasi tingkat tinggi

Elastic telah mengembangkan metodologi preskriptifnya sendiri untuk memigrasikan Elastic Cluster lokal ke Elastic Cloud. Metodologi Elastic secara langsung selaras dan melengkapi panduan migrasi AWS dan praktik terbaik, termasuk Well-Architected Framework dan AWS Migration Acceleration Program (MAP). Biasanya, tiga fase migrasi AWS adalah sebagai berikut:

- Menilai
- Memobilisasi
- Migrasi dan modernisasi

Elastis mengikuti fase migrasi serupa dengan terminologi komplementer:

- Memulai
- Rencana
- Implementasikan

- Mengirimkan
- Tutup

Elastic menggunakan Metodologi Implementasi Elastis untuk memfasilitasi penyampaian hasil proyek. Ini inklusif dengan desain untuk memastikan bahwa Elastic, tim konsultasi, dan tim pelanggan bekerja sama dengan jelas untuk bersama-sama memberikan hasil yang diinginkan.

Metodologi Elastis menggabungkan pentahapan air terjun tradisional dengan Scrum dalam fase implementasi. Konfigurasi persyaratan teknis disampaikan secara berulang secara kolaboratif sambil meminimalkan risiko.

Alat

Layanan AWS

- Amazon Route 53 - Amazon Route 53 adalah layanan web Sistem Nama Domain (DNS) yang sangat tersedia dan dapat diskalakan. Anda dapat menggunakan Route 53 untuk melakukan tiga fungsi utama dalam kombinasi apapun: pendaftaran domain, perutean DNS, dan pemeriksaan kondisi.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah layanan penyimpanan objek. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web. Pola ini menggunakan bucket S3 dan Amazon S3 Transfer Acceleration.
- Elastic Load Balancing — Elastic Load Balancing secara otomatis mendistribusikan lalu lintas masuk Anda ke beberapa target, EC2 seperti instance, kontainer, dan alamat IP, dalam satu atau beberapa Availability Zone.

Alat-alat lainnya

- Beats - Beats mengirimkan data dari Logstash atau Elasticsearch
- Elastic Cloud - Elastic Cloud adalah layanan terkelola untuk hosting Elasticsearch.
- Elasticsearch — Elasticsearch adalah mesin pencari dan analitik yang menggunakan Elastic Stack untuk menyimpan data Anda secara terpusat untuk penelusuran dan analitik yang berskala. Pola ini juga menggunakan pembuatan snapshot dan replikasi lintas cluster.

- Logstash - Logstash adalah pipeline pemrosesan data sisi server yang menyerap data dari berbagai sumber, mengubahnya, dan kemudian mengirimkannya ke penyimpanan data Anda.

Epik

Siapkan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi server yang menjalankan solusi Elastic lokal.	Konfirmasikan bahwa migrasi elastis didukung.	Pemilik aplikasi
Memahami konfigurasi server lokal.	Untuk memahami konfigura si server yang diperlukan untuk mengarahkan beban kerja dengan sukses di lokasi, temukan jejak perangkat keras server, konfigurasi jaringan, dan karakteristik penyimpanan yang saat ini digunakan	Support Aplikasi
Kumpulkan informasi akun pengguna dan aplikasi.	Identifikasi nama pengguna dan nama aplikasi yang digunakan oleh lingkungan Elastic lokal.	Administrator sistem, Dukungan aplikasi
Document Beats dan konfigurasi pengirim data.	Untuk mendokumentasikan konfigurasi, lihat sumber dan sink data yang ada. Untuk informasi selengkapnya, lihat dokumentasi Elastis.	Dukungan aplikasi
Tentukan kecepatan dan volume data.	Tetapkan garis dasar untuk berapa banyak data yang ditangani cluster.	Administrator sistem, Dukungan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dokumen skenario RPO dan RTO.	Skenario tujuan titik pemulihan dokumen (RPO) dan tujuan waktu pemulihan (RTO) dalam hal pemadaman dan perjanjia n tingkat layanan (). SLAs	Pemilik aplikasi, Administrator sistem, Dukungan aplikasi
Tentukan pengaturan siklus hidup snapshot yang optimal.	Tentukan seberapa sering data perlu diamankan dengan menggunakan snapshot Elastis selama dan setelah migrasi.	Pemilik aplikasi, Administrator sistem, Dukungan aplikasi
Tentukan ekspektasi kinerja pasca-migrasi.	Hasilkan metrik pada penyegaran layar saat ini dan yang diharapkan, runtime kueri, dan perilaku antarmuka pengguna.	Administrator sistem, Dukungan aplikasi
Dokumentasikan transportasi akses internet, bandwidth, dan persyaratan ketersediaan.	Pastikan kecepatan, latensi, dan ketahanan koneksi internet untuk menyalin snapshot ke Amazon S3.	Administrator jaringan
Dokumentasikan biaya runtime lokal saat ini untuk Elastic.	Pastikan bahwa ukuran lingkungan yang ditargetkan AWS dirancang agar berkinerj a tinggi dan hemat biaya.	DBA, Administrator sistem, Dukungan aplikasi
Identifikasi kebutuhan otentikasi dan otorisasi.	Fitur keamanan Elastic Stack menyediakan ranah bawaan seperti Lightweight Directory Access Protocol (LDAP), Security Assertion Markup Language (SAMP), dan OpenID Connect (OIDC).	DBA, Administrator sistem, Dukungan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memahami persyaratan peraturan khusus berdasarkan lokasi geografis.	Pastikan bahwa data diekspor dan dienkripsi sesuai dengan kebutuhan Anda dan dengan persyaratan nasional yang relevan.	DBA, Administrator sistem, Dukungan aplikasi

Menerapkan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan area pementasan di Amazon S3.	Untuk menerima snapshot di Amazon S3, buat bucket S3 dan peran AWS Identity and Access Managemen t (IAM) sementara dengan akses penuh ke bucket yang baru dibuat. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM. Gunakan AWS Security Token Service untuk meminta kredensil keamanan sementara. Simpan ID kunci akses, kunci akses rahasia, dan token sesi diamankan. Aktifkan Amazon S3 Transfer Acceleration di bucket.	Administrator AWS
Instal AWS CLI dan plugin Amazon S3 di tempat.	Pada setiap node Elasticse arch, jalankan perintah berikut.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`sudo bin/elasticsearch-plugin install repository-s3` Kemudian reboot node.	

Konfigurasikan akses klien Amazon S3.	Tambahkan kunci yang dibuat sebelumnya dengan menjalankan perintah berikut.
	
	`elasticsearch-keystore add s3.client.default.access_key`
	
	`elasticsearch-keystore add s3.client.default.secret_key`
	
	`elasticsearch-keystore add s3.client.default.session_token`
		Administrator AWS

Daftarkan repositori snapshot untuk data Elastis	Gunakan Kibana Dev Tools untuk memberi tahu kluster lokal lokal tempat bucket S3 jarak jauh mana yang akan ditulis.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kebijakan snapshot.	Untuk mengonfigurasi manajemen siklus hidup snapshot, pada tab Kebijakan Kibana, pilih kebijakan SLM, dan tentukan waktu, aliran data, atau indeks mana yang harus disertakan, dan nama apa yang akan digunakan. Konfigurasikan kebijakan yang sering mengambil snapshot. Snapshot bersifat inkrement al dan memanfaatkan penyimpanan secara efisien. Cocokkan keputusan penilaian kesiapan Anda. Kebijakan juga dapat menentukan kebijakan penyimpanan dan menghapus snapshot secara otomatis saat tidak diperlukan lagi.	Dukungan aplikasi
Verifikasi bahwa snapshot berfungsi.	Di Kibana Dev Tools, jalankan perintah berikut. `GET _snapshot/<your_re po_name>/_all`	Administrator AWS, Dukungan aplikasi,
Terapkan cluster baru di Elastic Cloud.	Masuk ke Elastic, dan pilih klaster untuk “observabilitas, pencarian, atau keamanan” yang berasal dari temuan bisnis Anda dalam penilaian kesiapan.	Administrator AWS, Dukungan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan akses penyimpanan kunci cluster.	Cluster baru membutuhkan akses ke bucket S3 yang akan menyimpan snapshot. Pada Konsol Layanan Elasticse arch, pilih Keamanan, dan masukkan akses dan kunci rahasia IAM yang Anda buat sebelumnya.	Administrator AWS
Konfigurasikan cluster yang dihosting Elastic Cloud untuk mengakses Amazon S3.	Siapkan akses cluster baru ke repositori snapshot yang dibuat sebelumnya di Amazon S3. Menggunakan Kibana, lakukan hal berikut: 1. Pilih Manajemen Stack, Pengaturan Snapshot, RegisterRepo. 2. Di bidang Alias, masukkan nama repositori. 3. Untuk nama Klien S3, pilih sekunder. 4. Tambahkan bucket S3 yang Anda buat sebelumnya ke repositori. 5. Pilih Kompres snapshot. 6. Untuk pengaturan Enkripsi, pertahankan nilai default.	Administrator AWS, Dukungan Aplikasi
Verifikasi repositori Amazon S3 baru.	Pastikan Anda dapat mengakses repositori baru yang dihosting di cluster Elastic Cloud.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Initilaize cluster layanan Elasticsearch.	Di Konsol Layanan Elasticsearch, inisialisasi kluster layanan Elasticsearch dari snapshot S3.
	
	Jalankan perintah berikut sebagai POST.
	
	`*/_close?expand_wildcards=all`
	
	`/_snapshot/<your-repo-name>/ <your-snapshot-name>/_restore`
	
	`*/_open?expand_wildcards=all`
		Support Aplikasi

Selesaikan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa pemulihan snapshot berhasil.	Menggunakan Kibana Dev Tools, jalankan perintah berikut. `GET _cat/indices`	Dukungan aplikasi
Layanan konsumsi Redploy.	Connect endpoint untuk Beats dan Logstash ke endpoint layanan Elasticsearch yang baru.	Dukungan aplikasi

Uji lingkungan cluster dan bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi lingkungan cluster.	Setelah lingkungan cluster Elastic lokal dimigrasikan ke AWS, Anda dapat menyambungkannya dan menggunakan alat pengujian penerimaan pengguna (UAT) Anda sendiri untuk memvalidasi lingkungan baru.	Dukungan aplikasi
Bersihkan sumber daya.	Setelah Anda memvalidasi bahwa cluster berhasil bermigrasi, hapus bucket S3 dan peran IAM yang digunakan untuk migrasi.	Administrator AWS

Sumber daya terkait

Referensi elastis

- Awan Elastis
- Elasticsearch dan Kibana Terkelola di AWS
- Pencarian perusahaan elastis
- Integrasi elastis
- Observabilitas elastis
- Keamanan elastis
- Ketukan
- APM elastis
- Migrasi ke manajemen siklus hidup indeks
- Langganan elastis
- Kontak Elastis

Posting blog elastis

- Cara bermigrasi dari Elasticsearch yang dikelola sendiri ke Elastic Cloud di AWS (posting blog)
- Bermigrasi ke Elastic Cloud (posting blog)

Dokumentasi elastis

- Tutorial: Otomatiskan cadangan dengan SLM
- ILM: Kelola siklus hidup indeks
- Logstash
- Replikasi lintas cluster (CCR)
- Saluran pipa menelan
- Jalankan permintaan API Elasticsearch
- Retensi snapshot

Video elastis dan webinar

- Migrasi awan elastis
- Elastic Cloud: Mengapa pelanggan bermigrasi (webinar)

Referensi AWS

- Awan Elastis di AWS Marketplace
- Antarmuka Baris Perintah AWS
- AWS Direct Connect
- AWS Migration Acceleration Program
- Penyeimbang Beban Jaringan
- Wilayah dan Zona Ketersediaan
- Amazon Route 53
- Layanan Penyimpanan Sederhana Amazon
- Amazon S3 Transfer Acceleration
- Koneksi VPN
- Kerangka Well-Architected

Informasi tambahan

Jika Anda berencana untuk memigrasikan beban kerja yang kompleks, gunakan Layanan Konsultasi Elastis. Jika Anda memiliki pertanyaan dasar terkait konfigurasi dan layanan, hubungi tim Elastic Support.

Migrasi data ke AWS Cloud dengan menggunakan Starburst

Antony Prasad Thevaraj dan Suresh Veeragoni, Amazon Web Services

Shaun Van Staden, Tidak Ada

Ringkasan

Starburst membantu mempercepat perjalanan migrasi data Anda ke Amazon Web Services (AWS) dengan menyediakan mesin kueri perusahaan yang menyatukan sumber data yang ada dalam satu titik akses. Anda dapat menjalankan analitik di berbagai sumber data untuk mendapatkan wawasan berharga, sebelum menyelesaikan rencana migrasi apa pun. Tanpa mengganggu business-as-usual analitik, Anda dapat memigrasikan data dengan menggunakan mesin Starburst atau aplikasi ekstrak, transformasi, dan muat (ETL) khusus.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Awan pribadi virtual (VPC)
- Cluster Amazon Elastic Kubernetes Service (Amazon EKS)
- Grup Auto Scaling Amazon Elastic Compute Cloud (Amazon EC2)
- Daftar beban kerja sistem saat ini yang perlu dimigrasikan
- Konektivitas jaringan dari AWS ke lingkungan lokal

Arsitektur

Arsitektur referensi

Diagram arsitektur tingkat tinggi berikut menunjukkan penerapan khas Starburst Enterprise di: AWS Cloud

1. Kluster Starburst Enterprise berjalan di dalam kluster Anda Akun AWS.
2. Pengguna mengautentikasi dengan menggunakan Lightweight Directory Access Protocol (LDAP) atau Open Authorization (OAuth) dan berinteraksi langsung dengan cluster Starburst.
3. Starburst dapat terhubung ke beberapa sumber AWS data, seperti AWS Glue, Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), dan Amazon

Redshift. Starburst menyediakan kemampuan kueri gabungan di seluruh sumber data di AWS Cloud, di tempat, atau di lingkungan cloud lainnya.

4. Anda meluncurkan Starburst Enterprise di klaster Amazon EKS dengan menggunakan bagan Helm.
5. Starburst Enterprise menggunakan grup Amazon EC2 Auto Scaling dan Instans Spot EC2 Amazon untuk mengoptimalkan infrastruktur.
6. Starburst Enterprise terhubung langsung ke sumber data lokal yang ada untuk membaca data secara real-time. Selain itu, jika Anda memiliki penyebaran Starburst Enterprise yang ada di lingkungan ini, Anda dapat langsung menghubungkan cluster Starburst baru Anda AWS Cloud ke cluster yang ada ini.

Harap perhatikan hal berikut:

- Starburst bukan platform virtualisasi data. Ini adalah mesin query massively parallel processing (MPP) berbasis SQL yang membentuk dasar strategi data mesh keseluruhan untuk analitik.
- Ketika Starburst digunakan sebagai bagian dari migrasi, Starburst memiliki konektivitas langsung ke infrastruktur lokal yang ada.
- Starburst menyediakan beberapa built-in enterprise dan open-source konektor yang memfasilitasi konektivitas ke berbagai sistem warisan. Untuk daftar lengkap konektor dan kemampuannya, lihat Konektor di panduan pengguna Starburst Enterprise.
- Starburst dapat melakukan kueri data secara real-time dari sumber data lokal. Ini mencegah gangguan operasi bisnis reguler saat data sedang dimigrasikan.
- Jika Anda bermigrasi dari penerapan Starburst Enterprise lokal yang ada, Anda dapat menggunakan konektor khusus, Starburst Stargate, untuk menghubungkan kluster Starburst Enterprise langsung ke kluster lokal. AWS Ini memberikan manfaat kinerja tambahan saat pengguna bisnis dan analis data menggabungkan kueri dari lingkungan lokal AWS Cloud ke lingkungan lokal Anda.

Ikhtisar proses tingkat tinggi

Anda dapat mempercepat proyek migrasi data dengan menggunakan Starburst karena Starburst mengaktifkan wawasan di semua data Anda, sebelum memigrasikannya. Gambar berikut menunjukkan proses khas untuk memigrasi data dengan menggunakan Starburst.

Peran

Peran berikut biasanya diperlukan untuk menyelesaikan migrasi menggunakan Starburst:

- Administrator cloud — Bertanggung jawab untuk membuat sumber daya cloud tersedia untuk menjalankan aplikasi Starburst Enterprise
- Administrator Starburst — Bertanggung jawab untuk menginstal, mengonfigurasi, mengelola, dan mendukung aplikasi Starburst
- Insinyur data — Bertanggung jawab untuk:
  - Migrasi data lama ke cloud
  - Membangun pandangan semantik untuk mendukung analitik
- Pemilik solusi atau sistem — Bertanggung jawab atas implementasi solusi secara keseluruhan

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.

Alat-alat lainnya

- Helm - Helm adalah manajer paket untuk Kubernetes yang membantu Anda menginstal dan mengelola aplikasi di klaster Kubernetes Anda.
- Starburst Enterprise adalah mesin query massively parallel processing (MPP) berbasis SQL yang membentuk dasar strategi data mesh secara keseluruhan untuk analitik.
- Starburst Stargate — Starburst Stargate menautkan katalog dan sumber data dalam satu lingkungan Starburst Enterprise, seperti cluster di pusat data lokal, ke katalog dan sumber data di lingkungan Starburst Enterprise lainnya, seperti cluster di. AWS Cloud

Epik

Menilai datanya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi dan prioritaskan data Anda.	Identifikasi data yang ingin Anda pindahkan. Sistem lama lokal yang besar dapat menyertakan data inti yang ingin Anda migrasi bersama data yang tidak ingin Anda pindahkan atau tidak dapat dipindahkan karena alasan kepatuhan. Dimulai dengan inventaris data membantu Anda memprioritaskan data mana yang harus Anda targetkan terlebih dahulu. Untuk informasi selengkap nya, lihat Memulai penemuan portofolio otomatis.	Insinyur data, DBA
Jelajahi, inventarisasi, dan buat cadangan data Anda.	Validasi kualitas, kuantitas , dan relevansi data untuk kasus penggunaan Anda. Cadangkan atau buat snapshot data sesuai kebutuhan, dan selesaikan lingkungan target untuk data.	Insinyur data, DBA

Siapkan lingkungan Starburst Enterprise

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Starburst Enterprise di file. AWS Cloud	Saat data sedang dikatalog kan, siapkan Starburst	Administrator AWS, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Enterprise di cluster Amazon EKS yang dikelola. Untuk informasi selengkapnya lihat, Menerapkan dengan Kubernetes di dokumentasi referensi Starburst Enterprise. Ini memungkinkan business-as-usual analitik saat migrasi data sedang dalam proses.	
Connect Starburst ke sumber data.	Setelah Anda mengidentifikasi data dan mengatur Starburst Enterprise, sambungkan Starburst ke sumber data. Starburst membaca data langsung dari sumber data sebagai query SQL. Untuk informasi selengkapnya, lihat dokumentasi referensi Starburst Enterprise.	Administrator AWS, Pengembang aplikasi

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membangun dan menjalankan pipeline ETL.	Mulailah proses migrasi data. Aktivitas ini dapat terjadi bersamaan dengan business-as-usual analitik. Untuk migrasi, Anda dapat menggunakan produk pihak ketiga atau Starburst. Starburst memiliki kemampuan untuk membaca dan menulis data	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di berbagai sumber. Untuk informasi selengkapnya, lihat dokumentasi referensi Starburst Enterprise.	
Validasi data.	Setelah data dimigrasi, validasi data untuk memastikan semua data yang diperlukan telah dipindahkan dan utuh.	Insinyur data, DevOps insinyur

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Potong data.	Setelah migrasi dan validasi data selesai, Anda dapat memotong data. Ini melibatkan mengubah tautan koneksi data di Starburst. Alih-alih menunjuk ke sumber lokal, Anda mengarahkan ke sumber cloud baru dan memperbarui tampilan semantik. Untuk informasi selengkapnya, lihat Konektor dalam dokumentasi referensi Starburst Enterprise.	Insinyur data, Cutover lead
Luncurkan ke pengguna.	Konsumen data mulai bekerja dari sumber data yang dimigrasi. Proses ini tidak terlihat oleh pengguna akhir analitik.	Cutover lead, Insinyur data

Sumber daya terkait

AWS Marketplace

- Galaksi Starburst
- Perusahaan Starburst
- Data Starburst JumpStart
- Starburst Enterprise dengan Graviton

Dokumentasi Starburst

- Panduan pengguna Starburst Enterprise
- Dokumentasi referensi Starburst Enterprise

AWS Dokumentasi lainnya

- Memulai dengan penemuan portofolio otomatis (Panduan AWS Preskriptif)
- Mengoptimalkan Biaya dan Kinerja Infrastruktur Cloud dengan Starburst di AWS (AWS posting blog)

Optimalkan konsumsi ETL ukuran file input di AWS

Apoorva Patrikar, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara mengoptimalkan langkah konsumsi proses ekstrak, transformasi, dan pemuatan (ETL) untuk big data dan beban kerja Apache Spark di AWS Glue dengan mengoptimalkan ukuran file sebelum memproses data Anda. Gunakan pola ini untuk mencegah atau menyelesaikan masalah file kecil. Artinya, ketika sejumlah besar file kecil memperlambat pemrosesan data karena ukuran agregat file. Misalnya, ratusan file yang masing-masing hanya beberapa ratus kilobyte dapat secara signifikan memperlambat kecepatan pemrosesan data untuk pekerjaan AWS Glue Anda. Ini karena AWS Glue harus menjalankan fungsi daftar internal di Amazon Simple Storage Service (Amazon S3) dan YARN (Yet Another Resource Negotiator) harus menyimpan sejumlah besar metadata. Untuk meningkatkan kecepatan pemrosesan data, Anda dapat menggunakan pengelompokan untuk mengaktifkan tugas ETL Anda membaca sekelompok file input ke dalam satu partisi dalam memori. Partisi secara otomatis mengelompokkan file yang lebih kecil bersama-sama. Atau, Anda dapat menggunakan kode kustom untuk menambahkan logika batch ke file yang ada.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Satu atau lebih pekerjaan lem AWS
- Satu atau lebih big data atau beban kerja Apache Spark
- Ember S3

Arsitektur

Pola berikut menunjukkan bagaimana data dalam format yang berbeda diproses oleh pekerjaan AWS Glue dan kemudian disimpan dalam bucket S3 untuk mendapatkan visibilitas ke kinerja.

Diagram menunjukkan alur kerja berikut:

1.
> **Note**
>
> Pekerjaan AWS Glue mengonversi file kecil dalam format CSV, JSON, dan Parquet menjadi frame dinamis. : Ukuran file input memiliki dampak paling signifikan pada kinerja pekerjaan AWS Glue.

2. Pekerjaan AWS Glue menjalankan fungsi daftar internal dalam bucket S3.

Alat

- AWS Glue adalah layanan ETL yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Epik

Gunakan pengelompokan untuk mengoptimalkan konsumsi ETL selama membaca

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan ukuran grup.	Jika Anda memiliki lebih dari 50.000 file, pengelompokan dilakukan secara default. Namun, Anda juga dapat menggunakan pengelompokan untuk kurang dari 50.000 file dengan menentukan ukuran grup dalam parameter. `connectionOptions` `connectionOptions` Parameternya ada dalam `create_dynamic_frame.from_options` metode.	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tulis kode pengelompokan.	Gunakan `create_dynamic_frame` metode ini untuk membuat bingkai dinamis. Contoh: `S3bucket_node1 = glueContext.create_dynamic_frame.from_options(format_options={"multiline": False}, connection_type="s3", format="json", connection_options={ "paths": ["s3://bucket/prefix/file.json"], "recurse": True, "groupFiles": 'inPartition', "groupSize": 1048576 }, transformation_ctx="S3bucket_node1",)`  **Note** Gunakan `groupFiles` untuk mengelompokkan file dalam grup partisi Amazon S3. Gunakan `groupSize` untuk	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengatur ukuran target grup yang akan dibaca dalam memori. Tentukan `groupSize` dalam byte (1048576 = 1 MB).	
Tambahkan kode ke alur kerja.	Tambahkan kode pengelompokan ke alur kerja pekerjaan Anda di AWS Glue.	Insinyur data

Gunakan logika khusus untuk mengoptimalkan konsumsi ETL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih bahasa dan platform pemrosesan.	Pilih bahasa scripting dan platform pemrosesan yang disesuaikan dengan kasus penggunaan Anda.	Arsitek awan
Tulis kodenya.	Tulis logika kustom untuk mengumpulkan file Anda bersama-sama.	Arsitek awan
Tambahkan kode ke alur kerja.	Tambahkan kode ke alur kerja pekerjaan Anda di AWS Glue. Ini memungkinkan logika kustom Anda diterapkan setiap kali pekerjaan dijalankan.	Insinyur data

Partisi ulang saat menulis data setelah transformasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menganalisis pola konsumsi.	Cari tahu bagaimana aplikasi hilir akan menggunakan data yang Anda tulis. Misalnya, jika mereka menanyakan data setiap hari dan Anda hanya mempartisi data per Wilayah atau memiliki file output yang sangat kecil, seperti 2,5 KB per file, maka ini tidak optimal untuk konsumsi.	DBA
Partisi ulang data sebelum menulis.	Partisi ulang berdasarkan gabungan atau kueri selama pemrosesan (berdasar kan logika pemrosesan) dan setelah pemrosesan (berdasarkan konsumsi) . Misalnya, partisi ulang berdasarkan ukuran byte, seperti `.repartit ion(100000)` , atau partisi ulang berdasarkan kolom, seperti. `.repartit ion("column_name")`	Insinyur data

Sumber daya terkait

- Membaca file input dalam kelompok yang lebih besar
- Memantau AWS Glue
- Memantau AWS Glue menggunakan CloudWatch metrik Amazon
- Pemantauan dan debugging Job

- Memulai ETL tanpa server di AWS Glue

Informasi tambahan

Menentukan ukuran file

Tidak ada cara mudah untuk menentukan apakah ukuran file terlalu besar atau terlalu kecil. Dampak ukuran file pada kinerja pemrosesan tergantung pada konfigurasi cluster Anda. Di inti Hadoop, kami menyarankan Anda menggunakan file yang 128 MB atau 256 MB untuk memaksimalkan ukuran blok.

Untuk sebagian besar beban kerja file teks di AWS Glue, kami merekomendasikan ukuran file antara 100 MB dan 1 GB untuk cluster 5-10 DPU. Untuk mengetahui ukuran file input terbaik, pantau bagian preprocessing dari pekerjaan AWS Glue Anda, lalu periksa pemanfaatan CPU dan pemanfaatan memori pekerjaan tersebut.

Pertimbangan tambahan

Jika kinerja pada tahap ETL awal merupakan hambatan, pertimbangkan untuk mengelompokkan atau menggabungkan file data sebelum diproses. Jika Anda memiliki kontrol penuh pada proses pembuatan file, akan lebih efisien untuk menggabungkan titik data pada sistem sumber itu sendiri sebelum data mentah dikirim ke AWS.

Mengatur pipeline ETL dengan validasi, transformasi, dan partisi menggunakan AWS Step Functions

Sandip Gangapadhyay, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara membuat pipeline ekstrak, transformasi, dan beban (ETL) tanpa server untuk memvalidasi, mengubah, mengompres, dan mempartisi kumpulan data CSV besar untuk kinerja dan pengoptimalan biaya. Pipeline diatur oleh AWS Step Functions dan mencakup penanganan kesalahan, coba ulang otomatis, dan fitur notifikasi pengguna.

Saat file CSV diunggah ke folder sumber bucket Amazon Simple Storage Service (Amazon S3), pipeline ETL mulai berjalan. Pipeline memvalidasi konten dan skema file CSV sumber, mengubah file CSV menjadi format Apache Parquet terkompresi, mempartisi kumpulan data berdasarkan tahun, bulan, dan hari, dan menyimpannya dalam folder terpisah untuk diproses alat analitik.

Kode yang mengotomatiskan pola ini tersedia di GitHub, di ETL Pipeline dengan AWS Step Functions repositori.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi dengan Anda Akun AWS, sehingga Anda dapat membuat AWS sumber daya dengan menerapkan AWS CloudFormation tumpukan. Kami merekomendasikan menggunakan AWS CLI versi 2. Untuk petunjuk, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLI dokumentasi. AWS CLI Untuk petunjuk konfigurasi, lihat Pengaturan konfigurasi dan file kredensi dalam AWS CLI dokumentasi.
- Bucket Amazon S3.
- Dataset CSV dengan skema yang benar. (Repositori kode yang disertakan dengan pola ini menyediakan contoh file CSV dengan skema dan tipe data yang benar yang dapat Anda gunakan.)
- Browser web yang mendukung Konsol Manajemen AWS. (Lihat daftar browser yang didukung.)
- AWS Glue akses konsol.
- AWS Step Functions akses konsol.

Batasan

- Dalam AWS Step Functions, batas maksimum untuk menyimpan catatan riwayat adalah 90 hari. Untuk informasi selengkapnya, lihat kuota layanan Step Functions dalam AWS Step Functions dokumentasi.

Versi produk

- Python 3.13 untuk AWS Lambda
- AWS Glue versi 4.0

Arsitektur

Alur kerja yang diilustrasikan dalam diagram terdiri dari langkah-langkah tingkat tinggi ini:

1. Pengguna mengunggah file CSV ke folder sumber di Amazon S3.
2. Peristiwa notifikasi Amazon S3 memulai AWS Lambda fungsi yang memulai mesin status. AWS Step Functions
3. Fungsi Lambda memvalidasi skema dan tipe data dari file CSV mentah.
4. Tergantung pada hasil validasi:
   a. Jika validasi file sumber berhasil, file pindah ke folder tahap untuk diproses lebih lanjut.
   b. Jika validasi gagal, file pindah ke folder kesalahan, dan pemberitahuan kesalahan dikirim melalui Amazon Simple Notification Service (Amazon SNS).
5. AWS Glue Crawler membuat skema file mentah dari folder panggung di Amazon S3.
6. AWS Glue Pekerjaan mengubah, mengompres, dan mempartisi file mentah ke dalam format Parket.
7. AWS Glue Pekerjaan juga memindahkan file ke folder transformasi di Amazon S3.
8. AWS Glue Crawler membuat skema dari file yang diubah. Skema yang dihasilkan dapat digunakan oleh pekerjaan analitik apa pun. Anda juga dapat menggunakan Amazon Athena untuk menjalankan kueri ad-hoc.
9. Jika pipeline selesai tanpa kesalahan, file skema dipindahkan ke folder arsip. Jika ada kesalahan yang ditemui, file dipindahkan ke folder kesalahan sebagai gantinya.

10 Amazon SNS mengirimkan notifikasi yang menunjukkan keberhasilan atau kegagalan berdasarkan status penyelesaian pipeline.

Semua AWS sumber daya yang digunakan dalam pola ini tanpa server. Tidak ada server untuk dikelola.

Alat

Layanan AWS

- AWS Glue— AWS Glue adalah layanan ETL yang dikelola sepenuhnya yang memudahkan pelanggan untuk menyiapkan dan memuat data mereka untuk analitik.
- AWS Step Functions- AWS Step Functions adalah layanan orkestrasi tanpa server yang memungkinkan Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting. Melalui konsol AWS Step Functions grafis, Anda melihat alur kerja aplikasi Anda sebagai serangkaian langkah berbasis peristiwa.
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) adalah layanan pesan yang sangat tersedia, tahan lama, aman, dan pub/sub terkelola sepenuhnya yang memungkinkan Anda memisahkan layanan mikro, sistem terdistribusi, dan aplikasi tanpa server.
- AWS Lambda— AWS Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. AWS Lambda menjalankan kode Anda hanya bila diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.

Kode

Kode untuk pola ini tersedia di GitHub, di ETL Pipeline dengan AWS Step Functions repositori. Repositori kode berisi file dan folder berikut:

- `template.yml`— AWS CloudFormation template untuk membuat pipa ETL dengan AWS Step Functions.
- `parameter.json`- Berisi semua parameter dan nilai parameter. Anda memperbarui file ini untuk mengubah nilai parameter, seperti yang dijelaskan di bagian Epik.

- `myLayer/python`folder - Berisi paket Python yang diperlukan untuk membuat AWS Lambda lapisan yang diperlukan untuk proyek ini.
- `lambda`folder - Berisi fungsi Lambda berikut:
  - `move_file.py`— Memindahkan kumpulan data sumber ke folder arsip, transformasi, atau kesalahan.
  - `check_crawler.py`— Memeriksa status AWS Glue crawler sebanyak yang dikonfigurasi oleh variabel `RETRYLIMIT` lingkungan sebelum mengirim pesan kegagalan.
  - `start_crawler.py`— Memulai AWS Glue crawler.
  - `start_step_function.py`— Mulai AWS Step Functions.
  - `start_codebuild.py`Memulai AWS CodeBuild proyek.
  - `validation.py`— Memvalidasi dataset mentah input.
  - `s3object.py`— Membuat struktur direktori yang diperlukan di dalam bucket Amazon S3.
  - `notification.py`— Mengirim pemberitahuan sukses atau kesalahan di akhir pipeline.

Untuk menggunakan kode sampel, ikuti instruksi di bagian Epik.

Epik

Siapkan file sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode sampel.	1. Buka Pipeline ETL dengan AWS Step Functions repositori. 2. Pilih Kode pada halaman repositori utama, di atas daftar file, dan salin URL yang tercantum di bawah Clone dengan HTTPS. 3. Ubah direktori kerja Anda ke lokasi tempat Anda ingin menyimpan file sampel.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pada terminal atau command prompt, ketik perintah: `git clone <repoURL>` dimana `<repoURL>` mengacu pada URL yang Anda salin di langkah 2.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui nilai parameter.	Dalam salinan lokal repositori Anda, edit `parameter.json` file dan perbarui nilai parameter default sebagai berikut: • `pS3BucketName` — Nama bucket Amazon S3 untuk menyimpan dataset. Template akan membuat ember ini untuk Anda. Nama bucket harus unik secara global. • `pSourceFolder` — Nama folder di dalam bucket Amazon S3 yang akan digunakan untuk mengunggah file CSV sumber. • `pStageFolder` — Nama folder di dalam bucket Amazon S3 yang akan digunakan sebagai area pementasan selama proses berlangsung. • `pTransformFolder` — Nama folder di dalam bucket Amazon S3 yang akan digunakan untuk menyimpan dataset yang diubah dan dipartisi. • `pErrorFolder` — Folder di dalam bucket Amazon S3 tempat file CSV sumber	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akan dipindahkan jika tidak dapat divalidasi. • `pArchiveFolder` – Nama folder di dalam bucket Amazon S3 yang akan digunakan untuk mengarsipkan file CSV sumber. • `pEmailforNotification` – Alamat email yang valid untuk menerima success/error pemberitahuan. • `pPrefix`– String awalan yang akan digunakan dalam nama AWS Glue crawler. • `pDatasetSchema` – Skema kumpulan data yang akan divalidasi terhadap file sumber. Paket Cerberus Python digunakan untuk validasi dataset sumber. Untuk informasi lebih lanjut, lihat situs web Cerberus.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode sumber ke bucket Amazon S3.	Sebelum Anda menerapkan AWS CloudFormation template yang mengotomatiskan pipeline ETL, Anda harus mengemas file sumber untuk template dan mengunggahnya ke bucket Amazon S3. Untuk melakukan ini, jalankan AWS CLI perintah berikut dengan profil Anda yang telah dikonfigurasi sebelumnya: `aws cloudformation package --template-file template.yml --s3-bucket <bucket_name> --output-template-file packaged.template --profile <profile_name>` di mana: • `<bucket_name>` adalah nama bucket Amazon S3 yang ada di Wilayah AWS tempat Anda ingin menyebarkan tumpukan. Bucket ini digunakan untuk menyimpan paket kode sumber untuk CloudFormation template. • `<profile_name>` adalah AWS CLI profil valid yang telah Anda konfigurasi	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sebelumnya saat Anda mengatur AWS CLI.	

Buat tumpukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	Untuk menyebarkan AWS CloudFormation template, jalankan AWS CLI perintah berikut: `aws cloudformation deploy --stack-name <stack_name> --template-file packaged.template --parameter-overrides file://parameter.json --capabilities CAPABILITY_IAM --profile <profile_name>` di mana: • `<stack_name>` adalah pengidentifikasi unik untuk CloudFormation tumpukan. • `<profile-name>` adalah AWS CLI profil Anda yang telah dikonfigurasi sebelumnya.	Developer
Periksa kemajuan.	Di AWS CloudFormation konsol, periksa kemajuan pengembangan tumpukan.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Ketika statusnya`CREATE_COMPLETE` , tumpukan telah berhasil digunakan.	
Perhatikan nama AWS Glue database.	Tab Output untuk tumpukan menampilkan nama AWS Glue database. Nama kunci tersebut adalah `GlueDBOutput` .	Developer

Uji pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai pipa ETL.	1. Arahkan ke folder sumber (`source`, atau nama folder yang Anda tetapkan dalam `parameter.json` file) di dalam bucket Amazon S3. 2. Unggah contoh file CSV ke folder ini. (Repositori kode menyediakan file contoh yang disebut `Sample_Bank_Transaction_Raw_Dataset.csv` yang dapat Anda gunakan.) Mengunggah file akan memulai pipeline ETL melalui Step Functions. 3. Pada konsol Step Functions , periksa status pipeline ETL.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa dataset yang dipartisi.	Saat pipeline ETL selesai, verifikasi bahwa kumpulan data yang dipartisi tersedia di folder transformasi Amazon S3 (`transform` , atau nama folder yang Anda tetapkan dalam file). `parameter .json`	Developer
Periksa database yang dipartisi. AWS Glue	1. Di AWS Glue konsol, pilih AWS Glue database yang dibuat oleh tumpukan (ini adalah database yang Anda catat di epik sebelumnya). 2. Verifikasi bahwa tabel yang dipartisi tersedia di. AWS Glue Data Catalog	Developer
Jalankan kueri.	(Opsional) Gunakan Amazon Athena untuk menjalank an kueri ad-hoc pada database yang dipartisi dan diubah. Untuk petunjuk, lihat Menjalankan kueri SQL di Amazon Athena dalam dokumentasi. AWS	Analis basis data

Pemecahan masalah

Isu	Solusi
AWS Identity and Access Management (IAM) izin untuk AWS Glue pekerjaan dan crawler	Jika Anda lebih lanjut menyesuaikan AWS Glue pekerjaan atau crawler, pastikan untuk memberikan izin IAM yang sesuai dalam peran

Isu	Solusi
	IAM yang digunakan oleh AWS Glue pekerjaan , atau memberikan izin data ke. AWS Lake Formation Lihat informasi yang lebih lengkap dalam dokumentasi AWS.

Sumber daya terkait

Layanan AWS dokumentasi

- AWS Step Functions
- AWS Glue
- AWS Lambda
- Amazon S3
- Amazon SNS

Informasi tambahan

Diagram berikut menunjukkan AWS Step Functions alur kerja untuk pipeline ETL yang sukses, dari panel Inspector AWS Step Functions .

Diagram berikut menunjukkan AWS Step Functions alur kerja untuk pipeline ETL yang gagal karena kesalahan validasi input, dari panel Step Functions Inspector.

Lakukan analitik lanjutan menggunakan Amazon Redshift ML

Po Hong dan Chyanna Antonio, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, Anda dapat menggunakan machine learning Amazon Redshift (Amazon Redshift ML) untuk melakukan analitik ML pada data yang disimpan di kluster Amazon Redshift atau di Amazon Simple Storage Service (Amazon S3). Amazon Redshift MLmendukung pembelajaran yang diawasi, yang biasanya digunakan untuk analitik tingkat lanjut. Kasus penggunaan untuk Amazon Redshift MLmencakup peramalan pendapatan, deteksi penipuan kartu kredit, dan nilai seumur hidup pelanggan (CLV) atau prediksi churn pelanggan.

Amazon Redshift ML memudahkan pengguna database untuk membuat, melatih, dan menerapkan model ML dengan menggunakan perintah SQL standar. Amazon Redshift ML menggunakan Amazon SageMaker Autopilot untuk secara otomatis melatih dan menyetel model ML terbaik untuk klasifikasi atau regresi berdasarkan data Anda, sementara Anda mempertahankan kontrol dan visibilitas.

Semua interaksi antara Amazon Redshift, Amazon S3, dan Amazon diabstraksikan dan SageMaker diotomatisasi. Setelah model ML dilatih dan digunakan, model ini akan tersedia sebagai fungsi yang ditentukan pengguna (UDF) di Amazon Redshift dan dapat digunakan dalam kueri SQL.

Pola ini melengkapi model Create, train, dan deploy ML di Amazon Redshift menggunakan SQL dengan Amazon Redshift ML dari AWS Blog, dan Build, train, dan deploy model ML dengan tutorial SageMaker Amazon dari Getting Started Resource Center.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Data yang ada di tabel Amazon Redshift

Keterampilan

- Keakraban dengan istilah dan konsep yang digunakan oleh Amazon Redshift ML, termasuk pembelajaran mesin, pelatihan, dan prediksi. Untuk informasi selengkapnya tentang hal ini, lihat Melatih model ML dalam dokumentasi Amazon Machine Learning (Amazon ML).

- Pengalaman dengan penyiapan pengguna Amazon Redshift, manajemen akses, dan sintaks SQL standar. Untuk informasi selengkapnya tentang ini, lihat Memulai Amazon Redshift di dokumentasi Amazon Redshift.
- Pengetahuan dan pengalaman dengan Amazon S3 dan AWS Identity and Access Management (IAM).
- Pengalaman menjalankan perintah di AWS Command Line Interface (AWS CLI) juga bermanfaat tetapi tidak diperlukan.

Batasan

- Cluster Amazon Redshift dan bucket S3 harus berada di Wilayah AWS yang sama.
- Pendekatan pola ini hanya mendukung model pembelajaran yang diawasi seperti regresi, klasifikasi biner, dan klasifikasi multikelas.

Arsitektur

Langkah-langkah berikut menjelaskan cara kerja Amazon Redshift ML SageMaker untuk membangun, melatih, dan menerapkan model ML:

1. Amazon Redshift mengekspor data pelatihan ke bucket S3.
2. SageMaker Autopilot secara otomatis memproses data pelatihan.
3. Setelah `CREATE MODEL` pernyataan dipanggil, Amazon Redshift ML SageMaker digunakan untuk pelatihan.
4. SageMaker Autopilot mencari dan merekomendasikan algoritma ML dan parameter hiper optimal yang mengoptimalkan metrik evaluasi.
5. Amazon Redshift ML mendaftarkan model keluaran keluaran sebagai fungsi SQL di cluster Amazon Redshift.
6. Fungsi model ML dapat digunakan dalam pernyataan SQL.

Tumpukan teknologi

- Amazon Redshift
- SageMaker

- Amazon S3

Alat

- Amazon Redshift — Amazon Redshift adalah layanan pergudangan data tingkat perusahaan, skala petabyte, dan terkelola sepenuhnya.
- Amazon Redshift ML — Amazon Redshift machine learning (Amazon Redshift ML) adalah layanan berbasis cloud yang kuat yang memudahkan analis dan ilmuwan data dari semua tingkat keahlian untuk menggunakan teknologi ML.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet.
- Amazon SageMaker - SageMaker adalah layanan ML yang dikelola sepenuhnya.
- Amazon SageMaker Autopilot — SageMaker Autopilot adalah kumpulan fitur yang mengotomatiskan tugas-tugas utama dari proses pembelajaran mesin otomatis (AutoML).

Kode

Anda dapat membuat model ML yang diawasi di Amazon Redshift dengan menggunakan kode berikut:

```
"CREATE MODEL customer_churn_auto_model
FROM (SELECT state,
             account_length,
             area_code,
             total_charge/account_length AS average_daily_spend,
             cust_serv_calls/account_length AS average_daily_cases,
             churn
      FROM customer_activity
      WHERE  record_date < '2020-01-01'
     )
TARGET churn
FUNCTION ml_fn_customer_churn_auto
IAM_ROLE 'arn:aws:iam::XXXXXXXXXXX:role/Redshift-ML'
SETTINGS (
  S3_BUCKET 'your-bucket'
);")
```

> **Note**
>
> `SELECTStatus` dapat merujuk ke tabel reguler Amazon Redshift, tabel eksternal Amazon Redshift Spectrum, atau keduanya.

Epik

Siapkan kumpulan data pelatihan dan uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kumpulan data pelatihan dan uji.	Masuk ke AWS Management Console dan buka SageMaker konsol Amazon. Ikuti instruksi dari tutorial Build, train, dan deploy model machine learning untuk membuat file.csv atau Apache Parquet yang memiliki kolom label (pelatihan yang diawasi) dan tanpa header. **Note** Kami menyarankan Anda mengacak dan membagi kumpulan data mentah menjadi satu set pelatihan untuk pelatihan model (70 persen) dan satu set pengujian untuk evaluasi kinerja model (30 persen).	Ilmuwan data

Siapkan dan konfigurasikan tumpukan teknologi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan cluster Amazon Redshift.	Di konsol Amazon Redshift, buat cluster sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya tentang ini, lihat Membuat klaster di dokumentasi Amazon Redshift. **Important** Cluster Amazon Redshift harus dibuat dengan jalur pemeliharaan. `SQL_PREVIEW` Untuk informasi selengkapnya tentang trek pratinjau, lihat Memilih trek pemeliharaan klaster di dokumentasi Amazon Redshift.	DBA, arsitek Cloud
Buat bucket S3 untuk menyimpan data pelatihan dan artefak model.	Di konsol Amazon S3, buat bucket S3 untuk data pelatihan dan pengujian. Untuk informasi selengkapnya tentang membuat bucket S3, lihat Membuat bucket S3 dari AWS Quick Starts.	DBA, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important Pastikan cluster Amazon Redshift dan bucket S3 Anda berada di Wilayah yang sama.	
Buat dan lampirkan kebijakan IAM ke cluster Amazon Redshift.	Buat kebijakan IAM untuk mengizinkan klaster Amazon Redshift SageMaker mengakses dan Amazon S3. Untuk petunjuk dan langkah-langkahnya, lihat Penyiapan klaster untuk menggunakan Amazon Redshift MLdalam dokumentasi Amazon Redshift.	DBA, arsitek Cloud
Izinkan pengguna dan grup Amazon Redshift mengakses skema dan tabel.	Berikan izin untuk memungkinkan pengguna dan grup di Amazon Redshift mengakses skema dan tabel internal dan eksternal. Untuk langkah dan petunjuk, lihat Mengelola izin dan kepemilikan di dokumentasi Amazon Redshift.	DBA

Buat dan latih model ML di Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan latih model ML di Amazon Redshift.	Buat dan latih model MLmu di Amazon Redshift Ml. Untuk	Pengembang, Ilmuwan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi selengkapnya, lihat `CREATE MODEL` pernyataa n di dokumentasi Amazon Redshift.	

Lakukan inferensi dan prediksi batch di Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan inferensi menggunak an fungsi model ML yang dihasilkan.	Untuk informasi selengkapnya tentang melakukan inferensi menggunakan fungsi model ML yang dihasilkan, lihat Prediksi dalam dokumentasi Amazon Redshift.	Ilmuwan data, Pengguna intelijen bisnis

Sumber daya terkait

Siapkan kumpulan data pelatihan dan uji

- Membangun, melatih, dan menerapkan model pembelajaran mesin dengan Amazon SageMaker

Siapkan dan konfigurasikan tumpukan teknologi

- Membuat cluster Amazon Redshift
- Memilih trek pemeliharaan klaster Amazon Redshift
- Membuat bucket S3
- Menyiapkan klaster Amazon Redshift untuk menggunakan Amazon Redshift ML
- Mengelola izin dan kepemilikan di Amazon Redshift

Buat dan latih model ML di Amazon Redshift

- Buat pernyataan MODEL di Amazon Redshift

Lakukan inferensi dan prediksi batch di Amazon Redshift

- Prediksi di Amazon Redshift

Sumber daya lainnya

- Memulai dengan Amazon Redshift MI
- Membuat, melatih, dan menerapkan model ML di Amazon Redshift menggunakan SQL dengan Amazon Redshift ML
- Mitra Amazon Redshift
- Mitra kompetensi pembelajaran mesin AWS

Kueri tabel Amazon DynamoDB dengan SQL dengan menggunakan Amazon Athena

Gavin Perrie, Ajit Ambike, dan Brad Yates, Amazon Web Services

Ringkasan

Jika data Anda menyertakan sumber selain Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), Anda dapat menggunakan kueri gabungan untuk mengakses sumber data relasional, non-relasional, objek, atau kustom tersebut. Pola ini menunjukkan cara mengonfigurasi akses kueri federasi melalui Amazon Athena ke Amazon DynamoDB dengan menggunakan konektor sumber data SQL.

Dengan menggunakan pola ini, Anda dapat melakukan hal berikut:

- Kueri tabel DynamoDB dengan menggunakan SQL.
- Jalankan kueri SQL federasi di Athena dan gabungkan tabel DynamoDB dengan sumber data lain yang didukung.

Prasyarat dan batasan

Prasyarat

- Sebuah tabel DynamoDB.
- Kelompok kerja Athena diatur untuk menggunakan mesin Athena versi 2. Untuk instruksi, lihat dokumentasi Athena.
- Bucket S3 tempat `AthenaDynamoDBConnector` AWS Lambda fungsi dapat menumpahkan data. Bucket S3 dan fungsi Lambda harus berada di Wilayah AWS yang sama.

Jika ini adalah pertama kalinya Anda mengakses Athena, Anda memerlukan bucket S3 tambahan untuk digunakan sebagai lokasi hasil kueri. Untuk instruksi, lihat dokumentasi Athena.

Batasan

- Operasi tulis seperti INSERT INTO tidak didukung.

Versi produk

- Federasi Kueri Athena dirilis pada GitHub

Arsitektur

Arsitektur target

Diagram berikut menunjukkan aliran koneksi setelah pola ditetapkan. Pengguna terhubung ke Amazon Athena untuk memberikan kueri. Athena meneruskan kueri dan target ke fungsi Lambda konektor sumber data DynamoDB, yang mengambil dan mengembalikan data ke Athena. Jika sejumlah besar data dikembalikan, Athena menyimpan hasil sementara di ember tumpahan sebelum mengemas dan mengembalikan kumpulan data lengkap.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon Simple Storage Service (Amazon S3) dengan menggunakan SQL standar. Pola ini menggunakan Amazon Athena DynamoDB Connector, alat yang dibangun menggunakan Amazon Athena Query Federation SDK dan diinstal sebagai aplikasi melalui file. AWS Lambda AWS Serverless Application Repository
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Athena Query Federation.

Epik

Siapkan dan uji konektor sumber data DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan AthenaDynamoDBConnector aplikasi.	Untuk menyebarkan AthenaDynamoDBConnecter, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan pilih Wilayah AWS yang Anda gunakan untuk tabel DynamoDB dan ember tumpahan. 2. Buka Serverless Application Repository di. https://console.aws.amazon.com/serverlessrepo/ 3. Di panel navigasi, pilih Aplikasi yang Tersedia. 4. Untuk akses AWS Identity and Access Management (IAM), di bawah bilah pencarian, pilih kotak centang Tampilkan aplikasi yang membuat peran IAM kustom atau kebijakan sumber daya. 5. Cari dan pilih AthenaDynamoDBConnector, dan pastikan bahwa penulis yang terdaftar adalah Federasi Amazon Athena.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Di Pengaturan Aplikasi, masukkan nilai berikut: • SpillBucket- Lokasi dimana fungsi dapat menumpahkan data ke. • AthenaCatalogName- Nama fungsi Lambda yang akan dibuat. Nama tersebut juga akan digunakan sebagai Nama Sumber Data di Athena. 7. Pilih kotak centang untuk mengetahui pembuatan peran dan kebijakan IAM. 8. Pilih Deploy.	
Buat sumber data untuk Athena.	Untuk membuat sumber data, lakukan hal berikut: 1. Buka konsol Athena di https://console.aws.amazon.com/athena/. 2. Perluas panel navigasi, dan pilih Sumber data. 3. Pilih Buat sumber data. 4. Pilih Amazon DynamoDB. 5. Masukkan Nama Sumber Data. 6. Pilih fungsi Lambda yang Anda buat. 7. Tinjau detailnya, dan pilih Buat sumber data.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan Athena untuk menanyakan tabel DynamoDB.	Untuk menanyakan tabel DynamoDB, lakukan hal berikut: 1. Di konsol Athena, perluas panel navigasi, dan pilih Editor kueri. 2. Pada daftar tarik-turun Sumber data, pilih sumber data yang Anda buat. 3. Konfirmasikan bahwa tabel DynamoDB tercantum di bawah Tabel. 4. Jalankan kueri.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Kueri gagal dengan `GENERIC_INTERNAL_ERROR: The bucket is in this region: <region>`.	Pastikan bahwa ember tumpahan Athena dan fungsi Lambda dibuat dalam hal yang sama. Wilayah AWS
Sumber data yang baru dibuat tidak terlihat di konsol Athena.	Katalog data Athena bersifat regional. Pastikan `AthenaDynamoDBConnector` telah dikerahkan di Wilayah tempat Anda mencoba menggunakan Athena.
Anda tidak dapat menjalankan kueri terhadap sumber data yang baru dibuat.	Periksa apakah Lokasi Hasil Kueri telah ditetapkan.

Sumber daya terkait

- Konektor DynamoDB Amazon Athena
- Pertanyaan Federasi Amazon Athena

Akses, kueri, dan gabungkan tabel Amazon DynamoDB menggunakan Athena

Moinul Al-Mamun, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara mengatur koneksi antara Amazon Athena dan Amazon DynamoDB dengan menggunakan konektor Amazon Athena DynamoDB. Konektor menggunakan fungsi AWS Lambda untuk menanyakan data di DynamoDB. Anda tidak perlu menulis kode apa pun untuk mengatur koneksi. Setelah koneksi dibuat, Anda dapat dengan cepat mengakses dan menganalisis tabel DynamoDB dengan menggunakan Athena Federated Query untuk menjalankan perintah SQL dari Athena. Anda juga dapat menggabungkan satu atau beberapa tabel DynamoDB satu sama lain atau ke sumber data lain, seperti Amazon Redshift atau Amazon Aurora.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan izin untuk mengelola tabel DynamoDB, sumber Data Athena, Lambda, dan AWS Identity and Access Management (IAM)
- Bucket Amazon Simple Storage Service (Amazon S3) tempat Athena dapat menyimpan hasil kueri
- Bucket S3 tempat Konektor DynamoDB Athena dapat menyimpan data dalam jangka pendek
- Wilayah AWS yang mendukung mesin Athena versi 2
- Izin IAM untuk mengakses Athena dan bucket S3 yang diperlukan
- Konektor Amazon Athena DynamoDB, diinstal

Batasan

Ada biaya untuk menanyakan tabel DynamoDB. Ukuran tabel melebihi beberapa gigabyte (GBs) dapat menimbulkan biaya tinggi. Kami menyarankan Anda mempertimbangkan biaya sebelum melakukan operasi SCAN tabel penuh. Untuk informasi selengkapnya, lihat harga Amazon DynamoDB. Untuk mengurangi biaya dan mencapai kinerja tinggi, kami sarankan Anda selalu menggunakan LIMIT dalam kueri Anda (misalnya,`SELECT * FROM table1 LIMIT 10`). Selain itu, sebelum Anda melakukan kueri JOIN atau GROUP BY di lingkungan produksi, pertimbangkan ukuran

tabel Anda. Jika tabel Anda terlalu besar, pertimbangkan opsi alternatif seperti memigrasikan tabel ke Amazon S3.

Arsitektur

Diagram berikut menunjukkan bagaimana pengguna dapat menjalankan query SQL pada tabel DynamoDB dari Athena.

Diagram menunjukkan alur kerja berikut:

1. Untuk menanyakan tabel DynamoDB, pengguna menjalankan kueri SQL dari Athena.
2. Athena memulai fungsi Lambda.
3. Fungsi Lambda menanyakan data yang diminta dalam tabel DynamoDB.
4. DynamoDB mengembalikan data yang diminta ke fungsi Lambda. Kemudian, fungsi mentransfer hasil kueri ke pengguna melalui Athena.
5. Fungsi Lambda menyimpan data di bucket S3.

Tumpukan teknologi

- Amazon Athena
- Amazon DynamoDB
- Amazon S3
- AWS Lambda

Alat

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar.
- Amazon Athena DynamoDB Connector adalah alat AWS yang memungkinkan Athena terhubung dengan DynamoDB dan mengakses tabel Anda dengan menggunakan kueri SQL.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan dapat diskalakan.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Epik

Buat contoh tabel DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel sampel pertama.	1. Masuk ke AWS Management Console dan buka konsol DynamoDB. 2. Pilih Buat tabel. 3. Untuk nama Tabel, masukkan dydbtable1. 4. Untuk kunci Partisi, masukkan PK1. 5. Untuk tombol Sortir, masukkan SK1. 6. Di bagian Pengaturan tabel, pilih Sesuaikan pengaturan. 7. Di bagian kelas Tabel, pilih DynamoDB Standard. 8. Di bagian Pengaturan kapasitas baca/tulis, untuk mode Kapasitas, pilih Sesuai permintaan. 9. Di bagian Enkripsi saat istirahat, pilih Dimiliki oleh Amazon DynamoDB. 10. Pilih Buat tabel.	Developer
Masukkan data sampel ke dalam tabel pertama.	1. Buka Konsol DynamoDB. 2. Di panel navigasi, pilih Tabel, lalu pilih tabel Anda di kolom Nama. 3. Pilih Tindakan, lalu pilih Buat item.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Pilih tampilan JSON.
	5. Di bilah judul editor Atribut, matikan Lihat DynamoDB JSON.
	6. Di editor Atribut, masukkan contoh data berikut satu per satu:
	
	`{ "PK1": "1234", "SK1": "info", "Salary": "5000" }`
	
	`{ "PK1": "1235", "SK1": "info", "Salary": "5200" }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel sampel kedua.	1. Buka Konsol DynamoDB. 2. Pilih Buat tabel. 3. Untuk nama Tabel, masukkan dydbtable2. 4. Untuk kunci Partisi, masukkan PK2. 5. Untuk tombol Sortir, masukkan SK2. 6. Di bagian Pengaturan tabel, pilih Sesuaikan pengaturan. 7. Di bagian kelas Tabel, pilih DynamoDB Standard. 8. Di bagian Pengaturan kapasitas baca/tulis, untuk mode Kapasitas, pilih Sesuai permintaan. 9. Di bagian Enkripsi saat istirahat, pilih Dimiliki oleh Amazon DynamoDB. 10. Pilih Buat tabel.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Masukkan data sampel ke dalam tabel kedua.	1. Buka Konsol DynamoDB.
	2. Di panel navigasi, pilih Tabel, lalu pilih tabel Anda di kolom Nama.
	3. Pilih Tindakan, lalu pilih Buat item.
	4. Di bilah judul editor Atribut, matikan Lihat DynamoDB JSON.
	5. Di editor Atribut, masukkan contoh data berikut satu per satu:
	
	`{ "PK2": "1234", "SK2": "bonus", "Bonus": "500" }`
	
	`{ "PK2": "1235", "SK2": "bonus", "Bonus": "1000" }`
		Developer

Buat sumber data di Athena untuk DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan konektor sumber data.	Buat sumber data untuk DynamoDB, lalu buat fungsi Lambda untuk terhubung ke sumber data tersebut.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Masuk ke AWS Management Console dan buka konsol Athena. 2. Di panel navigasi, pilih Sumber data, lalu pilih Buat sumber data. 3. Pilih sumber data Amazon DynamoDB, lalu pilih Berikutnya. 4. Di bagian Rincian sumber data, untuk nama sumber data, masukkan TestDynamoDB. 5. Di bagian Detail koneksi, pilih fungsi Lambda yang sudah digunakan atau pilih fungsi Buat Lambda jika Anda tidak memiliki fungsi Lambda untuk digunakan untuk pola ini. Catatan: Untuk informasi selengkapnya tentang membuat fungsi Lambda, lihat Memulai Lambda di Panduan Pengembang Lambda. 6. (Opsional) Jika Anda memilih fungsi Create Lambda, maka Anda harus mengonfigurasi CloudFormation template AWS yang disertakan oleh aplikasi Java sebelum menerapkan tumpukan itu. Template	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	termasuk ApplicationName, SpillBucket, AthenaCatalogName, dan pengaturan aplikasi lainnya. Catatan: Setelah Anda menerapkan aplikasi berbasis Java ini, tumpukan membuat fungsi Lambda yang memungkinkan Athena berkomunikasi dengan DynamoDB. Ini membuat tabel Anda dapat diakses melalui perintah SQL. 7. Terapkan fungsi Lambda Anda. 8. Pilih Berikutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa fungsi Lambda dapat mengakses bucket tumpahan S3.	1. Buka Konsol Lambda. 2. Di panel navigasi, pilih Fungsi, lalu pilih fungsi yang Anda buat sebelumnya. 3. Pilih tab Konfigurasi. 4. Di panel kiri, pilih variabel Lingkungan, dan kemudian konfirmasikan bahwa nilai untuk kuncinya adalah`spill_bucket` . 5. Di panel kiri, pilih Izin, lalu di bagian Peran eksekusi, pilih peran IAM terlampir. Catatan: Anda diarahkan ke peran IAM yang dilampirkan ke fungsi Lambda Anda di konsol IAM. 6. Konfirmasikan bahwa Anda memiliki izin menulis di `spill_bucket` bucket. Jika Anda mengalami kesalahan, lihat bagian Informasi tambahan dalam pola ini untuk panduan.	Developer

Akses tabel DynamoDB dari Athena

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kueri tabel DynamoDB.	1. Masuk ke AWS Management Console dan buka konsol Athena. 2. Di panel navigasi, pilih Sumber data, lalu pilih Buat sumber data. 3. Di panel navigasi, pilih Editor kueri. 4. Pada tab Editor, di bagian Data, untuk Sumber data, pilih sumber data Anda untuk Sumber data. 5. Untuk Database, pilih database Anda. 6. Untuk Query 1, masukkan query berikut: `SELECT * FROM dydbtable1 t1;` 7. Pilih Jalankan, lalu verifikasi output dalam tabel. 8. Untuk Query 2, masukkan query berikut: `SELECT * FROM dydbtable2 t2;` 9. Pilih Jalankan, lalu verifikasi output dalam tabel.	Developer
Bergabunglah dengan dua tabel DynamoDB.	DynamoDB adalah penyimpanan data NoSQL dan tidak mendukung operasi gabungan SQL. Akibatnya, Anda harus melakukan operasi gabungan pada dua tabel DynamoDB:	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Pilih ikon plus untuk membuat kueri lain. 2. Untuk Query 3, masukkan query berikut: `SELECT pk1, salary, bonus FROM dydbtable1 t1 JOIN dydbtable2 t2 ON t1.pk1 = t2.pk2;`	

Sumber daya terkait

- Konektor DynamoDB Amazon Athena (AWS Labs)
- Kueri sumber data apa pun dengan kueri federasi baru Amazon Athena (AWS Big Data Blog)
- Referensi versi mesin Athena (Panduan Pengguna Athena)
- Sederhanakan ekstraksi dan analisis data Amazon DynamoDB dengan menggunakan AWS Glue dan Amazon Athena (AWS Database Blog)

Informasi tambahan

Jika Anda menjalankan kueri di Athena dengan `spill_bucket` dalam `{bucket_name}/folder_name/` format, maka Anda dapat menerima pesan galat berikut:

```
"GENERIC_USER_ERROR: Encountered an exception[java.lang.RuntimeException] from your
 LambdaFunction[arn:aws:lambda:us-east-1:xxxxxx:function:testdynamodb] executed in
 context[retrieving meta-data] with message[You do NOT own the spill bucket with the
 name: s3://amzn-s3-demo-bucket/athena_dynamodb_spill_data/]
This query ran against the "default" database, unless qualified by the query. Please
 post the error message on our forum  or contact customer support with Query Id:
 [query-id]"
```

Untuk mengatasi kesalahan ini, perbarui variabel lingkungan fungsi Lambda `spill_bucket` ke`{bucket_name_only}`, lalu perbarui kebijakan IAM Lambda berikut untuk akses tulis bucket:

```
{
             "Action": [
                 "s3:GetObject",
                 "s3:ListBucket",
                 "s3:GetBucketLocation",
                 "s3:GetObjectVersion",
                 "s3:PutObject",
                 "s3:PutObjectAcl",
                 "s3:GetLifecycleConfiguration",
                 "s3:PutLifecycleConfiguration",
                 "s3:DeleteObject"
             ],
             "Resource": [
                 "arn:aws:s3:::spill_bucket",
                 "arn:aws:s3:::spill_bucket/*"
             ],
             "Effect": "Allow"
         }
```

Atau, Anda dapat menghapus konektor sumber data Athena yang Anda buat sebelumnya, dan membuatnya kembali dengan menggunakan hanya untuk. `{bucket_name}` `spill_bucket`

Siapkan ruang data minimum yang layak untuk berbagi data antar organisasi

Ramy Hcini, Michael Miller, dan Jorge Hernandez Suarez, Amazon Web Services

Ismail Abdellaoui dan Malte Gasseling, Berpikir-itu

Ringkasan

Ruang data adalah jaringan federasi untuk pertukaran data dengan kepercayaan dan kontrol atas data seseorang sebagai prinsip inti. Mereka memungkinkan organisasi untuk berbagi, bertukar, dan berkolaborasi pada data dalam skala besar dengan menawarkan solusi yang hemat biaya dan teknologi-agnostik.

Ruang data memiliki potensi untuk secara signifikan mendorong upaya untuk masa depan yang berkelanjutan dengan menggunakan pemecahan masalah berbasis data dengan end-to-end pendekatan yang melibatkan semua pemangku kepentingan yang relevan.

Pola ini memandu Anda melalui contoh bagaimana dua perusahaan dapat menggunakan teknologi ruang data di Amazon Web Services (AWS) untuk mendorong strategi pengurangan emisi karbon mereka ke depan. Dalam skenario ini, perusahaan X menyediakan data emisi karbon, yang dikonsumsi perusahaan Y. Lihat bagian Informasi tambahan untuk detail spesifikasi ruang data berikut:

- Peserta
- Kasus bisnis
- Otoritas ruang data
- Komponen ruang data
- Layanan ruang data
- Data yang akan dipertukarkan
- Model Data
- Konektor EDC Traktus-X

Pola tersebut mencakup langkah-langkah untuk hal-hal berikut:

- Menyebarkan infrastruktur yang dibutuhkan untuk ruang data dasar dengan dua peserta berjalan AWS.

- Bertukar data intensitas emisi karbon dengan menggunakan konektor dengan cara yang aman.

Pola ini menerapkan cluster Kubernetes yang akan meng-host konektor ruang data dan layanannya melalui Amazon Elastic Kubernetes Service (Amazon EKS).

Pesawat kontrol Eclipse Dataspace Components (EDC) dan bidang data keduanya digunakan di Amazon EKS. Bagan Helm Tractus-X resmi menyebarkan layanan PostgreSQL dan Vault sebagai dependensi. HashiCorp

Selain itu, layanan identitas digunakan di Amazon Elastic Compute Cloud EC2 (Amazon) untuk mereplikasi skenario kehidupan nyata dari ruang data minimum yang layak (MVDS).

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS untuk menyebarkan infrastruktur yang Anda pilih Wilayah AWS
- Pengguna AWS Identity and Access Management (IAM) dengan akses ke Amazon S3 yang akan digunakan sementara sebagai pengguna teknis (Konektor EDC saat ini tidak mendukung penggunaan peran. Kami menyarankan Anda membuat satu pengguna IAM khusus untuk demo ini dan bahwa pengguna ini akan memiliki izin terbatas yang terkait dengannya.)
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi dalam pilihan Anda Wilayah AWS
- AWS kredensi keamanan
- eksctl di workstation Anda
- Git di workstation Anda
- kubectl
- Helm
- Tukang pos
- AWS Certificate Manager Sertifikat (ACM) SSL/TLS
- Nama DNS yang akan menunjuk ke Application Load Balancer (nama DNS harus dicakup oleh sertifikat ACM)
- HashiCorp Vault (Untuk informasi tentang penggunaan AWS Secrets Manager untuk mengelola rahasia, lihat bagian Informasi tambahan.)

Versi produk

- AWS CLI versi 2+
- Koleksi Tukang Pos v2.1

Batasan

- Pemilihan konektor - Penerapan ini menggunakan konektor berbasis EDC. Namun, pastikan untuk mempertimbangkan kekuatan dan fungsionalitas konektor EDC dan FIWARE True untuk membuat keputusan berdasarkan informasi yang sesuai dengan kebutuhan spesifik penerapan.
- Pembuatan konektor EDC - Solusi penerapan yang dipilih bergantung pada bagan Helm Konektor EDC Tractus-X, opsi penerapan yang mapan dan diuji secara ekstensif. Keputusan untuk menggunakan bagan ini didorong oleh penggunaan umum dan penyertaan ekstensi penting dalam build yang disediakan. Meskipun PostgreSQL HashiCorp dan Vault adalah komponen default, Anda memiliki fleksibilitas untuk menyesuaikan build konektor Anda sendiri jika diperlukan.
- Akses kluster pribadi - Akses ke kluster EKS yang diterapkan dibatasi untuk saluran pribadi. Interaksi dengan cluster dilakukan secara eksklusif melalui penggunaan `kubectl` dan IAM. Akses publik ke sumber daya cluster dapat diaktifkan dengan menggunakan penyeimbang beban dan nama domain, yang harus diimplementasikan secara selektif untuk mengekspos layanan tertentu ke jaringan yang lebih luas. Namun, kami tidak menyarankan untuk menyediakan akses publik.
- Fokus keamanan - Penekanan ditempatkan pada abstraksi konfigurasi keamanan ke spesifikasi default sehingga Anda dapat berkonsentrasi pada langkah-langkah yang terlibat dalam pertukaran data konektor EDC. Meskipun pengaturan keamanan default dipertahankan, sangat penting untuk mengaktifkan komunikasi yang aman sebelum Anda mengekspos cluster ke jaringan publik. Tindakan pencegahan ini memastikan fondasi yang kuat untuk penanganan data yang aman.
- Biaya infrastruktur — Estimasi biaya infrastruktur dapat ditemukan dengan menggunakan. AWS Pricing Calculator Perhitungan sederhana menunjukkan bahwa biaya bisa mencapai 162,92 USD per bulan untuk infrastruktur yang digunakan.

Arsitektur

Arsitektur MVDS terdiri dari dua virtual private cloud (VPCs), satu untuk layanan identitas Dynamic Attribute Provisioning System (DAPS) dan satu untuk Amazon EKS.

Arsitektur DAPS

Diagram berikut menunjukkan DAPS berjalan pada EC2 instance yang dikendalikan oleh grup Auto Scaling. Application Load Balancer dan tabel rute mengekspos server DAPS. Amazon Elastic File System (Amazon EFS) menyinkronkan data di antara instans DAPS.

Arsitektur Amazon EKS

Ruang data dirancang untuk menjadi solusi teknologi-agnostik, dan beberapa implementasi ada. Pola ini menggunakan kluster Amazon EKS untuk menyebarkan komponen teknis ruang data. Diagram berikut menunjukkan penyebaran cluster EKS. Node pekerja dipasang di subnet pribadi. Pod Kubernetes mengakses Amazon Relational Database Service (Amazon RDS) untuk instance PostgreSQL yang juga ada di subnet pribadi. Pod Kubernetes menyimpan data bersama di Amazon S3.

Alat

AWS layanan

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di file AWS Cloud.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh EC2 instans, kontainer, dan alamat IP di satu atau beberapa Availability Zone.

Alat-alat lainnya

- eksctl adalah utilitas baris perintah untuk membuat dan mengelola cluster Kubernetes di Amazon EKS.
- Git adalah sistem kontrol versi terdistribusi open source.
- HashiCorp Vault menyediakan penyimpanan aman dengan akses terkontrol untuk kredensyal dan informasi sensitif lainnya.
- Helm adalah pengelola paket open source untuk Kubernetes yang membantu Anda menginstal dan mengelola aplikasi di klaster Kubernetes Anda.
- kubectl adalah antarmuka baris perintah yang membantu Anda menjalankan perintah terhadap klaster Kubernetes.
- Postman adalah platform API.

Repositori kode

File YAMM konfigurasi Kubernetes dan skrip Python untuk pola ini tersedia di repositori. GitHub aws-patterns-edc Pola ini juga menggunakan repositori Tractus-X EDC.

Praktik terbaik

Amazon EKS dan isolasi infrastruktur peserta

Namespace di Kubernetes akan memisahkan infrastruktur penyedia X perusahaan dari infrastruktur konsumen Y perusahaan dalam pola ini. Untuk informasi selengkapnya, lihat Panduan Praktik Terbaik EKS.

Dalam situasi yang lebih realistis, setiap peserta akan memiliki cluster Kubernetes terpisah yang berjalan sendiri. Akun AWS Infrastruktur bersama (DAPS dalam pola ini) akan dapat diakses oleh peserta ruang data sementara benar-benar terpisah dari infrastruktur peserta.

Epik

Siapkan lingkungan, dan sediakan cluster dan EC2 instance EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori ke workstation Anda, jalankan perintah berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/Think-iT-Labs/aws-patterns-edc` Workstation harus memiliki akses ke Anda Akun AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyediakan klaster Kubernetes dan menyiapkan namespace.	Untuk menerapkan cluster EKS default yang disederhanakan di akun Anda, jalankan `eksctl` perintah berikut di workstation tempat Anda mengkloning repo:
	`eksctl create cluster`
	Perintah membuat VPC dan subnet pribadi dan publik yang menjangkau tiga Availability Zone yang berbeda. Setelah lapisan jaringan dibuat, perintah membuat dua `m5.large` EC2 instance dalam grup Auto Scaling.
	Untuk informasi lebih lanjut dan contoh keluaran, lihat panduan eksctl.
	Setelah Anda menyediakan kluster pribadi, tambahkan kluster EKS baru ke konfigurasi Kubernetes lokal Anda dengan menjalankan perintah berikut:
	`aws eks update-kubeconfig --name <EKS CLUSTER NAME> --region <AWS REGION>`
	Pola ini menggunakan `eu-west-1` Wilayah AWS
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk menjalankan semua perintah. Namun, Anda dapat menjalankan perintah yang sama sesuai keinginan Anda Wilayah AWS. Untuk mengonfirmasi bahwa node EKS Anda sedang berjalan dan berada dalam keadaan siap, jalankan perintah berikut: `kubectl get nodes`	

Siapkan ruang nama.	Untuk membuat ruang nama untuk penyedia dan konsumen, jalankan perintah berikut:
	
	`kubectl create ns provider`
	`kubectl create ns consumer`
	
	Dalam pola ini, penting untuk menggunakan `provider` dan `consumer` sebagai ruang nama agar sesuai dengan konfigurasi di langkah berikutnya.
		DevOps insinyur

Menyebarkan layanan identitas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan DAPS dengan menggunakan. AWS CloudFormation	Untuk kemudahan mengelola operasi DAPS, server DAPS diinstal pada EC2 instance. Untuk menginstal DAPS, gunakan AWS CloudFormation template. Anda akan memerlukan sertifikat ACM dan nama DNS dari bagian Prasyarat. Template menyebarkan dan mengonfigurasi yang berikut ini: • Penyeimbang Beban Aplikasi • Nama grup Auto Scaling • EC2 instance yang dikonfigurasi dengan data pengguna untuk menginstal semua paket yang diperlukan • Peran IAM • DAPS Anda dapat menerapkan AWS CloudFormation template dengan masuk ke Konsol Manajemen AWS dan menggunakan AWS CloudFormation konsol. Anda juga dapat menerapkan template dengan menggunak	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an AWS CLI perintah seperti berikut ini: `aws cloudformation create-stack --stack-name daps \ --template-body file://aws-patterns-edc/cloudformation.yml --parameters \ ParameterKey=CertificateARN,ParameterValue=<ACM Certificate ARN> \ ParameterKey=DNSName,ParameterValue=<DNS name> \ ParameterKey=InstanceType,ParameterValue=<EC2 instance type> \ ParameterKey=EnvironmentName,ParameterValue=<Environment Name> --capabilities CAPABILITY_NAMED_IAM`  Nama lingkungan adalah pilihan Anda sendiri. Sebaiknya gunakan istilah yang bermakna, seperti`DapsInfrastructure`, karena akan tercermin dalam tag AWS sumber daya.  Untuk pola ini, `t3.small` cukup besar untuk menjalank	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an alur kerja DAPS, yang memiliki tiga kontainer Docker. Template menyebarkan EC2 instance di subnet pribadi. Ini berarti bahwa instans tidak dapat diakses secara langsung melalui SSH (Secure Shell) dari internet. Instans disediakan dengan peran IAM yang diperlukan dan AWS Systems Manager Agen untuk memungkinkan akses ke instans yang berjalan melalui Session Manager, kemampuan. AWS Systems Manager Kami merekomendasikan menggunakan Session Manager untuk akses. Atau, Anda dapat menyediakan host benteng untuk memungkin kan akses SSH dari internet. Saat menggunakan pendekata n host bastion, EC2 instance mungkin memerlukan beberapa menit lagi untuk mulai berjalan. Setelah AWS CloudForm ation template berhasil di-deploy, arahkan nama DNS ke nama Application Load Balancer DNS Anda. Untuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	mengonfirmasi, jalankan perintah berikut:
	`dig <DNS NAME>`
	Output harus serupa dengan yang berikut ini:
	`; <<>> DiG 9.16.1-Ubuntu <<>> edc-pattern.think-it.io`
	`;; global options: +cmd`
	`;; Got answer:`
	`;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42344`
	`;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1`
	`;; OPT PSEUDOSECTION:`
	`; EDNS: version: 0, flags:; udp: 65494`
	`;; QUESTION SECTION:`
	`;edc-pattern.think-it.io. IN A`
	`;; ANSWER SECTION:`
	`edc-pattern.think-it.io. 276 IN CNAME daps-alb-iap9zmwy3kn8-1328773120.eu-west-1.elb.amazonaws.com.`
	`daps-alb-iap9zmwy3kn8-1328773120.eu-west-1.elb.amazonaw`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`s.com. 36 IN A 52.208.240.129 daps-alb-iap9zmwy3kn8-1328773120.eu-west-1.elb.amazonaws.com. 36 IN A 52.210.155.124`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Daftarkan konektor peserta ke layanan DAPS.	Dari dalam salah satu EC2 contoh yang disediakan untuk DAPS, daftarkan peserta:
	1. Jalankan skrip yang tersedia pada EC2 instance dengan menggunakan pengguna root:
	`cd /srv/mvds/omejdn-daps`
	2. Daftarkan penyedia:
	`bash scripts/register_connector.sh <provider_name>`
	3. Daftarkan konsumen:
	`bash scripts/register_connector.sh <consumer_name>`
	Pilihan nama tidak memengaruhi langkah selanjutnya. Kami merekomendasikan menggunakan salah satu `provider` dan `consumer` atau `companyx` dan`companyy`.
	Perintah pendaftaran juga akan secara otomatis mengkonfigurasi layanan DAPS dengan informasi
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang diperlukan diambil dari sertifikat dan kunci yang dibuat. Saat Anda masuk ke server DAPS, kumpulkan informasi yang diperlukan untuk langkah-langkah selanjutnya dalam penginstalan: 1. Dari `omejdn-daps/config/clients.yml` dapatkan `client id` untuk penyedia dan konsumen. `client id`Nilainya adalah string panjang digit heksadesimal. 2. Dari `omejdn-daps/keys` direktori , salin isi`consumer.cert` ,, `consumer.key` `provider.cert` , dan `provider.key` file. Kami merekomendasikan untuk menyalin dan menempelkan teks ke file bernama serupa yang diawali `daps-` di workstation Anda. Anda harus memiliki klien IDs untuk penyedia dan konsumen dan harus memiliki empat file di direktori kerja Anda di workstation Anda:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Nama file sumber `consumer.cert` menjadi nama `daps-consumer.cert` file workstation. • Nama file sumber `consumer.key` menjadi nama `daps-consumer.key` file workstation. • Nama file sumber `provider.cert` menjadi nama `daps-provider.cert` file workstation. • Nama file sumber `provider.key` menjadi nama `daps-provider.key` file workstation.	

Menyebarkan konektor peserta

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori EDC Tractus-X dan gunakan versi 0.4.1.	Build konektor EDC Tractus-X membutuhkan layanan PostgreSQL (database aset) dan HashiCorp Vault (manajemen rahasia) untuk digunakan dan tersedia. Ada banyak versi bagan Tractus-X EDC Helm yang berbeda. Pola ini menentukan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	versi 0.4.1 karena menggunak an server DAPS.
	
	Versi terbaru menggunak an Managed Identity Wallet (MIW) dengan implement asi terdistribusi dari layanan identitas.
	
	Di workstation tempat Anda membuat dua ruang nama Kubernetes, kloning repositor i tractusx-edc, dan periksa cabangnya. `release/0 .4.1`
	
	`git clone https://g ithub.com/eclipse- tractusx/tractusx-edc`
	
	`cd tractusx-edc`
	
	`git checkout release/0 .4.1`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan bagan Helm EDC Tractus-X.	Ubah konfigurasi template bagan Tractus-X Helm untuk memungkinkan kedua konektor berinteraksi bersama.
	
	Untuk melakukan ini, Anda akan menambahkan namespace ke nama DNS layanan sehingga dapat diselesaikan oleh layanan lain di cluster. Modifikasi ini harus dilakukan pada `charts/tractusx-connector/templates/_helpers.tpl` file. Pola ini menyediakan versi modifikasi akhir dari file ini untuk Anda gunakan. Salin dan letakkan di `daps` bagian file`charts/tractusx-connector/templates/_helpers.tpl` .
	
	Pastikan untuk mengomentari semua dependensi DAPS di: `charts/tractusx-connector/Chart.yaml`
	
	`dependencies:`
	`  # IDS Dynamic Attribute Provisioning Service (IAM)`
	`  #  - name: daps`
	`  # version: 0.0.1`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> # repository:
	 "file://./subcharts/
	omejdn"
	  # alias: daps
	  # condition:
	 install.daps</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan konektor untuk menggunakan PostgreSQL di Amazon RDS.	(Opsional) Instans Amazon Relational Database Service (Amazon RDS) tidak diperlukan dalam pola ini. Namun, kami sangat menyarankan menggunakan Amazon RDS atau Amazon Aurora karena mereka menyediakan fitur seperti ketersediaan tinggi dan cadangan dan pemulihan.
	
	Untuk mengganti PostgreSQL di Kubernetes dengan Amazon RDS, lakukan hal berikut:
	
	1. Menyediakan Amazon RDS untuk instance PostgreSQL.
	2. Di `Chart.yaml` , komentari `PostgreSQL` bagian tersebut.
	3. Di `provider_values.yml` dan `consumer_values.yml` , konfigurasikan `postgresql` bagian sebagai berikut:
	
	`postgresql:`
	`  auth:`
	`  database: edc`
	`  password: <RDS PASSWORD>`
	`  username: <RDS Username>`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> jdbcUrl: jdbc:post
	gresql://<RDS DNS
	 NAME>:5432/edc
	  username: <RDS
	 Username>
	  password: <RDS
	 PASSWORD>
	  primary:
	  persistence:
	  enabled: false
	  readReplicas:
	  persistence:
	  enabled: false</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan terapkan konektor penyedia dan layanannya.	Untuk mengkonfigurasi konektor penyedia dan layanannya, lakukan hal berikut: 1. Untuk mengunduh `provider_edc.yaml` file dari `edc_helm_configs` direktori ke folder bagan Helm saat ini, jalankan perintah berikut: `wget -q https://raw.githubusercontent.com/Think-iT-Labs/aws-patterns-edc/main/edc_helm_configs/provider_edc.yaml> -P charts/tractusx-connector/` 2. Ganti variabel berikut (juga ditandai dalam file) dengan nilainya: • `CLIENT_ID` ID yang dihasilkan oleh DAPS. `CLIENT_ID` Seharusnya ada `/srv/mvds/omejdn-daps/config/clients.yml/config/clients.yml` di server DAPS. Ini harus string karakter heksadesimal.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• `DAPS_URL` URL server DAPS. Itu harus `https://{DNS name}` menggunakan nama DNS yang Anda atur ketika Anda menjalankan AWS CloudFormation template.
	• `VAULT_TOKEN` Token yang akan digunakan untuk otorisasi Vault. Pilih nilai apa pun.
	• `vault.fullnameOverride – vault-provider` .
	• `vault.hashicorp.url – http://vault-provider:8200/` .
	
	Nilai sebelumnya mengasumsikan bahwa nama penyebaran dan nama namespace adalah penyedia.
	3. Untuk menjalankan bagan Helm dari workstation Anda, gunakan perintah berikut:
	
	`cd charts/tractusx-connector`
	
	`helm dependency build`
	
	`helm upgrade --install provider ./ -f`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`provider_edc.yaml -n provider`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan sertifikat dan kunci ke brankas penyedia.	Untuk menghindari kebingungan, buat sertifikat berikut di luar `tractusx-edc/charts` direktori.
	
	Misalnya, jalankan perintah berikut untuk mengubah ke direktori home Anda:
	
	`cd ~`
	
	Anda sekarang perlu menambahkan rahasia yang dibutuhkan oleh penyedia ke dalam lemari besi.
	
	Nama-nama rahasia di dalam brankas adalah nilai kunci di `secretNames:` bagian `provider_edc.yml` file. Secara default, mereka dikonfigurasi sebagai berikut:
	
	`secretNames:`
	`transferProxyTokenSignerPrivateKey: transfer-proxy-token-signer-private-key`
	`transferProxyTokenSignerPublicKey: transfer-proxy-token-signer-public-key`
	`transferProxyTokenEncryptionAesKey: transfer-`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>proxy-token-encryption-aes-key
	  dapsPrivateKey: daps-private-key
	  dapsPublicKey: daps-public-key</pre>
	Kunci Advanced Encryption Standard (AES), kunci pribadi, kunci publik, dan sertifikat yang ditandatangani sendiri dibuat pada awalnya. Ini kemudian ditambahkan sebagai rahasia ke lemari besi.
	
	Selanjutnya, direktori ini harus berisi `daps-provider.cert` dan `daps-provider.key` file yang Anda salin dari server DAPS.
	
	1. Jalankan perintah berikut:
	
	<pre># generate a private key
	openssl ecparam -name prime256v1 -genkey -noout -out provider-private-key.pem
	# generate corresponding public key
	openssl ec -in provider-private-key.pem -pubout -out provider-public-key.pem</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>
create a self-sign
ed certificate
openssl req -new -
x509 -key provider-
private-key.pem -out
provider-cert.pem -
days 360
generate aes key
openssl rand -base64
32 > provider-
aes.key
</pre>	
	2. Sebelum menambahk an rahasia ke brankas, konversikan dari beberapa baris menjadi satu baris dengan mengganti jeda baris dengan\n:	
	<pre>
cat provider-private-
key.pem | sed 's/$/\
\\\n/'|tr -d '\\n' >
provider-private-k
ey.pem.line
cat provider-public-
key.pem | sed 's/$/\
\\\n/'|tr -d '\\n' >
provider-public-ke
y.pem.line
cat provider-cert.pem
| sed 's/$/\\\\
n/'|tr -d '\\n' >
provider-cert.pem.
line
cat provider-aes.key
| sed 's/$/\\\\
n/'|tr -d '\\n' >
provider-aes.key.l
ine
</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` ## The following block is for daps certifica te and key openssl x509 -in daps-provider.cert - outform PEM | sed 's/ $/\\\\n/'\|tr -d '\\n' > daps-provider.cert .line cat daps-provider.key \| sed 's/$/\\\\n/'| tr -d '\\n' > daps- provider.key.line ``` 3. Untuk memformat rahasia yang akan ditambahkan ke Vault, jalankan perintah berikut: ``` JSONFORMAT='{"cont ent": "%s"}' #create a single line in JSON format printf "${JSONFO RMAT}\\n" "`cat  provider-private- key.pem.line`" > provider-private-k ey.json printf "${JSONFO RMAT}\\n" "`cat  provider-public- key.pem.line`" > provider-public-ke y.json printf "${JSONFO RMAT}\\n" "`cat  provider-cert.pem. line`" > provider- cert.json ```	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`printf "${JSONFORMAT}\\n" "`cat provider-aes.key.line`" > provider-aes.json`
	
	`printf "${JSONFORMAT}\\n" "`cat daps-provider.key.line`" > daps-provider.key.json`
	`printf "${JSONFORMAT}\\n" "`cat daps-provider.cert.line`" > daps-provider.cert.json`
	
	Rahasianya sekarang dalam format JSON dan siap ditambahkan ke brankas.
	4. Untuk mendapatkan nama pod untuk vault, jalankan perintah berikut:
	`kubectl get pods -n provider|egrep "vault|NAME"`
	
	Nama pod akan mirip dengan`"vault-provider-0"`. Nama ini digunakan saat membuat port forward ke vault. Port forward memberi Anda akses ke brankas untuk menambahkan rahasia.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda harus menjalankan ini dari workstation yang memiliki kredensyal AWS yang dikonfigurasi. 5. Untuk mengakses vault, gunakan `kubectl` untuk mengonfigurasi port forward: `kubectl port-forward <VAULT_POD_NAME> 8200:8200 -n provider` Anda sekarang harus dapat mengakses brankas melalui browser atau CLI Anda. Browser 1. Menggunakan browser, navigasikan ke http://127.0.0.1:8200, yang akan menggunakan port forward yang Anda konfigurasikan. 2. Masuk menggunakan token yang Anda konfigurasikan sebelumnya`provider_edc.yml` . Di mesin rahasia, buat tiga rahasia. Setiap rahasia akan memiliki `Path for this secret` nilai, yang merupakan nama rahasia yang ditunjukkan dalam daftar berikut. Di dalam	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`secret data` bagian tersebut, nama kuncinya adalah `content` dan nilainya akan menjadi satu baris teks dari masing-masing file bernama `.line`. 3. Nama-nama rahasia bersumber dari `secretNames` bagian dalam `provider_edc.yml` file. 4. Buat rahasia berikut: • Rahasia `transfer-proxy-token-signer-private-key` dengan nama file `provider-private-key.pem.line` • Rahasia `transfer-proxy-token-signer-public-key` dengan nama file `provider-cert.pem.line` • Rahasia `transfer-proxy-token-encryption-aes-key` dengan nama file `provider-aes.key.line` • Rahasia `daps-private-key` dengan nama file `daps-provider.key.line`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Rahasia `daps-public-key` dengan nama file `daps-provider.cert.line` CLI Lemari Brankas CLI juga akan menggunakan port forward yang Anda konfigurasikan. 1. Di workstation Anda, instal Vault CLI dengan mengikuti petunjuk dalam dokumentasi Vault. HashiCorp 2. Untuk masuk ke vault dengan menggunakan token yang Anda atur`provider_edc.yml` , jalankan perintah berikut: `vault login -address=http://127.0.0.1:8200` Dengan token yang benar, Anda akan melihat pesan `"Success! You are now authenticated."` 3. Untuk membuat rahasia dengan menggunakan file berformat JSON yang Anda buat sebelumnya, jalankan kode berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`vault kv put -address=http://127.0.0.1:8200 secret/transfer-proxy-token-signer-private-key @provider-private-key.json`
	`vault kv put -address=http://127.0.0.1:8200 secret/transfer-proxy-token-signer-public-key @provider-cert.json`
	`vault kv put -address=http://127.0.0.1:8200 secret/transfer-proxy-token-encryption-aes-key @provider-aes.json`
	
	`vault kv put -address=http://127.0.0.1:8200 secret/daps-private-key @daps-provider.key.json`
	`vault kv put -address=http://127.0.0.1:8200 secret/daps-public-key @daps-provider.cert.json`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan dan gunakan konektor konsumen dan layanannya.	Langkah-langkah untuk mengonfigurasi dan menerapkan konsumen serupa dengan yang Anda selesaikan untuk penyedia:
	
	1. Untuk menyalin `consumer_edc.yaml` dari aws-patterns-edcrepo ke folder `tractusx-edc/charts/tractusx-connecto` r, jalankan perintah berikut:
	
	`cd tractusx-edc`
	
	`wget -q https://raw.githubusercontent.com/Think-iT-Labs/aws-patterns-edc/main/edc_helm_configs/consumer_edc.yaml -P charts/tractusx-connector/`
	
	2. Perbarui variabel berikut dengan nilai aktualnya:
	• `CONSUMER_CLIENT_ID` ID yang dihasilkan oleh DAPS. `CONSUMER_CLIENT_ID` Seharusnya ada `config/clients.yml` di server DAPS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• `DAPS_URL`URL DAPS yang sama yang Anda gunakan untuk penyedia.
	• `VAULT_TOKEN` Token yang akan digunakan untuk otorisasi Vault. Pilih nilai apa pun.
	• `vault.fullnameOverride – vault-consumer`
	• `vault.hashicorp.url – http://vault-provider:8200/`
	
	Nilai sebelumnya mengasumsikan bahwa nama penyebaran dan nama namespace adalah. `consumer`
	3. Untuk menjalankan bagan Helm, gunakan perintah berikut:
	
	`cd charts/tractusx-connector`
	
	`helm upgrade --install consumer ./ -f consumer_edc.yaml -n consumer`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan sertifikat dan kunci ke brankas konsumen.	Dari sudut pandang keamanan, kami merekomendasikan regenerasi sertifikat dan kunci untuk setiap peserta ruang data. Pola ini meregenerasi sertifikat dan kunci untuk konsumen. Langkah-langkahnya sangat mirip dengan yang ada di penyedia. Anda dapat memverifikasi nama rahasia dalam `consumer_edc.yml` file. Nama-nama rahasia di dalam lemari besi adalah nilai kunci di `secretNames:` bagian`consumer_edc.yml file`. Secara default, mereka dikonfigurasi sebagai berikut: `secretNames: transferProxyTokenSignerPrivateKey: transfer-proxy-token-signer-private-key transferProxyTokenSignerPublicKey: transfer-proxy-token-signer-public-key transferProxyTokenEncryptionAesKey: transfer-proxy-token-encryption-aes-key`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>dapsPrivateKey: daps-private-key
	dapsPublicKey: daps-public-key</pre>
	`daps-consumer.key` File `daps-consumer.cert` dan yang Anda salin dari server DAPS seharusnya sudah ada di direktori ini.
	
	1. Jalankan perintah berikut:
	
	<pre># generate a private key
	openssl ecparam -name prime256v1 -genkey -noout -out consumer-private-key.pem
	# generate corresponding public key
	openssl ec -in consumer-private-key.pem -pubout -out consumer-public-key.pem
	# create a self-signed certificate
	openssl req -new -x509 -key consumer-private-key.pem -out consumer-cert.pem -days 360
	# generate aes key
	openssl rand -base64 32 > consumer-aes.key</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Edit file secara manual untuk mengganti jeda baris dengan\n, atau gunakan tiga perintah yang mirip dengan berikut ini:
	
	`cat consumer-private-key.pem | sed 's/$/\\\n/'|tr -d '\\n' > consumer-private-key.pem.line`
	`cat consumer-public-key.pem | sed 's/$/\\\n/'|tr -d '\\n' > consumer-public-key.pem.line`
	`cat consumer-cert.pem | sed 's/$/\\\\n/'|tr -d '\\n' > consumer-cert.pem.line`
	`cat consumer-aes.key | sed 's/$/\\\\n/'|tr -d '\\n' > consumer-aes.key.line`
	`cat daps-consumer.cert | sed 's/$/\\\\n/'|tr -d '\\n' > daps-consumer.cert.line`
	`cat daps-consumer.key | sed 's/$/\\\\n/'| tr -d '\\n' > daps-consumer.key.line`
	
	3. Untuk memformat rahasia yang akan ditambahkan ke Vault, jalankan perintah berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`JSONFORMAT='{"content": "%s"}'`
	
	`#create a single line in JSON format`
	
	`printf "${JSONFORMAT}\\n" "`cat consumer-private-key.pem.line`" > consumer-private-key.json`
	`printf "${JSONFORMAT}\\n" "`cat consumer-public-key.pem.line`" > consumer-public-key.json`
	`printf "${JSONFORMAT}\\n" "`cat consumer-cert.pem.line`" > consumer-cert.json`
	`printf "${JSONFORMAT}\\n" "`cat consumer-aes.key.line`" > consumer-aes.json`
	
	`printf "${JSONFORMAT}\\n" "`cat daps-consumer.key.line`" > daps-consumer.key.json`
	`printf "${JSONFORMAT}\\n" "`cat daps-consumer.cert.line`" > daps-consumer.cert.json`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Rahasianya sekarang dalam format JSON dan siap ditambahkan ke brankas.
	4. Untuk mendapatkan nama pod untuk consumer vault, jalankan perintah berikut:
	`kubectl get pods -n consumer | egrep "vault|NAME"`
	Nama pod akan mirip dengan`"vault-consumer-0"` . Nama ini digunakan saat membuat port forward ke vault. Port forward memberi Anda akses ke brankas untuk menambahkan rahasia. Anda harus menjalankan ini dari workstation yang memiliki AWS kredensyal yang dikonfigurasi.
	5. Untuk mengakses vault, gunakan `kubectl` untuk mengonfigurasi port forward:
	`kubectl port-forward <VAULT_POD_NAME> 8201:8200 -n consumer`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Port lokal adalah 8201 kali ini sehingga Anda dapat memiliki port forward untuk produsen dan konsumen. Browser Anda dapat menggunakan browser Anda untuk terhubung ke http://localhost:8201/ untuk mengakses brankas konsumen dan membuat rahasia dengan nama dan konten seperti yang diuraikan. Rahasia dan file yang berisi konten adalah sebagai berikut: • Rahasia `transfer-proxy-token-signer-private-key` dengan nama file `consumer-private-key.pem.line` • Rahasia `transfer-proxy-token-signer-public-key` dengan nama file `consumer-cert.pem.line` • Rahasia `transfer-proxy-token-encryption-aes-key` dengan nama file `consumer-aes.key.line`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CLI Lemari Brankas Menggunakan Vault CLI, Anda dapat menjalankan perintah berikut untuk masuk ke lemari besi dan membuat rahasia: 1. Masuk ke brankas dengan menggunakan token yang Anda konfigurasikan di dalam`consumer_edc.yml` : ``` vault login -address=http://127.0.0.1:8201 ``` Dengan token yang benar, Anda akan melihat pesan `"Success! You are now authenticated."` 2. Untuk membuat rahasia menggunakan file berformat JSON yang Anda buat sebelumnya, jalankan kode berikut: ``` vault kv put -address=http://127.0.0.1:8201 secret/transfer-proxy-token-signer-private-key @consumer-private-key.json vault kv put -address=http://127.0.0.1:8201 secret/transfer-proxy-token	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`-signer-public-key @consumer-cert.json`
	`vault kv put -address= http://127.0.0.1:8 201 secret/transfer- proxy-token-encrypti on-aes-key @consumer -aes.json`
	`vault kv put -address= http://127.0.0.1:8 201 secret/daps- private-key @daps-con sumer.key.json`
	`vault kv put - address=http://12 7.0.0.1:8201 secret/ daps-public-key @daps-consumer.cer t.json`	

Siapkan klien HTTP untuk berinteraksi dengan API manajemen konektor

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mengatur penerusan port.	1. Untuk memeriksa status pod, jalankan perintah berikut:
	`kubectl get pods -n provider`
	`kubectl get pods -n consumer`
	2. Untuk memastikan bahwa penerapan Kubernete s berhasil, lihat log pod penyedia dan konsumen Kubernetes dengan
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	menjalankan perintah berikut:
	`kubectl logs -n provider <producer control plane pod name>`
	`kubectl logs -n consumer <consumer control plane pod name>`
	Cluster bersifat pribadi dan tidak dapat diakses publik. Untuk berinteraksi dengan konektor, gunakan fitur port-forwarding Kubernetes untuk meneruskan lalu lintas yang dihasilkan oleh mesin Anda ke bidang kontrol konektor.
	1. Pada terminal pertama, teruskan permintaan konsumen ke API manajemen melalui port 8300:
	`kubectl port-forward deployment/consumer-tractusx-connector-controlplane 8300:8081 -n consumer`
	2. Pada terminal kedua, teruskan permintaan penyedia ke API	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	manajemen melalui port 8400: `kubectl port-forward deployment/provider-tractusx-connector-controlplane 8400:8081 -n provider`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 untuk penyedia dan konsumen.	Konektor EDC saat ini tidak menggunakan kredensil AWS sementara, seperti yang disediakan dengan mengambil peran. EDC hanya mendukung penggunaa n ID kunci akses IAM dan kombinasi tombol akses rahasia. Dua ember S3 diperlukan untuk langkah selanjutnya. Satu bucket S3 digunakan untuk menyimpan data yang disediakan oleh penyedia. Bucket S3 lainnya adalah untuk data yang diterima oleh konsumen. Pengguna IAM harus memiliki izin untuk membaca dan menulis objek hanya di dua ember bernama. Access Key ID dan Secret Access Key Pair harus dibuat dan disimpan dengan aman. Setelah MVDS ini dinonakti fkan, pengguna IAM harus dihapus. Kode berikut adalah contoh kebijakan IAM untuk pengguna: {	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`"Version": "2012-10-17", "Statement": [{ "Sid": "Stmt1708699805237", "Action": ["s3:GetObject", "s3:GetObjectVersion", "s3:ListAllMyBuckets", "s3:ListBucket", "s3:ListBucketMultipartUploads", "s3:ListBucketVersions", "s3:PutObject"], "Effect": "Allow", "Resource": ["arn:aws:s3:::<S3 Provider Bucket>", "arn:aws:s3:::<S3 Consumer Bucket>", "arn:aws:s3:::<S3 Provider Bucket>/*", "arn:aws:s3:::<S3 Consumer Bucket>/*"] }] }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Postman untuk berinteraksi dengan konektor.	Anda sekarang dapat berinteraksi dengan konektor melalui EC2 instans Anda. Gunakan Postman sebagai klien HTTP, dan berikan Koleksi Tukang Pos untuk penyedia dan konektor konsumen. Impor koleksi dari `aws-pattern-edc` repositori ke instance Postman Anda. Pola ini menggunakan variabel koleksi Postman untuk memberikan masukan ke permintaan Anda.	Pengembang aplikasi, Insinyur data

Menyediakan data jejak emisi karbon X perusahaan melalui konektor

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan data intensitas emisi karbon untuk dibagikan.	Pertama, Anda perlu memutuskan aset data yang akan dibagikan. Data perusahaan X mewakili jejak emisi karbon dari armada kendaraannya. Berat adalah Berat Kendaraan Kotor (GVW) dalam ton, dan emisi dalam gram CO2 per ton-kilom eter (g CO2 e/t-km) menurut pengukuran (WTW): Wheel-to-Well	Insinyur data, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Jenis kendaraan: Van; berat: < 3.5; emisi: 800 • Jenis kendaraan: Truk perkotaan; berat: 3,5-7,5; emisi: 315 • Jenis kendaraan: Kendaraan barang menengah (MGV); berat: 7,5-20; emisi: 195 • Jenis kendaraan: Kendaraan barang berat (HGV); berat: > 20; emisi: 115 Contoh data ada di `carbon_emissions_data.json` file di `aws-patterns-edc` repositori. Perusahaan X menggunakan Amazon S3 untuk menyimpan objek. Buat bucket S3 dan simpan objek data contoh di sana. Perintah berikut membuat bucket S3 dengan pengaturan keamanan default. Kami sangat merekomendasikan konsultasi praktik terbaik Keamanan untuk Amazon S3. `aws s3api create-bucket <BUCKET_NAME> --region <AWS_REGION>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`# You need to add '--create-bucket-configuration # LocationConstraint=<AWS_REGION>' if you want to create # the bucket outside of us-east-1 region`
	
	`aws s3api put-object --bucket <BUCKET_NAME> \ --key <S3 OBJECT NAME> \ --body <PATH OF THE FILE TO UPLOAD>`
	
	Nama bucket S3 harus unik secara global. Untuk informasi selengkapnya tentang aturan penamaan, lihat dokumentasi AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Daftarkan aset data ke konektor penyedia dengan menggunakan Postman.	Aset data konektor EDC memegang nama data dan lokasinya. Dalam hal ini, aset data konektor EDC akan menunjuk ke objek yang dibuat di bucket S3: • Konektor: Penyedia • Permintaan: Buat Aset • Variabel Koleksi: PerbaruiASSET_NAME . Pilih nama yang berarti yang mewakili aset. • Request Body: Perbarui isi permintaan dengan bucket S3 yang Anda buat untuk penyedia. `"dataAddress": { "edc:type": "AmazonS3", "name": "Vehicle Carbon Footprint", "bucketName": "<REPLACE WITH THE SOURCE BUCKET NAME>", "keyName": "<REPLACE WITH YOUR OBJECT NAME>", "region": "<REPLACE WITH THE BUCKET REGION>", "accessKeyId": "<REPLACE WITH YOUR ACCESS KEY ID>", "secretAccessKey": "<REPLACE`	Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>WITH SECRET ACCESS KEY>"
	}</pre>
	• Respons: Permintaan yang berhasil mengembalikan waktu yang dibuat dan ID aset dari aset yang baru dibuat.
	<pre>{
	"@id": "c89aa31c-ec4c-44ed-9e8c-1647f19d7583"
	}</pre>
	• Variabel koleksi ASSET_ID: Perbarui variabel koleksi Postman `ASSET_ID` dengan ID yang dihasilkan secara otomatis oleh konektor EDC setelah pembuatan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tentukan kebijakan penggunaan aset.	Aset data EDC harus dikaitkan dengan kebijakan penggunaan yang jelas. Pertama, buat Definisi Kebijakan di konektor penyedia.
	
	Kebijakan perusahaan X adalah untuk memungkinkan peserta ruang data untuk menggunakan data jejak emisi karbon.
	
	• Badan Permintaan:
	  • Konektor: Penyedia
	  • Permintaan: Buat Kebijakan
	  • Variabel Koleksi: Perbarui `Policy Name` variabel dengan nama kebijakan.
	• Respons: Permintaan yang berhasil mengembalikan waktu yang dibuat dan ID kebijakan kebijakan yang baru dibuat. Perbarui variabel koleksi `POLICY_ID` dengan ID kebijakan yang dihasilkan oleh konektor EDC setelah pembuatan.
		Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan Penawaran Kontrak EDC untuk aset dan kebijakan penggunaannya.	Untuk memungkinkan peserta lain meminta akses ke data Anda, tawarkan dalam kontrak yang menentukan kondisi dan izin penggunaan: • Konektor: Penyedia • Permintaan: Buat Definisi Kontrak • Variabel Koleksi: Perbarui `Contract Name` variabel dengan nama untuk penawaran atau definisi kontrak.	Pengembang aplikasi, Insinyur data

Temukan aset dan capai kesepakatan pada kontrak yang ditentukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Minta katalog data yang dibagikan oleh perusahaan X.	Sebagai konsumen data di ruang data, perusahaa n Y pertama-tama perlu menemukan data yang dibagikan oleh peserta lain. Dalam pengaturan dasar ini, Anda dapat melakukan ini dengan meminta konektor konsumen untuk meminta katalog aset yang tersedia dari konektor penyedia secara langsung. • Konektor: Konsumen	Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Permintaan: Permintaan Katalog
	• Tanggapan: Semua aset data yang tersedia dari penyedia bersama dengan kebijakan penggunaan terlampirnya. Sebagai konsumen data, cari kontrak yang Anda minati dan perbarui variabel pengumpulan berikut yang sesuai.
	  • `CONTRACT_OFFER_ID` ID penawaran kontrak yang ingin dinegosiasikan konsumen
	  • `ASSET_ID`ID aset yang ingin dinegosiasikan konsumen
	  • `PROVIDER_CLIENT_ID` ID konektor penyedia untuk dinegosiasikan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai negosiasi kontrak untuk data intensitas emisi karbon dari perusahaan X.	Sekarang setelah Anda mengidentifikasi aset yang ingin Anda konsumsi, mulailah proses negosiasi kontrak antara konektor konsumen dan penyedia. • Konektor: Konsumen • Permintaan: Negosiasi Kontrak • Variabel Koleksi: Perbarui `CONSUMER_CLIENT_ID` variabel dengan ID konektor konsumen untuk dinegosia sikan. Prosesnya mungkin memakan waktu sebelum mencapai status VERIFIED. Anda dapat memeriksa status Negosiasi Kontrak dan ID Perjanjian yang sesuai dengan menggunakan `Get Negotiation` permintaan.	Pengembang aplikasi, Insinyur data

Konsumsi data dengan menggunakan perjanjian kontrak

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengkonsumsi data dari titik akhir HTTP.	(Opsi 1) Untuk menggunak an bidang data HTTP untuk mengkonsumsi data di ruang data, Anda dapat menggunak	Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an webhook.site untuk meniru server HTTP, dan memulai proses transfer di konektor konsumen: • Konektor: Konsumen • Permintaan: Negosiasi Kontrak • Variabel Koleksi: Perbarui `Contract Agreement ID` variabel dengan ID perjanjian kontrak yang dihasilkan oleh konektor EDC. • Request Body: Perbarui badan permintaan untuk ditentukan HTTP sebagai `dataDestination` di samping URL webhook: `{ "dataDestination": { "type": "HttpProxy" }, "privateProperties": { "receiverHttpEndpoint": "<WEBHOOK URL>" } }` Konektor akan mengirimkan informasi yang diperluka	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n untuk mengunduh file langsung ke URL webhook. Muatan yang diterima mirip dengan yang berikut: `{ "id": "dcc90391-3819-4b54-b401-1a005a029b78", "endpoint": "http://consumer-tractusx-connector-dataplane.consumer:8081/api/public", "authKey": "Authorization", "authCode": "<AUTH CODE YOU RECEIVE IN THE ENDPOINT>", "properties": { "https://w3id.org/edc/v0.0.1/ns/cid": "vehicle-carbon-footprint-contract:4563abf7-5dc7-4c28-bc3d-97f45e32edac:b073669b-db20-4c83-82df-46b583c4c062" } }` Gunakan kredensyal yang diterima untuk mendapatkan aset S3 yang dibagikan oleh penyedia.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pada langkah terakhir ini, Anda harus mengirim permintaan ke bidang data konsumen (port teruskan dengan benar), sebagaima na dinyatakan dalam payload (`endpoint`).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konsumsi data dari ember S3 secara langsung.	(Opsi 2) Gunakan integrasi Amazon S3 dengan konektor EDC, dan langsung arahkan ke bucket S3 di infrastruktur konsumen sebagai tujuan: • Request Body: Perbarui isi permintaan untuk menentukan bucket S3 sebagai DataDeStination. Ini harus menjadi bucket S3 yang sebelumnya Anda buat untuk menyimpan data yang diterima oleh konsumen. `{ "dataDestination": { "type": "AmazonS3", "bucketName": "{{ REPLACE WITH THE DESTINATION BUCKET NAME }}", "keyName": "{{ REPLACE WITH YOUR OBJECT NAME }}", "region": "{{ REPLACE WITH THE BUCKET REGION }}", "accessKeyId": "{{ REPLACE WITH YOUR ACCESS KEY ID }}", "secretAccessKey": "{{ REPLACE WITH SECRET ACCESS KEY }}" }`	Insinyur data, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre> } }</pre>	

Pemecahan masalah

Isu	Solusi
Konektor mungkin menimbulkan masalah tentang format PEM sertifikat.	Gabungkan isi setiap file ke satu baris dengan menambahkan. \n

Sumber daya terkait

- DSSC
- Membangun ruang data untuk kasus penggunaan keberlanjutan (strategi AWS Prescriptive Guidance by Think-IT)
- AWS untuk ruang data
- Dokumentasi Tractus-X
- DAPS
- Mengaktifkan berbagi data melalui ruang data dan AWS (posting blog)

Informasi tambahan

Spesifikasi ruang data

Peserta

Peserta	Deskripsi perusahaan	Fokus perusahaan
Perusahaan X	Mengoperasikan armada kendaraan di seluruh Eropa dan Amerika Selatan untuk mengangkut berbagai barang.	Bertujuan untuk membuat keputusan berbasis data untuk mengurangi intensitas jejak emisi karbonnya.

Perusahaan Y	Otoritas pengatur lingkungan	Menegakkan peraturan dan kebijakan lingkungan yang dirancang untuk memantau dan mengurangi dampak lingkungan dari bisnis dan industri, termasuk intensitas emisi karbon.

Kasus bisnis

Perusahaan X menggunakan teknologi ruang data untuk berbagi data jejak karbon dengan auditor kepatuhan, perusahaan Y, untuk mengevaluasi dan mengatasi dampak lingkungan dari operasi logistik perusahaan X.

Otoritas ruang data

Otoritas ruang data adalah konsorsium organisasi yang mengatur ruang data. Dalam pola ini, baik perusahaan X dan perusahaan Y membentuk badan tata kelola dan mewakili otoritas ruang data federasi.

Komponen ruang data

Komponen	Implementasi yang dipilih	Informasi tambahan
Protokol pertukaran dataset	Protokol Dataspace versi 0.8	• JSON-LD • Kosakata Katalog Data (DCAT)
Konektor ruang data	Konektor EDC Traktus-X versi 0.4.1	• Ekstensi EDC
Kebijakan pertukaran data	Kebijakan PENGGUNAAN Default	• Bahasa Hak Digital Terbuka (ODRL)

Layanan ruang data

Layanan	Implementasi	Informasi tambahan

Layanan identitas	Sistem Penyediaan Atribut Dinamis (DAPS)	Sistem Penyediaan Atribut Dinamis (DAPS) memiliki maksud untuk memastikan atribut tertentu ke organisas i dan konektor. Oleh karena itu, pihak ketiga tidak perlu mempercayai yang terakhir asalkan mereka mempercayai pernyataan DAPS.” — DAPS Untuk fokus pada logika konektor, ruang data diterapka n pada EC2 mesin Amazon menggunakan Docker Compose.
Layanan Penemuan	Katalog Federasi Gaia-X	“Katalog Federasi merupakan repositori terindeks dari Deskripsi Diri Gaia-X untuk memungkinkan penemuan dan pemilihan Penyedia dan penawaran layanan mereka. Deskripsi Diri adalah informasi yang diberikan oleh Peserta tentang diri mereka sendiri dan tentang layanan mereka dalam bentuk properti dan klaim. — Kickstarter Ekosistem Gaia-X

Data yang akan dipertukarkan

Aset data	Deskripsi	Format
Data emisi karbon	Nilai intensitas untuk berbagai jenis kendaraan di wilayah yang ditentukan (Eropa dan	File JSON

	Amerika Selatan) dari seluruh armada kendaraan

Model data

```
{
  "region": "string",
  "vehicles": [
// Each vehicle type has its Gross Vehicle Weight (GVW) category and its emission
 intensity in grams of CO2 per Tonne-Kilometer (g CO2 e/t-km) according to the "Well-
to-Wheel" (WTW) measurement.
    {
      "type": "string",
      "gross_vehicle_weight": "string",
      "emission_intensity": {
        "CO2": "number",
        "unit": "string"
      }
    }
  ]
}
```

Konektor EDC Traktus-X

Untuk dokumentasi setiap parameter Tractus-X EDC, lihat file nilai asli.

Tabel berikut mencantumkan semua layanan, bersama dengan port terbuka dan titik akhir yang sesuai untuk referensi.

Nama layanan	Pelabuhan dan jalur
Bidang kontrol	• manajemen: - Pelabuhan: 8081 Jalur: `/management` • kontrol - Port: 8083 Jalur: `/control` • Port protokol: 8084 Jalur: `/api/v1/dsp` • metrik - Port: 9090 Jalur: `/metrics`

	• observabilitas - Pelabuhan: 8085 Jalur: `/observability`
Bidang data	default - Port: 8080 Jalur: `/api` publik - Pelabuhan: 8081 Jalur: `/api/dataplane/control` proxy - Pelabuhan: 8186 Jalur: `/proxy` metrik - Port: 9090 Jalur: `/metrics` observabilitas - Pelabuhan: 8085 Jalur: `/observability`
Vault	Pelabuhan: 8200
PostgreSQL	Pelabuhan: 5432

Menggunakan AWS Secrets Manager Manajer

Dimungkinkan untuk menggunakan Secrets Manager alih-alih HashiCorp Vault sebagai manajer rahasia. Untuk melakukannya, Anda harus menggunakan atau membangun ekstensi AWS Secrets Manager EDC.

Anda akan bertanggung jawab untuk membuat dan memelihara citra Anda sendiri, karena Tractus-X tidak memberikan dukungan untuk Secrets Manager.

Untuk mencapai itu, Anda perlu memodifikasi file Gradle build dari bidang kontrol dan bidang data konektor dengan memperkenalkan ekstensi AWS Secrets Manager EDC Anda (lihat artefak maven ini sebagai contoh), lalu membangun, memelihara, dan mereferensikan image Docker.

Untuk wawasan lebih lanjut tentang refactoring gambar Docker konektor Tractus-X, lihat Bagan Helm EDC Refactor Tractus-X.

Untuk tujuan kesederhanaan, kami menghindari untuk membangun kembali gambar konektor dalam pola ini dan menggunakan HashiCorp Vault.

Siapkan penyortiran khusus bahasa untuk hasil kueri Amazon Redshift menggunakan UDF Python skalar

Ethan Stark, Amazon Web Services

Ringkasan

Pola ini menyediakan langkah-langkah dan kode contoh untuk menggunakan skalar Python UDF (fungsi yang ditentukan pengguna) untuk menyiapkan penyortiran linguistik yang tidak peka huruf besar/kecil untuk hasil kueri Amazon Redshift. Anda perlu menggunakan UDF Python skalar karena Amazon Redshift mengembalikan hasil berdasarkan urutan UTF-8 biner dan tidak mendukung penyortiran khusus bahasa. UDF Python adalah kode pemrosesan non-SQL yang didasarkan pada program Python dan berjalan di gudang data. Anda dapat menjalankan kode UDF Python dengan pernyataan SQL dalam satu kueri. Untuk informasi lebih lanjut, lihat Pengantar Python UDFs di posting Blog Big Data Amazon AWS Redshift.

Data sampel dalam pola ini didasarkan pada alfabet Turki untuk tujuan demonstrasi. UDF Python skalar dalam pola ini dibuat untuk membuat hasil kueri default Amazon Redshift sesuai dengan urutan linguistik karakter dalam bahasa Turki. Untuk informasi selengkapnya, lihat contoh bahasa Turki di bagian Informasi tambahan dari pola ini. Anda dapat memodifikasi UDF Python skalar dalam pola ini untuk bahasa lain.

Prasyarat dan batasan

Prasyarat

- Cluster Amazon Redshift dengan database, skema, dan tabel
- Pengguna Amazon Redshift dengan `CREATE TABLE` dan izin `CREATE FUNCTION`
- Python 2.7 atau yang lebih baru

Batasan

Penyortiran linguistik yang digunakan oleh kueri dalam pola ini tidak peka huruf besar/kecil.

Alat

Layanan AWS

- Amazon Redshift adalah layanan gudang data skala petabyte terkelola di. AWS Cloud Amazon Redshift terintegrasi dengan data lake Anda, yang memungkinkan Anda menggunakan data Anda untuk memperoleh wawasan baru untuk bisnis dan pelanggan Anda.

Alat-alat lainnya

- Python (UDFs) fungsi yang ditentukan pengguna adalah fungsi yang dapat Anda tulis dengan Python dan kemudian memanggil dalam pernyataan SQL.

Epik

Kembangkan kode untuk mengurutkan hasil kueri dalam urutan linguistik

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat tabel untuk data sampel Anda.	Untuk membuat tabel di Amazon Redshift dan menyisipkan data sampel Anda ke dalam tabel, gunakan pernyataan SQL berikut:
	`CREATE TABLE my_table (first_name varchar(30));`
	`INSERT INTO my_table (first_name) VALUES ('ali'), ('Ali'), ('ırmak'), ('IRMAK'), ('irem'), ('İREM'), ('oğuz'), ('OĞUZ'), ('ömer'), ('ÖMER'), ('sedat'), ('SEDAT'),`
		Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`('şule'),` **Note** Nama-nama depan dalam data sampel termasuk karakter khusus dari alfabet Turki. Untuk informasi lebih lanjut tentang pertimbangan bahasa Turki untuk contoh ini, lihat contoh bahasa Turki di bagian Informasi tambahan dari pola ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa penyortiran default dari data sampel.	Untuk melihat pengurutan default data sampel Anda di Amazon Redshift, jalankan kueri berikut:
	`SELECT first_name FROM my_table ORDER BY first_name;`
	Kueri mengembalikan daftar nama depan dari tabel yang Anda buat sebelumnya:
	`first_name`
	`---------------`
	`Ali`
	`IRMAK`
	`OĞUZ`
	`SEDAT`
	`ali`
	`irem`
	`oğuz`
	`sedat`
	`ÖMER`
	`ömer`
	`İREM`
	`ırmak`
	`ŞULE`
	`şule`
	Hasil kueri tidak dalam urutan yang benar karena urutan UTF-8 biner default tidak mengakomodasi urutan linguistik karakter khusus Turki.
		Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat UDF Python skalar.	Untuk membuat UDF Python skalar, gunakan kode SQL berikut: `CREATE OR REPLACE FUNCTION collate_sort (value varchar) RETURNS varchar IMMUTABLE AS $$ def sort_str(val): import string dictionary = { 'I': 'ı', 'ı': 'h~', 'İ': 'i', 'Ş': 's~', 'ş': 's~', 'Ğ': 'g~', 'ğ': 'g~', 'Ü': 'u~', 'ü': 'u~', 'Ö': 'o~', 'ö': 'o~', 'Ç': 'c~', 'ç': 'c~' } for key, value in dictionary.items(): val = val.replace(key, value) return val.lower()`	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`return sort_str( value)`
	
	`$$ LANGUAGE plpythonu;`	

Kueri data sampel.	Untuk melakukan query data sampel dengan menggunakan UDF Python, jalankan query SQL berikut:
	
	`SELECT first_name FROM my_table ORDER BY collate_sort(first _name);`
	
	Kueri sekarang mengembal ikan data sampel dalam urutan linguistik Turki:
	
	`first_name`
	`---------------`
	`ali`
	`Ali`
	`ırmak`
	`IRMAK`
	`irem`
	`İREM`
	`oğuz`
	`OĞUZ`
	`ömer`
	`Ömer`
	`sedat`
	`SEDAT`
	`şule`
	`ŞULE`
		Insinyur data

Sumber daya terkait

- Klausul ORDER BY (dokumentasi Amazon Redshift)
- Membuat UDF Python skalar (dokumentasi Amazon Redshift)

Informasi tambahan

Contoh bahasa Turki

Amazon Redshift mengembalikan hasil kueri berdasarkan urutan pengurutan UTF-8 biner, bukan urutan pengurutan khusus bahasa. Ini berarti bahwa jika Anda menanyakan tabel Amazon Redshift yang berisi karakter Turki, maka hasil kueri tidak diurutkan menurut urutan linguistik bahasa Turki. Bahasa Turki berisi enam karakter khusus (ç, ı, ğ, ö, ş, dan ü) yang tidak muncul dalam alfabet Latin. Karakter khusus ini ditempatkan di akhir set hasil yang diurutkan berdasarkan urutan UTF-8 biner, seperti yang ditunjukkan tabel berikut.

Pemesanan biner UTF-8	Urutan linguistik Turki
a	a
b	b
c	c
d	ç (*)
e	d
f	e
g	f
-h	g
saya	ğ (*)
j	-h
k	ı (*)

l	saya
m	j
n	k
o	l
p	m
r	n
detik	o
t	ö (*)
u	p
v	r
y	detik
z	ş (*)
ç (*)	t
ğ (*)	u
ı (*)	ü (*)
ö (*)	v
ş (*)	y
ü (*)	z

> **Note**
>
> Tanda bintang (*) menunjukkan karakter khusus dalam bahasa Turki.

Seperti yang diilustrasikan tabel di atas, karakter khusus ç adalah antara c dan d dalam urutan linguistik Turki, tetapi muncul setelah z dalam urutan UTF-8 biner. UDF Python skalar dalam pola ini menggunakan kamus pengganti karakter berikut untuk menggantikan karakter khusus Turki dengan karakter setara Latin yang sesuai.

Karakter khusus Turki	Karakter setara Latin
ç	c ~
ı	h ~
ğ	g ~
ö	o ~
ş	s ~
ü	u ~

> **Note**
>
> Karakter tilde (~) ditambahkan ke akhir karakter Latin yang menggantikan karakter khusus Turki yang sesuai.

Memodifikasi fungsi UDF Python skalar

Untuk memodifikasi fungsi UDF Python skalar dari pola ini sehingga fungsi menerima parameter locate dan mendukung kamus transaksi ganda, gunakan kode SQL berikut:

```
CREATE OR REPLACE FUNCTION collate_sort (value varchar, locale varchar)
RETURNS varchar
IMMUTABLE
AS
$$
    def sort_str(val):
       import string
       # Turkish Dictionary
       if locale == 'tr-TR':
            dictionary = {
```

```
                'I': 'ı',
                'ı': 'h~',
                'İ': 'i',
                'Ş': 's~',
                'ş': 's~',
                'Ğ': 'g~',
                'ğ': 'g~',
                'Ü': 'u~',
                'ü': 'u~',
                'Ö': 'o~',
                'ö': 'o~',
                'Ç': 'c~',
                'ç': 'c~'
        }
        # German Dictionary
        if locale == 'de-DE':
            dictionary = {
                ....
                ....
        }

        for key, value in dictionary.items():
            val = val.replace(key, value)

        return val.lower()

    return sort_str(value)

$$ LANGUAGE plpythonu;
```

Kode contoh berikut menunjukkan bagaimana untuk query UDF Python dimodifikasi:

```
SELECT first_name FROM my_table ORDER BY collate_order(first_name, 'tr-TR');
```

Berlangganan fungsi Lambda ke notifikasi acara dari bucket S3 di Wilayah AWS yang berbeda

Suresh Konathala, Amazon Web Services

Andrew Preston dan Arindom Sarkar, Tidak Ada

Ringkasan

Amazon Simple Storage Service (Amazon S3) Pemberitahuan Acara menerbitkan notifikasi untuk peristiwa tertentu di bucket S3 Anda (misalnya, peristiwa yang dibuat objek, peristiwa penghapusan objek, atau memulihkan peristiwa objek). Anda dapat menggunakan fungsi AWS Lambda untuk memproses notifikasi ini sesuai dengan persyaratan aplikasi Anda. Namun, fungsi Lambda tidak dapat langsung berlangganan notifikasi dari bucket S3 yang di-host di Wilayah AWS yang berbeda.

Pendekatan pola ini menerapkan skenario fanout untuk memproses notifikasi Amazon S3 dari bucket S3 lintas wilayah dengan menggunakan topik Amazon Simple Notification Service (Amazon SNS) untuk setiap Wilayah. Topik SNS Regional ini mengirimkan notifikasi peristiwa Amazon S3 ke antrian Amazon Simple Queue Service (Amazon SQS) di Wilayah tengah yang juga berisi fungsi Lambda Anda. Fungsi Lambda berlangganan antrean SQS ini dan memproses pemberitahuan acara sesuai dengan kebutuhan organisasi Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Bucket S3 yang ada di beberapa Wilayah, termasuk Wilayah pusat untuk meng-host antrean Amazon SQS dan fungsi Lambda.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang ini, lihat Menginstal, memperbarui, dan menghapus instalan AWS CLI dalam dokumentasi AWS CLI.
- Keakraban dengan skenario fanout di Amazon SNS. Untuk informasi selengkapnya tentang ini, lihat Skenario Amazon SNS umum di dokumentasi Amazon SNS.

Arsitektur

Diagram berikut menunjukkan arsitektur untuk pendekatan pola ini.

Diagram menunjukkan alur kerja berikut:

1. Amazon S3 mengirimkan pemberitahuan peristiwa tentang bucket S3 (misalnya, objek yang dibuat, objek dihapus, atau objek dipulihkan) ke topik SNS di Wilayah yang sama.
2. Topik SNS menerbitkan acara ke antrian SQS di Wilayah tengah.
3. Antrian SQS dikonfigurasi sebagai sumber peristiwa untuk fungsi Lambda Anda dan buffer pesan acara untuk fungsi Lambda.
4. Fungsi Lambda melakukan polling antrean SQS untuk pesan dan memproses notifikasi peristiwa Amazon S3 sesuai dengan kebutuhan aplikasi Anda.

Tumpukan teknologi

- Lambda
- Amazon SNS
- Amazon SQS
- Amazon S3

Alat

- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat sumber terbuka untuk berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda. Dengan konfigurasi minimal, Anda dapat menjalankan perintah AWS CLI yang mengimplementasikan fungsionalitas yang setara dengan yang disediakan oleh AWS Management Console berbasis browser dari prompt perintah.
- AWS CloudFormation — AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Anda dapat mengelola dan menyediakan tumpukan di beberapa akun AWS dan Wilayah AWS.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per

detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.

- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) mengoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien, termasuk server web dan alamat email. Pelanggan menerima semua pesan yang dipublikasikan ke topik tempat mereka berlangganan, dan semua pelanggan ke suatu topik menerima pesan yang sama.
- Amazon SQS — Amazon Simple Queue Service (Amazon SQS) menawarkan antrian host yang aman, tahan lama, dan tersedia yang memungkinkan Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi. Amazon SQS mendukung antrian standar dan FIFO.

Epik

Buat antrian SQS dan fungsi Lambda di Wilayah pusat Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat antrian SQS dengan pemicu Lambda.	Masuk ke AWS Management Console dan gunakan petunjuk dari tutorial Menggunakan Lambda dengan Amazon SQS di dokumentasi AWS Lambda untuk membuat sumber daya berikut di Wilayah pusat Anda: • Peran eksekusi Lambda • Fungsi Lambda untuk memproses peristiwa Amazon S3 • Antrian SQS **Note** Pastikan Anda mengonfigurasi antrian	AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	SQS sebagai sumber peristiwa untuk fungsi Lambda Anda.	

Buat topik SNS dan atur notifikasi acara untuk bucket S3 di setiap Wilayah yang diperlukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat topik SNS untuk menerima pemberitahuan acara Amazon S3.	Buat topik SNS di Wilayah tempat Anda ingin menerima pemberitahuan acara Amazon S3. Untuk informasi selengkapnya tentang ini, lihat Membuat topik SNS di dokumentasi Amazon SNS. **Important** Pastikan Anda merekam Nama Sumber Daya Amazon (ARN) topik SNS Anda.	AWS DevOps, arsitek Cloud
Berlangganan topik SNS ke antrian SQS pusat.	Berlangganan topik SNS Anda ke antrian SQS yang diselenggarakan oleh Wilayah pusat Anda. Untuk informasi selengkapnya tentang ini, lihat Berlangganan topik SNS di dokumentasi Amazon SNS.	AWS DevOps, arsitek Cloud
Perbarui kebijakan akses topik SNS.	1. Buka konsol Amazon SNS, pilih Topik, lalu pilih	AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	topik SNS yang Anda buat sebelumnya. 2. Pilih Edit dan kemudian perluas bagian Kebijakan akses - opsional. 3. Lampirkan kebijakan akses berikut ke topik SNS Anda untuk mengizinkan **`sns:publish`** izin Amazon S3, lalu pilih Simpan: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "0", "Effect": "Allow", "Principal": { "Service": "s3.amazonaws.com" }, "Action": "sns:Publish", "Resource": "arn:aws:sns:us-west-2::s3Events-SNSTopic-us-west-2" } ] }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan notifikasi untuk setiap bucket S3 di Wilayah.	Siapkan notifikasi acara untuk setiap bucket S3 di Wilayah. Untuk informasi selengkapnya tentang hal ini, lihat Mengaktifkan dan mengonfigurasi pemberitahuan peristiwa menggunakan konsol Amazon S3 dalam dokumentasi Amazon S3. **Note** Di bagian Tujuan, pilih topik SNS dan tentukan ARN dari topik SNS yang Anda buat sebelumnya.	AWS DevOps, arsitek Cloud
Ulangi epik ini untuk semua Wilayah yang diperlukan.	**Important** Ulangi tugas dalam epik ini untuk setiap Wilayah tempat Anda ingin menerima pemberitahuan acara Amazon S3, termasuk Wilayah pusat Anda.	AWS DevOps, arsitek Cloud

Sumber daya terkait

- Mengkonfigurasi kebijakan akses (dokumentasi Amazon SQS)
- Mengonfigurasi antrian SQS sebagai sumber peristiwa (dokumentasi AWS Lambda)
- Mengkonfigurasi antrian SQS untuk memulai fungsi Lambda (dokumentasi Amazon SQS)

- AWS::Lambda::Function sumber daya (CloudFormation dokumentasi AWS)

Tiga jenis pekerjaan AWS Glue ETL untuk mengonversi data ke Apache Parquet

Adnan Alvee, Nith Govindasivan, dan Karthikeyan Ramachandran, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, AWS Glue adalah layanan ekstrak, transformasi, dan pemuatan (ETL) yang dikelola sepenuhnya. AWS Glue membuatnya hemat biaya untuk mengkategorikan data Anda, membersihkannya, memperkayanya, dan memindahkannya dengan andal di antara berbagai penyimpanan data dan aliran data.

Pola ini menyediakan jenis pekerjaan yang berbeda di AWS Glue dan menggunakan tiga skrip berbeda untuk mendemonstrasikan pembuatan pekerjaan ETL.

Anda dapat menggunakan AWS Glue untuk menulis pekerjaan ETL di lingkungan shell Python. Anda juga dapat membuat pekerjaan ETL batch dan streaming dengan menggunakan Python PySpark () atau Scala di lingkungan Apache Spark yang dikelola. Untuk membantu Anda memulai penulisan pekerjaan ETL, pola ini berfokus pada pekerjaan ETL batch menggunakan Python shell,, dan Scala. PySpark Pekerjaan shell Python dimaksudkan untuk beban kerja yang membutuhkan daya komputasi yang lebih rendah. Lingkungan Apache Spark yang dikelola dimaksudkan untuk beban kerja yang membutuhkan daya komputasi tinggi.

Apache Parquet dibangun untuk mendukung skema kompresi dan pengkodean yang efisien. Ini dapat mempercepat beban kerja analitik Anda karena menyimpan data secara kolumnar. Mengonversi data ke Parket dapat menghemat ruang penyimpanan, biaya, dan waktu Anda dalam jangka panjang. Untuk mempelajari lebih lanjut tentang Parket, lihat posting blog Apache Parquet: Cara menjadi pahlawan dengan format data kolumnar sumber terbuka.

Prasyarat dan batasan

Prasyarat

- Peran AWS Identity and Access Management (IAM) (Jika Anda tidak memiliki peran, lihat bagian Informasi tambahan.)

Arsitektur

Tumpukan teknologi target

- AWS Glue
- Amazon Simple Storage Service (Amazon S3)
- Apache Parquet

Otomatisasi dan skala

- Alur kerja AWS Glue mendukung otomatisasi penuh dari pipeline ETL.
- Anda dapat mengubah jumlah unit pemrosesan data (DPUs), atau tipe pekerja, untuk menskalakan secara horizontal dan vertikal.

Alat

Layanan AWS

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Glue adalah layanan ETL yang dikelola sepenuhnya untuk mengkategorikan, membersihkan, memperkaya, dan memindahkan data Anda antara berbagai penyimpanan data dan aliran data.

Alat-alat lainnya

- Apache Parquet adalah format file data berorientasi kolom sumber terbuka yang dirancang untuk penyimpanan dan pengambilan.

Konfigurasi

Gunakan pengaturan berikut untuk mengonfigurasi daya komputasi AWS Glue ETL. Untuk mengurangi biaya, gunakan pengaturan minimal saat Anda menjalankan beban kerja yang disediakan dalam pola ini.

- Python shell — Anda dapat menggunakan 1 DPU untuk memanfaatkan 16 GB memori atau 0.0625 DPU untuk memanfaatkan 1 GB memori. Pola ini menggunakan 0,0625 DPU, yang merupakan default di konsol AWS Glue.
- Python atau Scala untuk Spark - Jika Anda memilih jenis pekerjaan terkait Spark di konsol, AWS Glue secara default menggunakan 10 pekerja dan tipe pekerja G.1X. Pola ini menggunakan dua

pekerja, yang merupakan jumlah minimum yang diperbolehkan, dengan tipe pekerja standar, yang cukup dan hemat biaya.

Tabel berikut menampilkan berbagai jenis pekerja AWS Glue untuk lingkungan Apache Spark. Karena pekerjaan shell Python tidak menggunakan lingkungan Apache Spark untuk menjalankan Python, itu tidak termasuk dalam tabel.

	Standar	G.1X	G.2X
vCPU	4	4	8
Memori	16 GB	16 GB	32 GB
Ruang disk	50 GB	64 GB	128 GB
Pelaksana per pekerja	2	1	1

Kode

Untuk kode yang digunakan dalam pola ini, termasuk peran IAM dan konfigurasi parameter, lihat bagian Informasi tambahan.

Epik

Unggah datanya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah data ke bucket S3 baru atau yang sudah ada.	Buat atau gunakan bucket S3 yang ada di akun Anda. Unggah file sample_data.csv dari bagian Lampiran, dan perhatikan bucket S3 dan lokasi awalan.	AWS Umum

Membuat dan menjalankan pekerjaan AWS Glue

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pekerjaan AWS Glue.	Di bawah bagian ETL konsol AWS Glue, tambahkan pekerjaan AWS Glue. Pilih jenis pekerjaan yang sesuai, versi AWS Glue, dan DPU/ Worker jenis serta jumlah pekerja yang sesuai. Untuk detailnya, lihat bagian Konfigurasi.	Pengembang, cloud atau data
Ubah lokasi input dan output.	Salin kode yang sesuai dengan tugas AWS Glue Anda, dan ubah lokasi input dan output yang Anda catat dalam epik Unggah data.	Pengembang, cloud atau data
Konfigurasikan parameter.	Anda dapat menggunakan cuplikan yang disediakan di bagian Informasi tambahan untuk mengatur parameter untuk pekerjaan ETL Anda. AWS Glue menggunakan empat nama argumen secara internal: • `--conf` • `--debug` • `--mode` • `--JOB_NAME` `--JOB_NAME` Parameter harus dimasukkan secara eksplisit pada konsol AWS	Pengembang, cloud atau data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Glue. Pilih Pekerjaan , Edit Job, Konfigurasi keamanan, pustaka skrip, dan parameter pekerjaan (opsional). Masukkan `--JOB_NAME` sebagai kunci dan berikan nilai. Anda juga dapat menggunakan AWS Command Line Interface (AWS CLI) atau AWS Glue API untuk mengatur parameter ini. `--JOB_NAME` Parameter ini digunakan oleh Spark dan tidak diperlukan dalam pekerjaan lingkungan shell Python. Anda harus menambahk an `--` sebelum setiap nama parameter; jika tidak, kode tidak akan berfungsi . Misalnya, untuk cuplikan kode, parameter lokasi harus dipanggil oleh dan. `--input_loc --output_loc`	
Jalankan pekerjaan ETL.	Jalankan pekerjaan Anda dan periksa hasilnya. Perhatika n berapa banyak ruang yang dikurangi dari file asli.	Pengembang, cloud atau data

Sumber daya terkait

Referensi

- Apache Spark
- AWS Glue: Cara kerjanya
- Harga AWS Glue

Tutorial dan video

- Apa itu AWS Glue?

Informasi tambahan

Peran IAM

Saat membuat pekerjaan AWS Glue, Anda dapat menggunakan peran IAM yang ada yang memiliki izin yang ditampilkan dalam cuplikan kode berikut atau peran baru.

Untuk membuat peran baru, gunakan kode YAMB berikut.

```
# (c) 2022 Amazon Web Services, Inc. or its affiliates. All Rights Reserved. This AWS
 Content is provided subject to the terms of the AWS Customer
# Agreement available at https://aws.amazon.com/agreement/ or other written agreement
 between Customer and Amazon Web Services, Inc.

AWSTemplateFormatVersion: "2010-09-09"

Description: This template will setup IAM role for AWS Glue service.

Resources:
  rGlueRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "glue.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole
```

```
      Policies:
        - PolicyName: !Sub "${AWS::StackName}-s3-limited-read-write-inline-policy"
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "s3:PutObject"
                  - "s3:GetObject"
                Resource: "arn:aws:s3:::*/*"
      Tags:
        - Key  : "Name"
          Value : !Sub "${AWS::StackName}"

Outputs:
  oGlueRoleName:
    Description: AWS Glue IAM role
    Value:
      Ref: rGlueRole
    Export:
      Name: !Join [ ":", [ !Ref "AWS::StackName", rGlueRole ] ]
```

AWS Glue Python Shell

Kode Python menggunakan Pandas dan PyArrow pustaka untuk mengkonversi data ke Parquet. Perpustakaan Pandas sudah tersedia. PyArrow Pustaka diunduh saat Anda menjalankan pola, karena ini adalah proses satu kali. Anda dapat menggunakan file roda PyArrow untuk mengonversi ke perpustakaan dan menyediakan file sebagai paket perpustakaan. Untuk informasi selengkapnya tentang mengemas file roda, lihat Menyediakan pustaka Python Anda sendiri.

Parameter cangkang AWS Glue Python

```
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["input_loc", "output_loc"])
```

Kode shell AWS Glue Python

```
from io import BytesIO
import pandas as pd
import boto3
import os
```

```
import io
import site
from importlib import reload
from setuptools.command import easy_install
install_path = os.environ['GLUE_INSTALLATION']
easy_install.main( ["--install-dir", install_path, "pyarrow"] )
reload(site)
import pyarrow


input_loc = "s3://bucket-name/prefix/sample_data.csv"
output_loc = "s3://bucket-name/prefix/"


input_bucket = input_loc.split('/', 1)[0]
object_key = input_loc.split('/', 1)[1]


output_loc_bucket = output_loc.split('/', 1)[0]
output_loc_prefix = output_loc.split('/', 1)[1]


s3 = boto3.client('s3')
obj = s3.get_object(Bucket=input_bucket, Key=object_key)
df = pd.read_csv(io.BytesIO(obj['Body'].read()))


parquet_buffer = BytesIO()
s3_resource = boto3.resource('s3')
df.to_parquet(parquet_buffer, index=False)
s3_resource.Object(output_loc_bucket, output_loc_prefix +  'data' +
 '.parquet').put(Body=parquet_buffer.getvalue())
```

Pekerjaan AWS Glue Spark dengan Python

Untuk menggunakan jenis pekerjaan AWS Glue Spark dengan Python, pilih Spark sebagai jenis pekerjaan. Pilih Spark 3.1, Python 3 dengan waktu startup pekerjaan yang ditingkatkan (Glue Version 3.0) sebagai versi AWS Glue.

Parameter AWS Glue Python

```
from awsglue.utils import getResolvedOptions
```

```
args = getResolvedOptions(sys.argv, ["JOB_NAME", "input_loc", "output_loc"])
```

Pekerjaan AWS Glue Spark dengan kode Python

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.transforms import *
from awsglue.dynamicframe import DynamicFrame
from awsglue.utils import getResolvedOptions
from awsglue.job import Job


sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)

input_loc = "s3://bucket-name/prefix/sample_data.csv"
output_loc = "s3://bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(\
    connection_type = "s3", \
    connection_options = {
        "paths": [input_loc]}, \
    format = "csv",
    format_options={
        "withHeader": True,
        "separator": ","
    })


outputDF = glueContext.write_dynamic_frame.from_options(\
    frame = inputDyf, \
    connection_type = "s3", \
    connection_options = {"path": output_loc \
        }, format = "parquet")
```

Untuk sejumlah besar file besar terkompresi (misalnya, 1.000 file yang masing-masing sekitar 3 MB), gunakan `compressionType` parameter dengan `recurse` parameter untuk membaca semua file yang tersedia dalam awalan, seperti yang ditunjukkan pada kode berikut.

```
input_loc = "bucket-name/prefix/"
```

```
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(
                    connection_type = "s3",
                    connection_options = {"paths": [input_loc],
                                          "compressionType":"gzip","recurse" :"True",
                                          },
                    format = "csv",
                    format_options={"withHeader": True,"separator": ","}
                    )
```

Untuk sejumlah besar file kecil terkompresi (misalnya, 1.000 file yang masing-masing sekitar 133 KB), gunakan `groupFiles` parameter, bersama dengan parameter `compressionType` dan parameter. `recurse` `groupFiles`Parameter mengelompokkan file kecil menjadi beberapa file besar, dan `groupSize` parameter mengontrol pengelompokan ke ukuran yang ditentukan dalam byte (misalnya, 1 MB). Cuplikan kode berikut memberikan contoh penggunaan parameter ini dalam kode.

```
input_loc = "bucket-name/prefix/"
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(
                    connection_type = "s3",
                    connection_options = {"paths": [input_loc],
                                          "compressionType":"gzip","recurse" :"True",
                                           "groupFiles" :"inPartition",
 "groupSize" :"1048576",
                                          },
                    format = "csv",
                    format_options={"withHeader": True,"separator": ","}
                    )
```

Tanpa perubahan apa pun pada node pekerja, pengaturan ini memungkinkan pekerjaan AWS Glue membaca banyak file (besar atau kecil, dengan atau tanpa kompresi) dan menuliskannya ke target dalam format Parket.

Pekerjaan AWS Glue Spark dengan Scala

Untuk menggunakan jenis pekerjaan AWS Glue Spark dengan Scala, pilih Spark sebagai tipe pekerjaan dan Bahasa sebagai Scala. Pilih Spark 3.1, Scala 2 dengan waktu startup pekerjaan yang ditingkatkan (Glue Version 3.0) sebagai versi AWS Glue. Untuk menghemat ruang penyimpanan,

sampel AWS Glue with Scala berikut juga menggunakan `applyMapping` fitur untuk mengonversi tipe data.

Parameter AWS Glue Scala

```
import com.amazonaws.services.glue.util.GlueArgParser val args =
 GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME", "inputLoc",
 "outputLoc").toArray)
```

Pekerjaan AWS Glue Spark dengan kode Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.DynamicFrame
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._


object GlueScalaApp {
  def main(sysArgs: Array[String]) {

    @transient val spark: SparkContext = SparkContext.getOrCreate()
    val glueContext: GlueContext = new GlueContext(spark)

    val inputLoc = "s3://bucket-name/prefix/sample_data.csv"
    val outputLoc = "s3://bucket-name/prefix/"

    val readCSV = glueContext.getSource("csv", JsonOptions(Map("paths" ->
 Set(inputLoc)))).getDynamicFrame()

    val applyMapping = readCSV.applyMapping(mappings = Seq(("_c0", "string", "date",
 "string"), ("_c1", "string", "sales", "long"),
    ("_c2", "string", "profit", "double")), caseSensitive = false)

    val formatPartition = applyMapping.toDF().coalesce(1)

    val dynamicFrame = DynamicFrame(formatPartition, glueContext)

    val dataSink = glueContext.getSinkWithFormat(
```

```
        connectionType = "s3",
        options = JsonOptions(Map("path" -> outputLoc )),
        transformationContext = "dataSink", format =
 "parquet").writeDynamicFrame(dynamicFrame)
  }
}
```

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Visualisasikan log audit Amazon Redshift menggunakan Amazon Athena dan Amazon QuickSight

Sanket Sirsikar dan Gopal Krishna Bhatia, Amazon Web Services

Ringkasan

Keamanan merupakan bagian integral dari operasi database di Amazon Web Services (AWS) Cloud. Organisasi Anda harus memastikan bahwa ia memantau aktivitas dan koneksi pengguna basis data untuk mendeteksi potensi insiden dan risiko keamanan. Pola ini membantu Anda memantau database Anda untuk tujuan keamanan dan pemecahan masalah, yang merupakan proses yang dikenal sebagai audit basis data.

Pola ini menyediakan skrip SQL yang mengotomatiskan pembuatan tabel Amazon Athena dan tampilan untuk dasbor pelaporan di Amazon yang membantu Anda QuickSight mengaudit log Amazon Redshift. Ini memastikan bahwa pengguna yang bertanggung jawab untuk memantau aktivitas database memiliki akses mudah ke fitur keamanan data.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Cluster Amazon Redshift yang ada. Untuk informasi selengkapnya tentang ini, lihat Membuat klaster Amazon Redshift di dokumentasi Amazon Redshift.
- Akses ke workgroup Athena yang ada. Untuk informasi selengkapnya, lihat Cara kerja kelompok kerja di dokumentasi Amazon Athena.
- Bucket sumber Amazon Simple Storage Service (Amazon S3) yang sudah ada dengan izin AWS Identity and Access Management (IAM) yang diperlukan. Untuk informasi selengkapnya, lihat Izin Bucket untuk pencatatan audit Amazon Redshift dari pencatatan audit Database di dokumentasi Amazon Redshift.

Arsitektur

Tumpukan teknologi

- Athena
- Amazon Redshift
- Amazon S3
- QuickSight

Alat

- Amazon Athena - Athena adalah layanan kueri interaktif yang memudahkan untuk menganalisis data di Amazon S3 menggunakan SQL standar.
- Amazon QuickSight - QuickSight adalah layanan intelijen bisnis (BI) yang dapat diskalakan, tanpa server, dapat disematkan, dan didukung pembelajaran mesin.
- Amazon Redshift — Amazon Redshift adalah layanan pergudangan data tingkat perusahaan, skala petabyte, dan terkelola sepenuhnya.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet.

Epik

Konfigurasikan cluster Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan pencatatan audit untuk klaster Amazon Redshift.	1. Masuk ke AWS Management Console, buka konsol Amazon Redshift, pilih CLUSTERS, lalu pilih klaster yang ingin Anda aktifkan pencatatan. 2. Pilih tab Properti dan kemudian aktifkan audit dengan mengikuti petunjuk dari Mengonfigurasi audit menggunakan konsol dalam	DBA, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dokumentasi Amazon Redshift.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan logging di grup parameter cluster Amazon Redshift.	Anda dapat mengaktif kan audit log koneksi, log pengguna, dan log aktivitas pengguna secara bersamaan dengan menggunakan AWS Management Console, referensi Amazon Redshift API, atau AWS Command Line Interface (AWS CLI). Untuk mengaudit log aktivitas pengguna, Anda harus mengaktifkan parameter `enable_user_activi ty_logging` database. Jika Anda hanya mengaktif kan fitur audit logging tetapi bukan parameter terkait, audit database mencatat informasi logging untuk koneksi dan log pengguna tetapi tidak untuk log aktivitas pengguna. `enable_user_activi ty_logging` Parameter tidak diaktifkan secara default, tetapi Anda dapat mengaktif kannya dengan mengubahnya dari `false` ke`true`. ⚠ Important Anda perlu membuat grup parameter cluster baru dengan `user_acti`	DBA, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`vity_logging` parameter diaktifkan dan melampirkannya ke cluster Amazon Redshift Anda. Untuk informasi selengkap nya tentang hal ini, lihat Memodifikasi klaster di dokumentasi Amazon Redshift. Untuk informasi selengkap nya tentang tugas ini, lihat grup parameter Amazon Redshift dan Mengonfigurasi audit menggunakan konsol dalam dokumentasi Amazon Redshift.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan izin bucket S3 untuk pencatatan klaster Amazon Redshift.	Saat Anda mengaktifkan logging, Amazon Redshift mengumpulkan informasi pencatatan dan mengungga hnya ke file log yang disimpan dalam bucket S3. Anda dapat menggunakan bucket S3 yang sudah ada atau membuat bucket baru. **⚠ Important** Pastikan Amazon Redshift memiliki izin IAM yang diperluka n untuk mengakses bucket S3. Untuk informasi selengkap nya tentang hal ini, lihat Izin Bucket untuk pencatatan audit Amazon Redshift dari pencatatan audit Database di dokumentasi Amazon Redshift.	DBA, Insinyur data

Buat tabel dan tampilan Athena

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel Athena dan tampilan untuk menanyaka	Buka konsol Amazon Athena dan gunakan kueri bahasa definisi data (DDL) dari skrip	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
n data log audit Amazon Redshift dari bucket S3.	`AuditLogging.sql` SQL (terlampir) untuk membuat tabel dan tampilan untuk log aktivitas pengguna, log pengguna, dan log koneksi. Untuk informasi dan petunjuk selengkapnya, lihat tutorial Buat tabel dan jalankan kueri dari Amazon Athena Workshop.	

Siapkan pemantauan log di QuickSight dasbor

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat QuickSight dasbor menggunakan Athena sebagai sumber data.	Buka QuickSight konsol Amazon dan buat QuickSigh t dasbor dengan mengikuti petunjuk di Visualize with QuickSight using Athena tutorial dari Amazon Athena Workshop.	DBA, Insinyur data

Sumber daya terkait

- Buat tabel dan jalankan kueri di Athena
- Visualisasikan dengan QuickSight menggunakan Athena

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Visualisasikan laporan kredensi IAM untuk semua akun AWS menggunakan Amazon Quick Sight

Parag Nagwekar dan Arun Chandapillai, Amazon Web Services

Ringkasan

> Peringatan: Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Anda dapat menggunakan laporan kredensi AWS Identity and Access Management (IAM) untuk membantu Anda memenuhi persyaratan keamanan, audit, dan kepatuhan organisasi Anda. Laporan kredensyal menyediakan daftar semua pengguna di akun AWS Anda dan menunjukkan status kredensialnya, seperti kata sandi, kunci akses, dan perangkat autentikasi multi-faktor (MFA). Anda dapat menggunakan laporan kredensyal untuk beberapa akun AWS yang dikelola oleh AWS Organizations.

Pola ini mencakup langkah-langkah dan kode untuk membantu Anda membuat dan membagikan laporan kredensi IAM untuk semua akun AWS di organisasi Anda dengan menggunakan dasbor Amazon Quick Sight. Anda dapat berbagi dasbor dengan pemangku kepentingan di organisasi Anda. Laporan dapat membantu organisasi Anda mencapai hasil bisnis yang ditargetkan berikut:

- Identifikasi insiden keamanan yang terkait dengan pengguna IAM
- Lacak migrasi real-time pengguna IAM ke otentikasi sistem masuk tunggal (SSO)
- Lacak Wilayah AWS yang diakses oleh pengguna IAM
- Tetap patuh
- Berbagi informasi dengan pemangku kepentingan lainnya

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif

- Organisasi dengan akun anggota
- Peran IAM dengan izin untuk mengakses akun di Organizations
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi
- Berlangganan Amazon Quick Suite Enterprise edisi

Arsitektur

Tumpukan teknologi

- Amazon Athena
- Amazon EventBridge
- Penglihatan Cepat Amazon
- Amazon Simple Storage Service (Amazon S3)
- AWS Glue
- AWS Identity and Access Management (IAM)
- AWS Lambda
- AWS Organizations

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk menyiapkan alur kerja yang menangkap data laporan kredensyal IAM dari beberapa akun AWS.

1. EventBridge memanggil fungsi Lambda setiap hari.
2. Fungsi Lambda mengasumsikan peran IAM di setiap akun AWS di seluruh organisasi. Kemudian, fungsi membuat laporan kredensyal IAM dan menyimpan data laporan dalam bucket S3 terpusat. Anda harus mengaktifkan enkripsi dan menonaktifkan akses publik pada bucket S3.
3. Crawler AWS Glue merayapi bucket S3 setiap hari dan memperbarui tabel Athena yang sesuai.
4. Quick Sight mengimpor dan menganalisis data dari laporan kredensyal dan membangun dasbor yang dapat divisualisasikan oleh dan dibagikan dengan pemangku kepentingan.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang memudahkan untuk menganalisis data di Amazon S3 dengan menggunakan SQL standar.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data real-time dari berbagai sumber. Misalnya, fungsi Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun AWS lainnya.
- Amazon Quick Sight adalah layanan intelijen bisnis skala cloud (BI) yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda dalam satu dasbor. Quick Sight adalah komponen inti dalam Amazon Quick Suite, menyediakan visualisasi data interaktif, analisis dalam memori SPICE, analitik tertanam, dan berbagi dasbor.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Kode

Kode untuk pola ini tersedia di GitHub getiamcredsreport-allaccounts-orgrepositori. Anda dapat menggunakan kode dari repositori ini untuk membuat laporan kredensi IAM di seluruh akun AWS di Organizations dan menyimpannya di lokasi pusat.

Epik

Menyiapkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan edisi Amazon Quick Suite Enterprise.	1. Aktifkan edisi Amazon Quick Suite Enterpris e di akun AWS Anda. Untuk informasi selengkap	Administrator AWS, AWS DevOps, administrator Cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nya, lihat Mengelola akses pengguna di dalam Amazon Quick Suite dalam dokumentasi Quick Suite. 2. Untuk memberikan izin dasbor, dapatkan Nama Sumber Daya Amazon (ARN) dari pengguna Quick Sight.	
Integrasikan Amazon Quick Sight dengan Amazon S3 dan Athena.	Anda harus mengotorisasi Quick Sight untuk menggunakan Amazon S3 dan Athena sebelum menerapkan tumpukan AWS. CloudFormation	Administrator AWS, AWS DevOps, administrator Cloud, arsitek Cloud

Menyebarkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	1. Kloning GitHub getiamcredsreport-allaccounts-orgrepositori ke mesin lokal Anda dengan menjalankan perintah berikut: `git clone https://github.com/aws-samples/getiamcredsreport-allaccounts-org`	Administrator AWS
Menyebarkan infrastruktur.	1. Masuk ke AWS Management Console	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan buka CloudFormation konsol. 2. Di panel navigasi, pilih Buat tumpukan, lalu pilih Dengan sumber daya baru (standar). 3. Pada halaman Identifikasi sumber daya, pilih Berikutnya. 4. Pada halaman Tentukan templat, untuk Sumber templat, pilih Unggah file templat. 5. Pilih Pilih file, pilih **`Cloudformation-createcredrepo.yaml`** file dari GitHub repositori kloning Anda, lalu pilih Berikutnya. 6. Di Parameter, perbarui `IAMRoleName` dengan peran IAM Anda. Ini harus menjadi peran IAM yang Anda ingin Lambda untuk mengambil alih di setiap akun organisasi. Peran ini membuat laporan kredensi. Catatan: Peran tidak harus ada di semua akun pada langkah pembuatan tumpukan ini. 7. Di Parameter, perbarui `S3BucketName` dengan nama bucket S3 tempat	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Lambda dapat menyimpan kredensyal untuk semua akun. 8. Untuk nama Stack, masukkan nama tumpukan Anda. 9. Pilih Kirim. 10. Perhatikan nama peran fungsi Lambda.	
Buat kebijakan izin IAM.	Buat kebijakan IAM untuk setiap akun AWS di seluruh organisasi Anda dengan izin berikut: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iam:GenerateCredentialReport", "iam:GetCredentialReport" ], "Resource": "*" } ] }`	AWS DevOps, Administrator Cloud, Arsitek cloud, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM dengan kebijakan kepercayaan.	1. Buat peran IAM untuk akun AWS dan lampirkan kebijakan izin yang Anda buat di langkah sebelumnya. 2. Lampirkan kebijakan kepercayaan berikut ke peran IAM: `{ "Version": "2012-10-17", "Statement":[ { "Effect": "Allow", "Principal":{ "AWS":[ "arn:aws:iam::<MasterAccountID>:role/<LambdaRole>" ] }, "Action": "sts:AssumeRole" } ] }` **Important** Ganti `arn:aws:iam::<MasterAccountID>:role/<LambdaRole>`	Administrator cloud, arsitek Cloud, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan ARN dari peran Lambda yang Anda catat sebelumnya. **Note** Organizations biasanya menggunakan otomatisasi untuk membuat peran IAM untuk akun AWS mereka. Kami menyarankan Anda menggunakan otomatisasi ini, jika tersedia. Atau, Anda dapat menggunakan `CreateRoleforOrg.py` skrip dari repositori kode. Skrip memerlukan peran administratif yang ada atau peran IAM lainnya yang memiliki izin untuk membuat kebijakan dan peran IAM di setiap akun AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Amazon Quick Sight untuk memvisualisasikan data.	1. Masuk ke Quick Suite dengan kredensialmu. 2. Buat kumpulan data dengan menggunakan Athena (menggunakan `iamcredreportdb` database `"cfn_iamc redreport"` dan tabel), lalu menyegarkan dataset secara otomatis. 3. Buat analisis di Quick Sight. 4. Buat dasbor Quick Sight.	AWS DevOps, Administrator Cloud, Arsitek cloud, Insinyur data

Informasi tambahan

Pertimbangan tambahan

Pertimbangkan hal berikut:

- Setelah menggunakan CloudFormation infrastruktur, Anda dapat menunggu laporan yang dibuat di Amazon S3 dan dianalisis oleh Athena hingga Lambda dan AWS Glue berjalan sesuai jadwal mereka. Atau, Anda dapat menjalankan Lambda secara manual untuk mendapatkan laporan di Amazon S3, lalu menjalankan crawler AWS Glue untuk mendapatkan tabel Athena yang dibuat dari data.
- Quick Suite adalah alat yang ampuh untuk menganalisis dan memvisualisasikan data berdasarkan kebutuhan bisnis Anda. Anda dapat menggunakan parameter di Quick Suite untuk mengontrol data widget berdasarkan bidang data yang Anda pilih. Selain itu, Anda dapat menggunakan analisis Quick Suite untuk membuat parameter (misalnya, bidang Akun, Tanggal, dan Pengguna sepertipartition_0,partition_1, dan `user` masing-masing) dari kumpulan data Anda untuk menambahkan kontrol parameter untuk Akun, Tanggal, dan Pengguna.
- Untuk membuat dasbor Quick Sight Anda sendiri, lihat Lokakarya Quick Suite dari situs web AWS Workshop Studio.

- Untuk melihat contoh dasbor Quick Sight, lihat repositori GitHub getiamcredsreport-allaccounts-orgkode.

Hasil bisnis yang ditargetkan

Anda dapat menggunakan pola ini untuk mencapai hasil bisnis yang ditargetkan berikut:

- Identifikasi insiden keamanan yang terkait dengan pengguna IAM — Selidiki setiap pengguna di setiap akun AWS di organisasi Anda dengan menggunakan satu panel kaca. Anda dapat melacak tren Wilayah AWS individual yang paling baru diakses pengguna IAM dan layanan yang mereka gunakan.
- Lacak migrasi real-time pengguna IAM ke otentikasi SSO — Dengan menggunakan SSO, pengguna dapat masuk sekali dengan satu kredensi dan mengakses beberapa akun dan aplikasi AWS. Jika Anda berencana untuk memigrasikan pengguna IAM ke SSO, pola ini dapat membantu Anda beralih ke SSO dan melacak semua penggunaan kredensi pengguna IAM (seperti akses ke AWS Management Console atau penggunaan kunci akses) di semua akun AWS.
- Lacak Wilayah AWS yang diakses oleh pengguna IAM — Anda dapat mengontrol akses pengguna IAM ke Wilayah untuk berbagai tujuan, seperti kedaulatan data dan pengendalian biaya. Anda juga dapat melacak penggunaan Wilayah oleh setiap pengguna IAM.
- Tetap patuh - Dengan mengikuti prinsip hak istimewa terkecil, Anda hanya dapat memberikan izin IAM tertentu yang diperlukan untuk melakukan tugas tertentu. Selain itu, Anda dapat melacak akses ke layanan AWS, AWS Management Console, dan penggunaan kredensyal jangka panjang.
- Berbagi informasi dengan pemangku kepentingan lainnya — Anda dapat berbagi dasbor yang dikuratori dengan pemangku kepentingan lainnya, tanpa memberi mereka akses ke laporan kredensi IAM atau akun AWS.

Lebih banyak pola

- Mengautentikasi pengguna aplikasi React yang ada dengan menggunakan Amazon Cognito dan AWS Amplify UI
- Secara otomatis mengekstrak konten dari file PDF menggunakan Amazon Ttract
- Buat model peramalan awal dingin dengan menggunakan DeepAR untuk deret waktu di SageMaker Amazon AI Studio Lab
- Membuat laporan biaya dan penggunaan terperinci untuk klaster EMR Amazon dengan menggunakan AWS Cost Explorer
- Buat laporan biaya dan penggunaan terperinci untuk Amazon RDS dan Amazon Aurora
- Buat laporan biaya dan penggunaan terperinci untuk pekerjaan AWS Glue dengan menggunakan AWS Cost Explorer
- Mengirimkan data DynamoDB ke Amazon S3 menggunakan Kinesis Data Streams dan Firehose dengan AWS CDK
- Menerapkan pekerjaan AWS Glue dengan pipeline AWS CodePipeline CI/CD
- Sematkan komponen visual Amazon Quick Sight ke dalam aplikasi web dengan menggunakan otomatisasi Amazon Cognito dan IAc
- Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB
- Jalankan kueri Amazon Redshift SQL dengan menggunakan Terraform
- Ekstrak dan kueri atribut AWS IoT SiteWise metadata di danau data
- Hasilkan wawasan data dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight
- Hasilkan wawasan z/OS data Db2 dengan menggunakan dan AWS Mainframe Modernization Amazon Q di Quick Sight
- Berikan instans SageMaker notebook akses sementara ke CodeCommit repositori di akun AWS lain
- Identifikasi dan beri tahu kapan sumber daya Amazon Data Firehose tidak dienkripsi dengan kunci AWS KMS
- Impor pustaka psycopg2 AWS Lambda untuk berinteraksi dengan database PostgreSQL Anda
- Menyerap dan menganalisis log keamanan AWS di Microsoft Sentinel
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dalam mode SSL dengan menggunakan AWS DMS

- Migrasikan database Oracle ke Amazon RDS for Oracle dengan menggunakan adaptor file datar Oracle GoldenGate
- Migrasikan Database Oracle ke Amazon Redshift menggunakan AWS DMS dan AWS SCT
- Memigrasikan data dari lingkungan Hadoop lokal ke Amazon S3 menggunakan AWS untuk Amazon S3 DistCp PrivateLink
- Migrasi dari Couchbase Server ke Couchbase Capella di AWS
- Memigrasikan beban kerja Cloudera lokal ke Platform Data Cloudera di AWS
- Kelola failover Multi-AZ untuk cluster EMR dengan menggunakan Application Recovery Controller
- Mengotomatiskan blue/green penyebaran database global Amazon Aurora dengan menggunakan prinsip IAc
- Menyediakan AWS Service Catalog produk berdasarkan AWS CloudFormation template dengan menggunakan GitHub Actions
- Jalankan pengujian unit untuk pekerjaan ETL Python dalam AWS Glue menggunakan kerangka pytest
- Siapkan dasbor pemantauan Grafana untuk AWS ParallelCluster
- Membongkar data dari klaster Amazon Redshift di seluruh akun ke Amazon S3
- Visualisasikan hasil AI/ML model menggunakan Flask dan AWS Elastic Beanstalk

Hitung

Topik

- Kontainer & layanan mikro
- Nirserver
- Jaringan
- Pengiriman konten

Kontainer & layanan mikro

Topik

- Akses database Amazon Neptunus dari wadah Amazon EKS
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS PrivateLink dan Network Load Balancer
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS Fargate, PrivateLink AWS, dan Network Load Balancer
- Akses aplikasi kontainer secara pribadi di Amazon EKS menggunakan AWS PrivateLink dan Network Load Balancer
- Mengotomatiskan pencadangan untuk Amazon RDS untuk instans DB Amazon RDS for PostgreSQL dengan menggunakan AWS Batch
- Otomatiskan penerapan Node Termination Handler di Amazon EKS dengan menggunakan pipeline CI/CD
- Secara otomatis membangun dan menyebarkan aplikasi Java ke Amazon EKS menggunakan pipeline CI/CD
- Salin gambar wadah Amazon ECR di seluruh Akun AWS dan Wilayah AWS
- Buat definisi tugas Amazon ECS dan pasang sistem file pada EC2 instans menggunakan Amazon EFS
- Menyebarkan fungsi Lambda dengan gambar kontainer
- Menerapkan layanan mikro Java di Amazon ECS menggunakan AWS Fargate
- Menerapkan resource dan paket Kubernetes menggunakan Amazon EKS dan repositori bagan Helm di Amazon S3
- Menerapkan cluster CockroachDB di Amazon EKS dengan menggunakan Terraform
- Menerapkan contoh layanan mikro Java di Amazon EKS dan mengekspos layanan mikro menggunakan Application Load Balancer
- Menerapkan aplikasi yang dikelompokkan ke Amazon ECS dengan menggunakan AWS Copilot
- Menerapkan aplikasi berbasis GRPC di cluster Amazon EKS dan mengaksesnya dengan Application Load Balancer
- Menerapkan aplikasi kontainer saat AWS IoT Greengrass V2 berjalan sebagai wadah Docker
- Menyebarkan wadah dengan menggunakan Elastic Beanstalk
- Hasilkan alamat IP keluar statis menggunakan fungsi Lambda, Amazon VPC, dan arsitektur tanpa server

- Identifikasi gambar kontainer duplikat secara otomatis saat bermigrasi ke repositori Amazon ECR
- Instal Agen SSM di node pekerja Amazon EKS dengan menggunakan Kubernetes DaemonSet
- Instal Agen dan CloudWatch agen SSM di node pekerja Amazon EKS menggunakan preBootstrapCommands
- Migrasikan Pengontrol Ingress NGINX saat mengaktifkan Mode Otomatis Amazon EKS
- Migrasikan beban kerja kontainer Anda dari Azure Red Hat OpenShift (ARO) ke (ROSA Layanan OpenShift Red Hat di AWS)
- Jalankan tugas Amazon ECS di Amazon WorkSpaces dengan Amazon ECS Anywhere
- Jalankan wadah API Docker web ASP.NET Core pada instance Amazon Linux EC2
- Jalankan beban kerja stateful dengan penyimpanan data persisten dengan menggunakan Amazon EFS di Amazon EKS dengan AWS Fargate
- Siapkan penskalaan otomatis berbasis peristiwa di Amazon EKS dengan menggunakan Amazon EKS Pod Identity dan KEDA
- Merampingkan penerapan PostgreSQL di Amazon EKS dengan menggunakan PGO
- Sederhanakan otentikasi aplikasi dengan TLS timbal balik di Amazon ECS dengan menggunakan Application Load Balancer
- Lebih banyak pola

Akses database Amazon Neptunus dari wadah Amazon EKS

Ramakrishnan Palaninathan, Amazon Web Services

Ringkasan

Pola ini membuat koneksi antara Amazon Neptunus, yang merupakan database grafik yang dikelola sepenuhnya, dan Amazon Elastic Kubernetes Service (Amazon EKS), layanan orkestrasi kontainer, untuk mengakses database Neptunus. Cluster DB Neptunus terbatas dalam virtual private cloud (VPC) aktif. AWS Untuk alasan ini, mengakses Neptunus memerlukan konfigurasi VPC yang cermat untuk mengaktifkan konektivitas.

Tidak seperti Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL, Neptune tidak bergantung pada kredensyal akses database yang khas. Sebaliknya, ia menggunakan peran AWS Identity and Access Management (IAM) untuk otentikasi. Oleh karena itu, menghubungkan ke Neptunus dari Amazon EKS melibatkan pengaturan peran IAM dengan izin yang diperlukan untuk mengakses Neptunus.

Selain itu, titik akhir Neptunus hanya dapat diakses dalam VPC tempat cluster berada. Ini berarti Anda harus mengonfigurasi pengaturan jaringan untuk memfasilitasi komunikasi antara Amazon EKS dan Neptunus. Bergantung pada kebutuhan spesifik dan preferensi jaringan Anda, ada berbagai pendekatan untuk mengonfigurasi VPC untuk memungkinkan konektivitas tanpa batas antara Neptunus dan Amazon EKS. Setiap metode menawarkan keuntungan dan pertimbangan yang berbeda, yang memberikan fleksibilitas dalam merancang arsitektur database Anda agar sesuai dengan kebutuhan aplikasi Anda.

Prasyarat dan batasan

Prasyarat

- Instal kubectl versi terbaru (lihat instruksi). Untuk memeriksa versi Anda, jalankan:

```
kubectl version --short
```

- Instal versi terbaru eksctl (lihat instruksi). Untuk memeriksa versi Anda, jalankan:

```
eksctl info
```

- Instal versi terbaru dari AWS Command Line Interface (AWS CLI) versi 2 (lihat instruksi). Untuk memeriksa versi Anda, jalankan:

```
aws --version
```

- Buat cluster DB Neptunus (lihat instruksi). Pastikan untuk membuat komunikasi antara VPC cluster dan Amazon EKS melalui VPC peering,, AWS Transit Gatewayatau metode lain. Pastikan juga bahwa status cluster “tersedia” dan memiliki aturan masuk pada port 8182 untuk grup keamanan.
- Konfigurasikan penyedia OpenID Connect (OIDC) IAM di kluster Amazon EKS yang ada (lihat instruksi).

Versi produk

- Amazon EKS 1.27
- Mesin Amazon Neptunus versi 1.3.0.0 (2023-11-15)

Arsitektur

Diagram berikut menunjukkan hubungan antara pod Kubernetes di cluster Amazon EKS dan Neptunus untuk menyediakan akses ke database Neptunus.

Otomatisasi dan skala

Anda dapat menggunakan Amazon EKS Horizontal Pod Autoscaler untuk menskalakan solusi ini.

Alat

Layanan

- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Neptunus adalah layanan database grafik yang membantu Anda membangun dan menjalankan aplikasi yang bekerja dengan kumpulan data yang sangat terhubung.

Praktik terbaik

Untuk praktik terbaik, lihat Identity and Access Management di Panduan Praktik Terbaik Amazon EKS.

Epik

Tetapkan variabel lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi konteks cluster.	Sebelum berinteraksi dengan klaster Amazon EKS menggunakan Helm atau alat baris perintah lainnya, Anda harus menentukan variabel lingkungan yang merangkum detail klaster Anda. Variabel-variabel ini digunakan dalam perintah berikutnya untuk memastikan bahwa mereka menargetkan cluster dan sumber daya yang benar. Pertama, konfirmasikan bahwa Anda beroperasi dalam konteks cluster yang benar. Ini memastikan bahwa setiap perintah berikutnya dikirim ke cluster Kubernetes yang dimaksud. Untuk memverifikasi konteks saat ini, jalankan perintah berikut. `kubectl config current-context`	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan `CLUSTER_NAME` variabelnya.	Tentukan variabel `CLUSTER_NAME` lingkungan untuk klaster Amazon EKS Anda. Dalam perintah berikut, ganti nilai sampel `us-west-2` dengan yang benar Wilayah AWS untuk cluster Anda. Ganti nilai sampel `eks-workshop` dengan nama cluster yang ada. `export CLUSTER_NAME=$(aws eks describe-cluster --region us-west-2 --name eks-workshop --query "cluster.name" --output text)`	Administrator AWS, Administrator cloud
Validasi keluaran.	Untuk memvalidasi bahwa variabel telah diatur dengan benar, jalankan perintah berikut. `echo $CLUSTER_NAME` Verifikasi bahwa output dari perintah ini cocok dengan input yang Anda tentukan pada langkah sebelumnya.	Administrator AWS, Administrator cloud

Buat peran IAM dan kaitkan dengan Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat akun layanan.	Anda menggunakan peran IAM untuk akun layanan untuk memetakan akun layanan Kubernetes Anda ke peran IAM, untuk mengaktifkan manajemen izin berbutir halus untuk aplikasi Anda yang berjalan di Amazon EKS. Anda dapat menggunakan eksctl untuk membuat dan mengaitkan peran IAM dengan akun layanan Kubernetes tertentu dalam klaster Amazon EKS Anda. Kebijakan AWS terkelola `NeptuneFullAccess` memungkinkan akses tulis dan baca ke klaster Neptunus yang Anda tentukan. **Important** Anda harus memiliki endpoint OIDC yang terkait dengan cluster Anda sebelum menjalankan perintah ini. Buat akun layanan yang ingin Anda kaitkan dengan kebijakan AWS terkelola	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	bernamaNeptuneFullAccess .
	
	`eksctl create iamserviceaccount --name eks-neptune-sa --namespace default --cluster $CLUSTER_NAME --attach-policy-arn arn:aws:iam::aws:policy/NeptuneFullAccess --approve --override-existing-serviceaccounts`
	
	di `eks-neptune-sa` mana nama akun layanan yang ingin Anda buat.
	
	Setelah selesai, perintah ini menampilkan respons berikut:
	
	`2024-02-07 01:12:39 [#] created serviceaccount "default/eks-neptune-sa"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi bahwa akun sudah diatur dengan benar.	Pastikan akun `eks-neptune-sa` layanan diatur dengan benar di namespace default di cluster Anda.
	`kubectl get sa eks-neptune-sa -o yaml`
	Outputnya akan terlihat seperti ini:
	`apiVersion: v1`
	`kind: ServiceAccount`
	`metadata:`
	`  annotations:`
	`  eks.amazonaws.com/role-arn: arn:aws:iam::123456789123:role/eksctl-eks-workshop-addon-iamserviceaccount-d-Role1-Q35yKgdQOlmM`
	`  creationTimestamp: "2024-02-07T01:12:39Z"`
	`  labels:`
	`  app.kubernetes.io/managed-by: eksctl`
	`  name: eks-neptune-sa`
	`  namespace: default`
	`  resourceVersion: "5174750"`
	`  uid: cd6ba2f7-a0f5-40e1-a6f4-4081e0042316`
		Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa konektivitas.	Terapkan pod sampel yang disebut `pod-util` dan periksa konektivitas dengan Neptunus.
	
	`apiVersion: v1`
	`kind: Pod`
	`metadata:`
	`  name: pod-util`
	`  namespace: default`
	`spec:`
	`  serviceAccountName: eks-neptune-sa`
	`  containers:`
	`  - name: pod-util`
	`  image: public.ecr.aws/patrickc/troubleshoot-util`
	`  command:`
	`  - sleep`
	`  - "3600"`
	`  imagePullPolicy: IfNotPresent`
	
	`kubectl apply -f pod-util.yaml`
	
	`kubectl exec --stdin --tty pod-util -- /bin/bash`
	`bash-5.1# curl -X POST -d '{"gremlin":"g.V().limit(1)"}' https://db-neptune-1.cluster-xxxxxxxxxxxx.us-west-2.neptune.amazonaws.com:8182/gremlin`
		Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`{"requestId":"a4964f2 d-12b1-4ed3-8a14-e ff511431a0e","stat us":{"message":"", "code":200,"attrib utes":{"@type":"g: Map","@value":[]}} ,"result":{"data": {"@type":"g:List", "@value":[]},"meta ":{"@type":"g:Map" ,"@value":[]}}} bash-5.1# exit exit`	

Validasi aktivitas koneksi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan otentikasi database IAM.	Secara default, otentikasi database IAM dinonaktifkan saat Anda membuat cluster DB Neptunus. Anda dapat mengaktifkan atau menonakti fkan otentikasi database IAM dengan menggunakan file. Konsol Manajemen AWS Ikuti langkah-langkah dalam AWS dokumentasi untuk mengaktifkan otentikasi database IAM di Neptunus.	Administrator AWS, Administr ator cloud
Verifikasi koneksi.	Pada langkah ini, Anda berinteraksi dengan `pod-util` wadah, yang sudah dalam status berjalan, untuk	Administrator AWS, Administr ator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	menginstal awscurl dan memverifikasi koneksi.
	
	1. Jalankan perintah berikut untuk menemukan pod.
	
	`kubectl get pods`
	
	Outputnya akan terlihat seperti ini:
	
	`NAME READY STATUS RESTARTS AGE pod-util 1/1 Running 0 50m`
	
	2. Jalankan perintah berikut untuk menginstal awscurl.
	
	`kubectl exec --stdin --tty pod-util -- /bin/bash bash-5.1# pip3 install awscurl Installing collected packages: idna, configparser, configargparse, charset-normalizer, certifi, requests, awscurl Successfully installed awscurl-0.32 certifi-2024.2.2 charset-normalizer-3.3.2 configargparse-1.7 configparser-6.0.0 idna-3.6 requests-2.31.0 bash-5.1# awscurl https://db-neptune-1.cluster-xxxxxxx`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	xxxxx.us-west-2.neptune.amazonaws.com:8182/status --region us-west-2 --service neptune-db {"status":"healthy","startTime":"Thu Feb 08 01:22:14 UTC 2024","dbEngineVersion":"1.3.0.0.R1","role":"writer","dfeQueryEngine":"viaQueryHint","gremlin":{"version":"tinkerpop-3.6.4"},"sparql":{"version":"sparql-1.1"},"opencypher":{"version":"Neptune-9.0.20190305-1.0"},"labMode":{"ObjectIndex":"disabled","ReadWriteConflictDetection":"enabled"},"features":{"SlowQueryLogs":"disabled","ResultCache":{"status":"disabled"},"IAMAuthentication":"enabled","Streams":"disabled","AuditLog":"disabled"},"settings":{"clusterQueryTimeoutInMs":"120000","SlowQueryLogsThreshold":"5000"}}	

Pemecahan masalah

Isu	Solusi
Tidak dapat mengakses database Neptunus.	Tinjau kebijakan IAM yang dilampirkan ke akun layanan. Pastikan bahwa itu memungkin kan tindakan yang diperlukan (misalnya ,`neptune:Connec`,`neptune:Desc ribeDBInstances`) untuk operasi yang ingin Anda jalankan.

Sumber daya terkait

- Berikan akses beban kerja Kubernetes untuk AWS menggunakan Akun Layanan Kubernetes (dokumentasi Amazon EKS)
- Peran IAM untuk akun layanan (dokumentasi Amazon EKS)
- Membuat cluster DB Neptunus baru (dokumentasi Amazon Neptunus)

Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS PrivateLink dan Network Load Balancer

Kirankumar Chandrashekar, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara meng-host aplikasi kontainer Docker secara pribadi di Amazon Elastic Container Service (Amazon ECS) di belakang Network Load Balancer, dan mengakses aplikasi dengan menggunakan AWS. PrivateLink Anda kemudian dapat menggunakan jaringan pribadi untuk mengakses layanan dengan aman di Amazon Web Services (AWS) Cloud. Amazon Relational Database Service (Amazon RDS) menghosting database relasional untuk aplikasi yang berjalan di Amazon ECS dengan ketersediaan tinggi (HA). Amazon Elastic File System (Amazon EFS) digunakan jika aplikasi memerlukan penyimpanan persisten.

Layanan Amazon ECS yang menjalankan aplikasi Docker, dengan Network Load Balancer di ujung depan, dapat dikaitkan dengan titik akhir virtual private cloud (VPC) untuk akses melalui AWS. PrivateLink Layanan endpoint VPC ini kemudian dapat dibagikan dengan orang lain VPCs dengan menggunakan titik akhir VPC mereka.

Anda juga dapat menggunakan AWS Fargate alih-alih grup Amazon Auto EC2 Scaling. Untuk informasi selengkapnya, lihat Mengakses aplikasi container secara pribadi di Amazon ECS dengan menggunakan AWS Fargate, PrivateLink AWS, dan Network Load Balancer.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi di Linux, macOS, atau Windows
- Docker, diinstal dan dikonfigurasi di Linux, macOS, atau Windows
- Aplikasi yang berjalan di Docker

Arsitektur

Tumpukan teknologi

- Amazon CloudWatch
- Amazon Elastic Compute Cloud (Amazon EC2)
- EC2 Auto Scaling Amazon
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- Amazon RDS
- Amazon Simple Storage Service (Amazon S3)
- AWS Lambda
- AWS PrivateLink
- AWS Secrets Manager
- Penyeimbang Beban Aplikasi
- Penyeimbang Beban Jaringan
- VPC

Otomatisasi dan skala

- Anda dapat menggunakan AWS CloudFormation untuk membuat pola ini dengan menggunakan Infrastructure as Code.

Alat

- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud.
- EC2 Auto Scaling Amazon — Amazon EC2 Auto Scaling membantu Anda memastikan bahwa Anda memiliki jumlah instans EC2 Amazon yang benar yang tersedia untuk menangani pemuatan aplikasi Anda.
- Amazon ECS - Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel, cepat, yang memudahkan untuk menjalankan, menghentikan, dan mengelola kontainer di cluster.
- Amazon ECR — Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri image container AWS terkelola yang aman, terukur, dan andal.

- Amazon EFS — Amazon Elastic File System (Amazon EFS) menyediakan sistem file NFS elastis yang sederhana, dapat diskalakan, dan dikelola sepenuhnya untuk digunakan dengan layanan AWS Cloud dan sumber daya lokal.
- AWS Lambda — Lambda adalah layanan komputasi untuk menjalankan kode tanpa menyediakan atau mengelola server.
- Amazon RDS - Amazon Relational Database Service (Amazon RDS) adalah layanan web yang memudahkan pengaturan, pengoperasian, dan skala database relasional di AWS Cloud.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet. Ini dirancang untuk membuat komputasi skala web lebih mudah bagi pengembang.
- AWS Secrets Manager — Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan menyediakan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan.
- Elastic Load Balancing — Elastic Load Balancing mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target, seperti instans EC2 Amazon, kontainer, dan alamat IP, di beberapa Availability Zone.
- Docker — Docker membantu pengembang untuk mengemas, mengirim, dan menjalankan aplikasi apa pun sebagai wadah yang ringan, portabel, dan mandiri.

Epik

Buat komponen jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	1. Masuk ke AWS Management Console dan buka konsol Amazon VPC. Pilih Buat VPC, lalu pilih VPC dan lainnya. 2. Masukkan nama untuk VPC Anda, dan pilih rentang blok CIDR yang sesuai.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Tentukan dua Availability Zones, dua subnet publik, empat subnet pribadi. Dua subnet pribadi untuk tugas Amazon ECS, dan dua subnet pribadi untuk database Amazon RDS. 4. Tentukan satu gateway NAT untuk setiap Availabil ity Zone. 5. Pilih Buat VPC.	

Buat penyeimbang beban

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Network Load Balancer.	1. Buka EC2 konsol Amazon dan pilih Wilayah AWS yang berisi VPC Anda. 2. Di bawah Load balancing , pilih Load balancer, dan pilih Create load balancer. 3. Pilih Network Load Balancer, dan pilih Buat. 4. Pada halaman Configure load balancer, konfigura sikan Network Load Balancer dan listener Anda. Penting: Pastikan Anda memilih skema Network Load Balancer sebagai Internal.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Pilih pengaturan keamanan yang berlaku, konfigura sikan grup keamanan dan grup target. Pilih Instance atau IP sebagai tipe Target di bagian Configure routing. Pastikan Anda tidak mendaftarkan target. 6. Ketika Anda telah mengonfigurasi semua pengaturan, pilih Berikutny a: Tinjau, lalu pilih Buat.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Application Load Balancer.	1. Di EC2 konsol Amazon, pilih Wilayah yang sama yang berisi VPC Anda. 2. Di bawah Load balancing , pilih Load balancer, dan pilih Create load balancer. 3. Pilih Application Load Balancer, dan pilih Create. 4. **Important** Konfigurasikan Application Load Balancer Anda dan pendengar nya. Pastikan Anda memilih skema Application Load Balancer sebagai Internal. 5. Pilih pengaturan keamanan yang berlaku, konfigura sikan grup keamanan dan grup target. Pilih Instance atau IP sebagai tipe Target di bagian Configure routing. Pastikan Anda tidak mendaftarkan target. 6. Ketika Anda telah mengonfigurasi semua pengaturan, pilih Berikutny a: Tinjau, lalu pilih Buat.	Administrator awan

Buat sistem file Amazon EFS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file Amazon EFS.	1. Buka konsol Amazon EFS, dan pilih Buat sistem file. 2. Dalam kotak dialog Buat sistem file, masukkan nama untuk sistem file Anda, dan pilih VPC Anda. 3. Pilih Buat untuk membuat sistem file. 4. Siapkan dan konfigurasikan sistem file Amazon EFS Anda.	Administrator awan
Pasang target untuk subnet.	1. Kembali ke konsol Amazon EFS, dan pilih Sistem file. Halaman sistem File menunjukkan sistem file Amazon EFS di akun Anda. 2. Pilih sistem file yang Anda buat, dan pilih Kelola untuk menampilkan Availability Zones. Untuk menambahk an target mount, pilih Add mount target, dan tambahkan empat subnet pribadi yang Anda buat.	Administrator awan
Verifikasi bahwa subnet dipasang sebagai target.	1. Di konsol Amazon EFS, pilih Sistem file. 2. Pilih Jaringan untuk menampilkan daftar target pemasangan yang	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ada. Pastikan bahwa ini termasuk empat subnet yang Anda buat.	

Buat Bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buka konsol Amazon S3 dan buat bucket S3 untuk menyimpan aset statis aplikasi Anda, jika diperlukan.	Administrator awan

Buat rahasia Secrets Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kunci AWS KMS untuk mengenkripsi rahasia Secrets Manager.	Buka konsol AWS Key Management Service (AWS KMS) dan buat kunci KMS.	Administrator awan
Buat rahasia Secrets Manager untuk menyimpan kata sandi Amazon RDS.	1. Buka konsol AWS Secrets Manager, dan buat rahasia baru dengan memilih Simpan rahasia baru. 2. Pilih kunci KMS yang Anda buat, dan simpan rahasia baru Anda.	Administrator Awan

Buat instans Amazon RDS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup subnet DB.	1. Buka konsol Amazon RDS dan pilih grup Subnet. 2. Pilih Buat grup subnet DB, dan masukkan nama dan deskripsi untuk grup subnet DB Anda. 3. Pilih VPC yang Anda buat sebelumnya, dan pilih Availability Zones dan subnet. Lalu pilih Buat.	Administrator awan
Buat instance Amazon RDS.	Buat dan konfigurasikan instans Amazon RDS dalam subnet pribadi. Pastikan Multi-AZ dihidupkan untuk HA.	Administrator awan
Muat data ke instans Amazon RDS.	Muat data relasional yang diperlukan oleh aplikasi Anda ke instans Amazon RDS Anda. Proses ini akan bervariasi tergantung pada kebutuhan aplikasi Anda, serta bagaimana skema database Anda didefinisikan dan dirancang.	Administrator awan, DBA

Buat komponen Amazon ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster ECS.	1. Buka konsol Amazon ECS, dan pilih Clusters.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pilih Buat cluster, dan siapkan cluster ECS sesuai dengan spesifikasi yang Anda butuhkan.	
Buat gambar Docker.	Buat gambar Docker dengan mengikuti petunjuk di bagian Sumber daya terkait.	Administrator awan
Buat repositori Amazon ECR.	1. Di konsol Amazon ECR, pilih Repositori. 2. Pilih Buat repositori, dan masukkan nama unik untuk repositori Anda. 3. Konfigurasikan repositor i sesuai dengan spesifika si Anda, termasuk enkripsi AWS KMS jika diperlukan.	Administrator cloud, DevOps insinyur
Otentikasi klien Docker Anda untuk repositori Amazon ECR.	Untuk mengautentikasi klien Docker Anda untuk repositor i Amazon ECR, jalankan perintah `aws ecr get-login-password "` di AWS CLI.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dorong gambar Docker ke repositori Amazon ECR.	1. Identifikasi image Docker yang ingin Anda dorong, dan jalankan `docker images` perintah di AWS CLI. 2. Tandai gambar Anda dengan registri Amazon ECR, repositori, dan kombinasi nama tag gambar opsional. 3. Dorong gambar Docker dengan menjalankan `docker push` perintah. 4. Ulangi langkah-langkah ini untuk semua gambar yang diperlukan.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat definisi tugas Amazon ECS.	Definisi tugas diperlukan untuk menjalankan kontainer Docker di Amazon ECS. 1. Kembali ke konsol Amazon ECS, pilih Definisi tugas, lalu pilih Buat definisi tugas baru. 2. Pada halaman Pilih kompatibilitas, pilih jenis peluncuran yang harus digunakan tugas Anda, dan pilih Langkah berikutnya. **Important** Untuk bantuan dalam mengatur definisi tugas, lihat “Membuat definisi tugas” di bagian Sumber daya terkait. Pastikan Anda memberikan gambar Docker yang Anda dorong ke Amazon ECR.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat layanan Amazon ECS.	Buat layanan Amazon ECS dengan menggunakan cluster ECS yang Anda buat sebelumnya. Pastikan Anda memilih Amazon EC2 sebagai tipe peluncuran, dan pilih definisi tugas yang dibuat pada langkah sebelumnya, serta kelompok target Application Load Balancer.	Administrator awan

Buat grup EC2 Auto Scaling Amazon

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat konfigurasi peluncuran.	Buka EC2 konsol Amazon, dan buat konfigurasi peluncuran. Pastikan bahwa data pengguna memiliki kode untuk memungkinkan EC2 instance bergabung dengan cluster ECS yang diinginkan. Untuk contoh kode yang diperlukan, lihat bagian Sumber daya terkait.	Administrator awan
Buat grup EC2 Auto Scaling Amazon.	Kembali ke EC2 konsol Amazon dan di bawah Auto Scaling, pilih grup Auto Scaling. Siapkan grup EC2 Auto Scaling Amazon. Pastikan Anda memilih subnet pribadi dan meluncurkan	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	konfigurasi yang Anda buat sebelumnya.	

Siapkan AWS PrivateLink

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan PrivateLink titik akhir AWS.	1. Di konsol VPC Amazon, buat titik akhir PrivateLink AWS. 2. Kaitkan titik akhir ini dengan Network Load Balancer, yang membuat aplikasi yang dihosting di Amazon ECS tersedia secara pribadi bagi pelanggan. Untuk informasi selengkap nya, lihat bagian Sumber daya terkait.	Administrator awan

Buat VPC endpoint

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC endpoint	Buat titik akhir VPC untuk PrivateLink titik akhir AWS yang Anda buat sebelumnya. Endpoint VPC Fully Qualified Domain Name (FQDN) akan mengarah ke FQDN titik akhir AWS. PrivateLink Ini menciptakan sebuah elastic	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	network interface ke layanan endpoint VPC yang dapat diakses oleh endpoint DNS.	

Buat fungsi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda.	Di konsol AWS Lambda, buat fungsi Lambda untuk memperbarui alamat IP Application Load Balancer sebagai target untuk Network Load Balancer. Untuk informasi selengkapnya tentang ini, lihat posting blog Menggunakan AWS Lambda untuk mengaktifkan alamat IP statis untuk Application Load Balancers.	Pengembang aplikasi

Sumber daya terkait

Buat penyeimbang beban:

- Menggunakan Network Load Balancer untuk Amazon ECS
- Membuat Network Load Balancer
- Menggunakan Application Load Balancer untuk Amazon ECS
- Buat Application Load Balancer

Buat sistem file Amazon EFS:

- Buat sistem file Amazon EFS

- Buat target pemasangan di Amazon EFS

Buat ember S3:

- Buat ember S3

Buat rahasia Secrets Manager:

- Buat kunci di AWS KMS
- Buat rahasia di AWS Secrets Manager

Buat instance Amazon RDS:

- Buat instans Amazon RDS DB

Buat komponen Amazon ECS:

- Buat cluster Amazon ECS
- Buat gambar Docker
- Buat repositori Amazon ECR
- Otentikasi Docker dengan repositori Amazon ECR
- Dorong gambar ke repositori Amazon ECR
- Buat definisi tugas Amazon ECS
- Buat layanan Amazon ECS

Buat grup EC2 Auto Scaling Amazon:

- Buat konfigurasi peluncuran
- Membuat grup Auto Scaling menggunakan konfigurasi peluncuran
- Contoh wadah bootstrap dengan data EC2 pengguna Amazon

Siapkan AWS PrivateLink:

- Layanan titik akhir VPC (AWS) PrivateLink

Buat titik akhir VPC:

- Titik akhir VPC Antarmuka (AWS) PrivateLink

Buat fungsi Lambda:

- Buat fungsi Lambda

Sumber daya lainnya:

- Menggunakan alamat IP statis untuk Application Load Balancers
- Mengakses layanan dengan aman melalui AWS PrivateLink

Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS Fargate, PrivateLink AWS, dan Network Load Balancer

Kirankumar Chandrashekar, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara meng-host aplikasi kontainer Docker secara pribadi di Amazon Web Services (AWS) Cloud dengan menggunakan Amazon Elastic Container Service (Amazon ECS) Container Service (Amazon ECS) dengan tipe peluncuran AWS Fargate, di belakang Network Load Balancer, dan mengakses aplikasi dengan menggunakan AWS. PrivateLink Amazon Relational Database Service (Amazon RDS) menghosting database relasional untuk aplikasi yang berjalan di Amazon ECS dengan ketersediaan tinggi (HA). Anda dapat menggunakan Amazon Elastic File System (Amazon EFS) jika aplikasi memerlukan penyimpanan persisten.

Pola ini menggunakan tipe peluncuran Fargate untuk layanan Amazon ECS yang menjalankan aplikasi Docker, dengan Network Load Balancer di ujung depan. Ini kemudian dapat dikaitkan dengan titik akhir virtual private cloud (VPC) untuk akses melalui AWS. PrivateLink Layanan endpoint VPC ini kemudian dapat dibagikan dengan orang lain VPCs dengan menggunakan titik akhir VPC mereka.

Anda dapat menggunakan Fargate dengan Amazon ECS untuk menjalankan container tanpa harus mengelola server atau cluster instans Amazon Elastic Compute Cloud (EC2Amazon). Anda juga dapat menggunakan grup Amazon EC2 Auto Scaling alih-alih Fargate. Untuk informasi selengkapnya, lihat Mengakses aplikasi container secara pribadi di Amazon ECS dengan menggunakan AWS PrivateLink dan Network Load Balancer.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi di Linux, macOS, atau Windows
- Docker, diinstal dan dikonfigurasi di Linux, macOS, atau Windows
- Aplikasi yang berjalan di Docker

Arsitektur

Tumpukan teknologi

- Amazon CloudWatch
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- Amazon EFS
- Amazon RDS
- Amazon Simple Storage Service (Amazon S3)
- AWS Fargate
- AWS PrivateLink
- AWS Secrets Manager
- Penyeimbang Beban Aplikasi
- Penyeimbang Beban Jaringan
- VPC

Otomatisasi dan skala

- Anda dapat menggunakan AWS CloudFormation untuk membuat pola ini dengan menggunakan Infrastructure as Code.

Alat

Layanan AWS

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri image container AWS terkelola yang aman, terukur, dan andal.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel, cepat, yang memudahkan untuk menjalankan, menghentikan, dan mengelola kontainer di cluster.
- Amazon Elastic File System (Amazon EFS) menyediakan sistem file NFS elastis yang sederhana, dapat diskalakan, dan dikelola sepenuhnya untuk digunakan dengan layanan AWS Cloud dan sumber daya lokal.

- AWS Fargateadalah teknologi yang dapat Anda gunakan dengan Amazon ECS untuk menjalankan container tanpa harus mengelola server atau cluster instance Amazon EC2 .
- Amazon Relational Database Service (Amazon RDS) adalah layanan web yang membuatnya lebih mudah untuk mengatur, mengoperasikan, dan menskalakan basis data relasional di. AWS Cloud
- Amazon Simple Storage Service (Amazon S3) adalah penyimpanan untuk internet. Ini dirancang untuk membuat komputasi skala web lebih mudah bagi pengembang.
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target, seperti EC2 instance, kontainer, dan alamat IP, di beberapa Availability Zone.

Alat-alat lainnya

- Docker membantu pengembang untuk dengan mudah mengemas, mengirim, dan menjalankan aplikasi apa pun sebagai wadah yang ringan, portabel, dan mandiri.

Epik

Buat komponen jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	1. Masuk ke AWS Management Console, dan buka konsol Amazon VPC. Pilih Buat VPC, lalu pilih VPC dan lainnya. 2. Masukkan nama untuk VPC Anda, dan pilih rentang blok CIDR yang sesuai. 3. Tentukan dua Availability Zones, dua subnet publik,	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	empat subnet pribadi. Dua subnet pribadi untuk tugas Amazon ECS, dan dua subnet pribadi untuk database Amazon RDS. 4. Tentukan satu gateway NAT untuk setiap Availabil ity Zone. 5. Pilih Buat VPC.	

Buat penyeimbang beban

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Network Load Balancer.	1. Buka EC2 konsol Amazon, dan pilih Wilayah AWS yang berisi VPC Anda. 2. Di bawah Load balancing , pilih Load balancer, dan pilih Create load balancer. 3. Pilih Network Load Balancer, dan pilih Buat. 4. Pada halaman Configure load balancer, konfigura sikan Network Load Balancer dan listener Anda. Penting: Pastikan Anda memilih skema Network Load Balancer sebagai Internal. 5. Pilih pengaturan keamanan yang berlaku, konfigura sikan grup keamanan dan	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	grup target. Pilih IP sebagai tipe Target di bagian Configure routing. Pastikan Anda tidak mendaftarkan target. 6. Ketika Anda telah mengonfigurasi semua pengaturan, pilih Berikutny a: Tinjau, lalu pilih Buat. Untuk bantuan tentang ini dan cerita lainnya, lihat bagian Sumber daya terkait.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Application Load Balancer.	1. Di EC2 konsol Amazon, pilih Wilayah yang sama yang berisi VPC Anda. 2. Di bawah Load balancing , pilih Load balancer, dan pilih Create load balancer. 3. Pilih Application Load Balancer, dan pilih Create. 4. **Important** Konfigurasikan Application Load Balancer Anda dan pendengar nya. Pastikan Anda memilih skema Application Load Balancer sebagai Internal. 5. Pilih pengaturan keamanan yang berlaku, konfigura sikan grup keamanan dan grup target. Pilih IP sebagai tipe Target di bagian Configure routing. Pastikan Anda tidak mendaftarkan target. 6. Ketika Anda telah mengonfigurasi semua pengaturan, pilih Berikutny a: Tinjau, lalu pilih Buat.	Administrator awan

Buat sistem file Amazon EFS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file Amazon EFS.	1. Buka konsol Amazon EFS, dan pilih Buat sistem file. 2. Dalam kotak dialog Buat sistem file, masukkan nama untuk sistem file Anda, dan pilih VPC Anda. 3. Pilih Buat untuk membuat sistem file. 4. Siapkan dan konfigurasikan sistem file Amazon EFS Anda.	Administrator awan
Pasang target untuk subnet.	1. Kembali ke konsol Amazon EFS, dan pilih Sistem file. Halaman sistem File menunjukkan sistem file Amazon EFS di akun Anda. 2. Pilih sistem file yang Anda buat, dan pilih Kelola untuk menampilkan Availability Zone. 3. Untuk menambahkan target mount, pilih Add mount target, dan tambahkan empat subnet pribadi yang Anda buat.	Administrator awan
Verifikasi bahwa subnet dipasang sebagai target.	1. Di konsol Amazon EFS, pilih Sistem file. 2. Pilih Jaringan untuk menampilkan daftar	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	target pemasangan yang ada. Pastikan bahwa ini termasuk empat subnet yang Anda buat.	

Buat Bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buka konsol Amazon S3 dan buat bucket S3 untuk menyimpan aset statis aplikasi Anda, jika diperlukan.	Administrator awan

Buat rahasia Secrets Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kunci AWS KMS untuk mengenkripsi rahasia Secrets Manager.	Buka konsol AWS Key Management Service (AWS KMS) dan buat kunci KMS.	Administrator awan
Buat rahasia Secrets Manager untuk menyimpan kata sandi Amazon RDS.	1. Buka konsol AWS Secrets Manager, dan buat rahasia baru dengan memilih Simpan rahasia baru. 2. Pilih kunci KMS yang Anda buat, dan simpan rahasia baru Anda.	Administrator awan

Buat instans Amazon RDS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup subnet DB.	1. Buka konsol Amazon RDS, dan pilih grup Subnet. 2. Pilih Buat grup subnet DB, dan masukkan nama dan deskripsi untuk grup subnet DB Anda. 3. Pilih VPC yang Anda buat sebelumnya, dan pilih Availability Zones dan subnet. Lalu pilih Buat.	Administrator awan
Buat instans Amazon RDS.	Buat dan konfigurasikan instans Amazon RDS dalam subnet pribadi. Pastikan Multi-AZ dihidupkan untuk ketersediaan tinggi (HA).	Administrator awan
Muat data ke instans Amazon RDS.	Muat data relasional yang diperlukan oleh aplikasi Anda ke instans Amazon RDS Anda. Proses ini akan bervariasi tergantung pada kebutuhan aplikasi Anda, serta bagaimana skema database Anda didefinisikan dan dirancang.	DBA

Buat komponen Amazon ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster ECS.	1. Buka konsol Amazon ECS, dan pilih Clusters. 2. Pilih Buat cluster, dan siapkan cluster ECS sesuai dengan spesifikasi yang Anda butuhkan.	Administrator awan
Buat gambar Docker.	Buat gambar Docker dengan mengikuti petunjuk dalam AWS dokumentasi.	Administrator awan
Buat repositori Amazon ECR.	1. Buka konsol Amazon ECR, dan pilih Repositori. 2. Pilih Buat repositori, dan masukkan nama unik untuk repositori Anda. 3. Konfigurasikan repositor i sesuai dengan spesifika si Anda, termasuk enkripsi AWS KMS jika diperlukan.	Administrator cloud, DevOps insinyur
Dorong gambar Docker ke repositori Amazon ECR.	1. Identifikasi image Docker yang ingin Anda dorong, dan jalankan `docker images` perintah di AWS CLI. 2. Tandai gambar Anda dengan registri Amazon ECR, repositori, dan kombinasi nama tag gambar opsional.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Dorong gambar Docker dengan menjalankan `docker push` perintah. 4. Ulangi langkah-langkah ini untuk semua gambar yang diperlukan.	
Buat definisi tugas Amazon ECS.	Definisi tugas diperlukan untuk menjalankan kontainer Docker di Amazon ECS. 1. Kembali ke konsol Amazon ECS, pilih Definisi tugas, lalu pilih Buat definisi tugas baru. 2. Pada halaman Pilih kompatibilitas, pilih jenis peluncuran yang harus digunakan tugas Anda, dan pilih Langkah berikutnya. **Important** Untuk bantuan dalam mengatur definisi tugas, lihat “Membuat definisi tugas” di bagian Sumber daya terkait. Pastikan Anda memberikan gambar Docker yang Anda dorong ke Amazon ECR.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat layanan ECS dan pilih Fargate sebagai tipe peluncuran.	1. Buat layanan Amazon ECS dengan menggunakan cluster ECS yang Anda buat sebelumnya. Pastikan Anda memilih Fargate sebagai tipe peluncuran. 2. Pilih definisi tugas yang dibuat pada langkah sebelumnya, dan pilih kelompok target Application Load Balancer.	Administrator awan

Siapkan AWS PrivateLink

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan PrivateLink titik akhir AWS.	1. Buka konsol Amazon VPC, dan buat titik akhir. AWS PrivateLink 2. Kaitkan titik akhir ini dengan Network Load Balancer, yang membuat aplikasi yang dihosting di Amazon ECS tersedia secara pribadi bagi pelanggan.	Administrator awan

Buat VPC endpoint

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC endpoint	Buat titik akhir VPC untuk PrivateLink titik akhir AWS yang Anda buat sebelumnya.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Endpoint VPC Fully Qualified Domain Name (FQDN) akan mengarah ke FQDN titik akhir AWS. PrivateLink Ini menciptakan sebuah elastic network interface ke layanan endpoint VPC yang dapat diakses oleh endpoint Domain Name Service.	

Tetapkan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan Application Load Balancer sebagai target.	Untuk menambahkan Application Load Balancer sebagai target Network Load Balancer, ikuti petunjuk dalam dokumentasi.AWS	Pengembang aplikasi

Sumber daya terkait

Buat penyeimbang beban:

- Menggunakan Network Load Balancer untuk Amazon ECS
- Membuat Network Load Balancer
- Menggunakan Application Load Balancer untuk Amazon ECS
- Buat Application Load Balancer

Buat sistem file Amazon EFS:

- Buat sistem file Amazon EFS
- Buat target pemasangan di Amazon EFS

Buat rahasia Secrets Manager:

- Buat kunci di AWS KMS
- Buat rahasia di AWS Secrets Manager

Buat instance Amazon RDS:

- Buat instans Amazon RDS DB

Buat komponen Amazon ECS

- Buat repositori Amazon ECR
- Otentikasi Docker dengan repositori Amazon ECR
- Dorong gambar ke repositori Amazon ECR
- Buat definisi tugas Amazon ECS
- Buat layanan Amazon ECS

Sumber daya lainnya:

- Mengakses layanan dengan aman melalui AWS PrivateLink

Akses aplikasi kontainer secara pribadi di Amazon EKS menggunakan AWS PrivateLink dan Network Load Balancer

Kirankumar Chandrashekar, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara meng-host aplikasi kontainer Docker secara pribadi di Amazon Elastic Kubernetes Service (Amazon EKS) di belakang Network Load Balancer, dan mengakses aplikasi dengan menggunakan AWS. PrivateLink Anda kemudian dapat menggunakan jaringan pribadi untuk mengakses layanan dengan aman di Amazon Web Services (AWS) Cloud.

Cluster Amazon EKS yang menjalankan aplikasi Docker, dengan Network Load Balancer di ujung depan, dapat dikaitkan dengan titik akhir virtual private cloud (VPC) untuk akses melalui AWS. PrivateLink Layanan endpoint VPC ini kemudian dapat dibagikan dengan orang lain VPCs dengan menggunakan titik akhir VPC mereka.

Pengaturan yang dijelaskan oleh pola ini adalah cara aman untuk berbagi akses aplikasi di antara VPCs dan akun AWS. Ini tidak memerlukan konektivitas khusus atau konfigurasi perutean, karena koneksi antara akun konsumen dan penyedia berada di tulang punggung AWS global dan tidak melintasi internet publik.

Prasyarat dan batasan

Prasyarat

- Docker, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- Aplikasi yang berjalan di Docker.
- Akun AWS yang aktif.
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- Cluster Amazon EKS yang ada dengan subnet pribadi yang diberi tag dan dikonfigurasi untuk meng-host aplikasi. Untuk informasi selengkapnya, lihat Penandaan subnet di dokumentasi Amazon EKS.
- Kubectl, diinstal dan dikonfigurasi untuk mengakses sumber daya di klaster Amazon EKS Anda. Untuk informasi selengkapnya, lihat Menginstal kubectl di dokumentasi Amazon EKS.

Arsitektur

Tumpukan teknologi

- Amazon EKS
- AWS PrivateLink
- Penyeimbang Beban Jaringan

Otomatisasi dan skala

- Manifes Kubernetes dapat dilacak dan dikelola pada repositori berbasis GIT, dan digunakan dengan menggunakan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) di AWS. CodePipeline
- Anda dapat menggunakan AWS CloudFormation untuk membuat pola ini dengan menggunakan infrastruktur sebagai kode (IAc).

Alat

- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat sumber terbuka yang memungkinkan Anda berinteraksi dengan layanan AWS menggunakan perintah di shell baris perintah Anda.
- Elastic Load Balancing — Elastic Load Balancing mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target, seperti instans Amazon Elastic Compute Cloud (EC2Amazon), container, dan alamat IP, dalam satu atau beberapa Availability Zone.

- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) adalah layanan terkelola yang dapat Anda gunakan untuk menjalankan Kubernetes di AWS tanpa perlu menginstal, mengoperasikan, dan memelihara control plane atau node Kubernetes Anda sendiri.
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan.
- Kubectl — Kubectl adalah utilitas baris perintah untuk menjalankan perintah terhadap klaster Kubernetes.

Epik

Menerapkan penerapan Kubernetes dan file manifes layanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file manifes penerapan Kubernetes.	Buat file manifes penyebaran dengan memodifikasi file contoh berikut sesuai dengan kebutuhan Anda.
	
	`apiVersion: apps/v1`
	`kind: Deployment`
	`metadata:`
	`  name: sample-app`
	`spec:`
	`  replicas: 3`
	`  selector:`
	`  matchLabels:`
	`  app: nginx`
	`  template:`
	`  metadata:`
	`  labels:`
	`  app: nginx`
	`  spec:`
	`  containers:`
	`  - name: nginx`
	`  image:`
	` public.ecr.aws/z9d2n7e1/nginx:1.19.5`
	`  ports:`
	`  - name: http`
	`  container Port: 80`
	
	Note
	Ini adalah file konfigurasi sampel NGINX yang digunakan dengan
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggunakan image NGINX Docker. Untuk informasi selengkap nya, lihat Cara menggunakan image resmi NGINX Docker dalam dokumentasi Docker.	
Menerapkan file manifes penerapan Kubernetes.	Jalankan perintah berikut untuk menerapkan file manifes penerapan ke cluster Amazon EKS Anda: `kubectl apply -f <your_deployment_file_name>`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file manifes layanan Kubernetes.	Buat file manifes layanan dengan memodifikasi file contoh berikut sesuai dengan kebutuhan Anda.
	
	`apiVersion: v1`
	`kind: Service`
	`metadata:`
	`  name: sample-service`
	`  annotations:`
	`  service.beta.kubernetes.io/aws-load-balancer-type: nlb`
	`  service.beta.kubernetes.io/aws-load-balancer-internal: "true"`
	`spec:`
	`  ports:`
	`  - port: 80`
	`  targetPort: 80`
	`  protocol: TCP`
	`  type: LoadBalancer`
	`  selector:`
	`  app: nginx`
	
	⚠ Important
	Pastikan Anda menyertakan yang berikut ini `annotations` untuk menentukan Network Load Balancer internal:
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`service.beta.kubernetes.io/aws-load-balancer-type: nlb` `service.beta.kubernetes.io/aws-load-balancer-internal: "true"`	
Menerapkan file manifes layanan Kubernetes.	Jalankan perintah berikut untuk menerapkan file manifes layanan ke cluster Amazon EKS Anda: `kubectl apply -f <your_service_file_name>`	DevOps insinyur

Untuk membuat titik akhir

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rekam nama Network Load Balancer.	Jalankan perintah berikut untuk mengambil nama Network Load Balancer: `kubectl get svc sample-service -o wide` Rekam nama Network Load Balancer, yang diperlukan untuk membuat titik PrivateLink akhir AWS.	DevOps insinyur
Buat PrivateLink titik akhir AWS.	Masuk ke AWS Management Console, buka konsol Amazon	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	VPC, lalu buat titik akhir PrivateLink AWS. Kaitkan titik akhir ini dengan Network Load Balancer, ini membuat aplikasi tersedia secara pribadi bagi pelanggan. Untuk informasi selengkapnya, lihat layanan titik akhir VPC (AWS PrivateLink) di dokumentasi Amazon VPC. **Important** Jika akun konsumen memerlukan akses ke aplikasi, ID akun AWS akun konsumen harus ditambahkan ke daftar prinsipal yang diizinkan untuk konfigurasi titik akhir PrivateLink AWS. Untuk informasi selengkapnya, lihat Menambahkan dan menghapus izin untuk layanan endpoint Anda di dokumentasi Amazon VPC.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC endpoint	Di konsol Amazon VPC, pilih Endpoint Services, lalu pilih Create Endpoint Service. Buat titik akhir VPC untuk titik akhir AWS. PrivateLink Nama domain (FQDN) yang sepenuhnya memenuhi syarat dari titik akhir VPC menunjuk ke FQDN untuk titik akhir AWS. PrivateLink Ini menciptakan sebuah elastic network interface ke layanan endpoint VPC yang dapat diakses oleh endpoint DNS.	Administrator awan

Sumber daya terkait

- Menggunakan gambar resmi NGINX Docker
- Penyeimbangan beban jaringan di Amazon EKS
- Membuat layanan titik akhir VPC (AWS) PrivateLink
- Menambahkan dan menghapus izin untuk layanan endpoint Anda

Mengotomatiskan pencadangan untuk Amazon RDS untuk instans DB Amazon RDS for PostgreSQL dengan menggunakan AWS Batch

Kirankumar Chandrashekar, Amazon Web Services

Ringkasan

Mencadangkan database PostgreSQL Anda adalah tugas penting dan biasanya dapat diselesaikan dengan utilitas pg_dump, yang menggunakan perintah COPY secara default untuk membuat skema dan dump data dari database PostgreSQL. Namun, proses ini dapat menjadi berulang jika Anda memerlukan backup reguler untuk beberapa database PostgreSQL. Jika database PostgreSQL Anda di-host di cloud, Anda juga dapat memanfaatkan fitur pencadangan otomatis yang disediakan oleh Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL juga. Pola ini menjelaskan cara mengotomatiskan backup reguler untuk Amazon RDS untuk instans DB Amazon RDS for PostgreSQL menggunakan utilitas pg_dump.

Catatan: Instruksi mengasumsikan bahwa Anda menggunakan Amazon RDS. Namun, Anda juga dapat menggunakan pendekatan ini untuk database PostgreSQL yang di-host di luar Amazon RDS. Untuk mengambil cadangan, fungsi AWS Lambda harus dapat mengakses database Anda.

CloudWatch Acara Amazon Events berbasis waktu memulai fungsi Lambda yang mencari tag cadangan tertentu yang diterapkan ke metadata instans PostgreSQL DB di Amazon RDS. Jika instans PostgreSQL DB memiliki tag DBDump BKP:Automated = Active dan tag cadangan lain yang diperlukan, fungsi Lambda mengirimkan tugas individual untuk setiap backup database ke AWS Batch.

AWS Batch memproses pekerjaan ini dan mengunggah data cadangan ke bucket Amazon Simple Storage Service (Amazon S3). Pola ini menggunakan Dockerfile dan file entrypoint.sh untuk membuat image container Docker yang digunakan untuk membuat backup dalam pekerjaan AWS Batch. Setelah proses pencadangan selesai, AWS Batch mencatat detail pencadangan ke tabel inventaris di Amazon DynamoDB. Sebagai perlindungan tambahan, CloudWatch peristiwa Acara memulai notifikasi Amazon Simple Notification Service (Amazon SNS) jika pekerjaan gagal di AWS Batch.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.

- Lingkungan komputasi terkelola atau tidak terkelola yang ada. Untuk informasi selengkapnya, lihat Lingkungan komputasi terkelola dan tidak terkelola dalam dokumentasi AWS Batch.
- AWS Command Line Interface (CLI) versi 2 image Docker, diinstal dan dikonfigurasi.
- Amazon RDS yang ada untuk instans DB Amazon RDS for PostgreSQL.
- Bucket S3 yang ada.
- Docker, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- Keakraban dengan pengkodean di Lambda.

Arsitektur

Tumpukan teknologi

- CloudWatch Acara Amazon
- Amazon DynamoDB
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon RDS
- Amazon SNS
- Amazon S3
- AWS Batch
- AWS Key Management Service (AWS KMS)
- AWS Lambda
- AWS Secrets Manager
- Docker

Alat

- CloudWatch Acara Amazon — CloudWatch Acara menghadirkan aliran peristiwa sistem yang mendekati waktu nyata yang menjelaskan perubahan sumber daya AWS.
- Amazon DynamoDB - DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi dengan skalabilitas yang mulus.
- Amazon ECR — Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri image container AWS terkelola yang aman, terukur, dan andal.

- Amazon RDS - Amazon Relational Database Service (Amazon RDS) adalah layanan web yang memudahkan pengaturan, pengoperasian, dan skala database relasional di AWS Cloud.
- Amazon SNS - Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet.
- AWS Batch — AWS Batch membantu Anda menjalankan beban kerja komputasi batch di AWS Cloud.
- AWS KMS — AWS Key Management Service (AWS KMS) adalah layanan terkelola yang memudahkan Anda membuat dan mengontrol kunci enkripsi yang digunakan untuk mengenkripsi data Anda.
- AWS Lambda — Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa menyediakan atau mengelola server.
- AWS Secrets Manager — Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Docker — Docker membantu pengembang dengan mudah mengemas, mengirim, dan menjalankan aplikasi apa pun sebagai wadah yang ringan, portabel, dan mandiri.

Instans PostgreSQL DB Anda di Amazon RDS harus memiliki tag yang diterapkan ke metadatanya. Fungsi Lambda mencari tag untuk mengidentifikasi instance DB yang harus dicadangkan, dan tag berikut biasanya digunakan.

Tag	Deskripsi
BKP:Otomatis = Aktif DBDump	Mengidentifikasi instans Amazon RDS DB sebagai kandidat untuk backup.
bkp: = AutomatedBackupSecret <secret_name>	Mengidentifikasi rahasia Secrets Manager yang berisi kredensyal login Amazon RDS.
BKP: S3Bucket otomatis DBDump = <s3_bucket_name>	Mengidentifikasi bucket S3 untuk mengirim cadangan ke.
DBDumpBKP: Frekuensi Otomatis	Identifikasi frekuensi dan waktu kapan database harus dicadangkan.

DBDumpBKP: Waktu Otomatis

bkp:pgdumpcommand = <pgdump_command>	Mengidentifikasi database yang backup perlu diambil.

Epik

Buat tabel inventaris di DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel di DynamoDB.	Masuk ke AWS Management Console, buka konsol Amazon DynamoDB, dan buat tabel. Untuk bantuan tentang ini dan cerita lainnya, lihat bagian Sumber daya terkait.	Administrator awan, Administrator basis data
Konfirmasikan bahwa tabel telah dibuat.	Jalankan perintah `aws dynamodb describe-table --table-name <table-name> | grep TableStatus` . Jika tabel ada, perintah akan mengembalikan `"TableStatus": "ACTIVE",` hasilnya.	Administrator awan, Administrator basis data

Membuat topik SNS untuk peristiwa pekerjaan yang gagal di AWS Batch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat sebuah topik SNS.	Buka konsol Amazon SNS, pilih Topik, dan buat topik SNS dengan namanya. `JobFailedAlert` Berlangganan alamat	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	email aktif ke topik tersebut, dan periksa kotak masuk email Anda untuk mengonfir masi email langganan SNS dari Pemberitahuan AWS.	
Buat aturan peristiwa pekerjaan yang gagal untuk AWS Batch.	Buka CloudWatch konsol Amazon, pilih Acara, lalu pilih Buat aturan. Pilih Tampilkan opsi lanjutan, dan pilih Edit. Untuk Membangun pola yang memilih peristiwa untuk diproses oleh target Anda, ganti teks yang ada dengan kode “Peristiwa pekerjaan gagal” dari bagian Informasi tambahan. Kode ini mendefini sikan aturan CloudWatch Peristiwa yang dimulai saat AWS Batch memiliki `Failed` peristiwa.	Administrator awan
Tambahkan target aturan acara.	Di Target, pilih Tambahkan target, dan pilih topik `JobFailedAlert` SNS. Konfigurasikan detail yang tersisa dan buat aturan Cloudwatch Events.	Administrator awan

Buat image Docker dan dorong ke repositori Amazon ECR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR.	Buka konsol Amazon ECR dan pilih Wilayah AWS tempat	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda ingin membuat repositor i. Pilih Repositori, lalu pilih Buat repositori. Konfigura sikan repositori sesuai dengan kebutuhan Anda.	
Tulis Dockerfile.	Masuk ke Docker dan gunakan “Sample Dockerfil e” dan “Contoh file entrypoin t.sh” dari bagian Informasi tambahan untuk membuat Dockerfile.	DevOps insinyur
Buat image Docker dan dorong ke repositori Amazon ECR.	Bangun Dockerfile menjadi image Docker dan dorong ke repositori Amazon ECR. Untuk bantuan dengan cerita ini, lihat bagian Sumber daya terkait.	DevOps insinyur

Membuat komponen AWS Batch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat definisi pekerjaan AWS Batch.	Buka konsol AWS Batch dan buat definisi pekerjaan yang menyertakan Uniform Resource Identifier (URI) repositori Amazon ECR sebagai properti. `Image`	Administrator awan
Konfigurasikan antrian pekerjaan AWS Batch.	Di konsol AWS Batch, pilih antrian Job, lalu pilih Buat antrean. Buat antrean pekerjaan yang akan menyimpan pekerjaan hingga	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS Batch menjalank annya pada sumber daya dalam lingkungan komputasi Anda. Penting: Pastikan Anda menulis logika AWS Batch untuk merekam detail pencadangan ke tabel inventaris DynamoDB.	

Membuat dan menjadwalkan fungsi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda untuk mencari tag.	Buat fungsi Lambda yang mencari tag pada instans PostgreSQL DB Anda dan mengidentifikasi kandidat cadangan. Pastikan fungsi Lambda Anda dapat mengidentifikasi `bkp:AutomatedDBDump = Active` tag dan semua tag lain yang diperlukan. Penting: Fungsi Lambda juga harus dapat menambahkan pekerjaan ke antrean pekerjaan AWS Batch.	DevOps insinyur
Buat CloudWatch acara Acara berbasis waktu.	Buka CloudWatch konsol Amazon dan buat CloudWatch acara Acara yang menggunakan ekspresi cron untuk menjalankan fungsi Lambda Anda pada jadwal reguler. Penting: Semua acara	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	terjadwal menggunakan zona waktu UTC.	

Uji otomatisasi cadangan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kunci Amazon KMS.	Buka konsol Amazon KMS dan buat kunci KMS yang dapat digunakan untuk mengenkripsi kredensyal Amazon RDS yang disimpan di AWS Secrets Manager.	Administrator awan
Buat rahasia AWS Secrets Manager.	Buka konsol AWS Secrets Manager dan simpan kredensyal database Amazon RDS for PostgreSQL Anda sebagai rahasia.	Administrator awan
Tambahkan tag yang diperlukan ke instance PostgreSQL DB.	**Important** Buka konsol Amazon RDS dan tambahkan tag ke instans PostgreSQL DB yang ingin Anda cadangkan secara otomatis. Anda dapat menggunak an tag dari tabel di bagian Alat. Jika Anda memerlukan backup dari beberapa database PostgreSQ L dalam instance	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon RDS yang sama, maka gunakan sebagai nilai untuk tag. `-d test:-d test1 bkp:pgdum pcommand testdan test1` merupakan nama database. Pastikan tidak ada spasi setelah usus besar (:).	
Verifikasi otomatisasi cadangan.	Untuk memverifikasi otomatisasi cadangan, Anda dapat menjalankan fungsi Lambda atau menunggu jadwal pencadangan dimulai. Setelah proses pencadangan selesai, periksa apakah tabel inventaris DynamoDB memiliki entri cadangan yang valid untuk instans PostgreSQL DB Anda. Jika cocok, maka proses otomatisasi cadangan berhasil.	Administrator awan

Sumber daya terkait

Buat tabel inventaris di DynamoDB

- Buat tabel Amazon DynamoDB

Membuat topik SNS untuk peristiwa pekerjaan yang gagal di AWS Batch

- Buat topik Amazon SNS
- Kirim peringatan SNS untuk peristiwa pekerjaan yang gagal di AWS Batch

Buat image Docker dan dorong ke repositori Amazon ECR

- Buat repositori Amazon ECR
- Tulis Dockerfile, buat gambar Docker, dan dorong ke Amazon ECR

Membuat komponen AWS Batch

- Membuat definisi pekerjaan AWS Batch
- Konfigurasikan lingkungan komputasi dan antrean pekerjaan AWS Batch
- Buat antrean pekerjaan di AWS Batch

Buat fungsi Lambda

- Buat fungsi Lambda dan tulis kode
- Gunakan Lambda dengan DynamoDB

Buat CloudWatch acara Acara

- Buat CloudWatch acara Acara berbasis waktu
- Gunakan ekspresi cron di Acara Cloudwatch

Uji otomatisasi cadangan

- Buat kunci Amazon KMS
- Buat rahasia Secrets Manager
- Tambahkan tag ke instans Amazon RDS

Informasi tambahan

Acara pekerjaan yang gagal:

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

Contoh Dockerfile:

```
FROM alpine:latest
RUN apk --update add py-pip postgresql-client jq bash && \
pip install awscli && \
rm -rf /var/cache/apk/*
ADD entrypoint.sh /usr/bin/
RUN chmod +x /usr/bin/entrypoint.sh
ENTRYPOINT ["entrypoint.sh"]
```

Contoh file entrypoint.sh:

```
 #!/bin/bash
set -e
DATETIME=`date +"%Y-%m-%d_%H_%M"`
FILENAME=RDS_PostGres_dump_${RDS_INSTANCE_NAME}
FILE=${FILENAME}_${DATETIME}
```

```
aws configure --profile new-profile set role_arn arn:aws:iam::${TargetAccountId}:role/
${TargetAccountRoleName}
aws configure --profile new-profile set credential_source EcsContainer

echo "Central Account access provider IAM role is: "
aws sts get-caller-identity

echo "Target Customer Account access provider IAM role is: "
aws sts get-caller-identity --profile new-profile

securestring=$(aws secretsmanager get-secret-value --secret-id $SECRETID --output json
 --query 'SecretString' --region=$REGION --profile new-profile)

if [[ ${securestring} ]]; then
    echo "successfully accessed secrets manager and got the credentials"
    export PGPASSWORD=$(echo $securestring | jq --raw-output | jq -r '.DB_PASSWORD')
    PGSQL_USER=$(echo $securestring | jq --raw-output | jq -r '.DB_USERNAME')
    echo "Executing pg_dump for the PostGres endpoint ${PGSQL_HOST}"
    # pg_dump -h $PGSQL_HOST -U $PGSQL_USER -n dms_sample | gzip -9 -c  | aws s3 cp -
 --region=$REGION  --profile new-profile s3://$BUCKET/$FILE
    # in="-n public:-n private"
    IFS=':' list=($EXECUTE_COMMAND);
    for command in "${list[@]}";
      do
        echo $command;
        pg_dump -h $PGSQL_HOST -U $PGSQL_USER ${command} | gzip -9 -c  | aws s3 cp - --
region=$REGION --profile new-profile s3://${BUCKET}/${FILE}-${command}".sql.gz"
        echo $?;
        if  [[ $? -ne 0 ]]; then
            echo "Error occurred in database backup process. Exiting now....."
            exit 1
        else
            echo "Postgresql dump was successfully taken for the RDS endpoint
 ${PGSQL_HOST} and is uploaded to the following S3 location s3://${BUCKET}/${FILE}-
${command}.sql.gz"
            #write the details into the inventory table in central account
            echo "Writing to DynamoDB inventory table"
            aws dynamodb put-item --table-name ${RDS_POSTGRES_DUMP_INVENTORY_TABLE} --
region=$REGION --item '{ "accountId": { "S": "'"${TargetAccountId}"'" }, "dumpFileUrl":
 {"S": "'"s3://${BUCKET}/${FILE}-${command}.sql.gz"'" }, "DumpAvailableTime": {"S":
 "'"`date +"%Y-%m-%d::%H::%M::%S"` UTC"'"}}'
            echo $?
            if  [[ $? -ne 0 ]]; then
```

```
                echo "Error occurred while putting item to DynamoDb Inventory Table.
 Exiting now....."
                exit 1
            else
                echo "Successfully written to DynamoDb Inventory Table
 ${RDS_POSTGRES_DUMP_INVENTORY_TABLE}"
            fi
        fi
      done;
else
    echo "Something went wrong {$?}"
    exit 1
fi

exec "$@"
```

Otomatiskan penerapan Node Termination Handler di Amazon EKS dengan menggunakan pipeline CI/CD

Sandip Gangapadhyay, Sandeep Gawande, Viyoma Sachdeva, Pragtideep Singh, dan John Vargas, Amazon Web Services

Ringkasan

Pemberitahuan: AWS CodeCommit tidak lagi tersedia untuk pelanggan baru. Pelanggan AWS yang sudah ada CodeCommit dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

Di Amazon Web Services (AWS) Cloud, Anda dapat menggunakan AWS Node Termination Handler, sebuah project open-source, untuk menangani shutdown instans Amazon Elastic Compute Cloud (Amazon EC2) di Kubernetes dengan anggun. AWS Node Termination Handler membantu memastikan bahwa control plane Kubernetes merespons dengan tepat peristiwa yang dapat menyebabkan instans Anda menjadi tidak tersedia. EC2 Peristiwa tersebut meliputi:

- EC2 contoh pemeliharaan terjadwal
- Gangguan Instans EC2 Spot Amazon
- Skala grup Auto Scaling di
- Penyeimbangan ulang grup Auto Scaling di seluruh Availability Zone
- EC2 penghentian instans melalui API atau AWS Management Console

Jika suatu peristiwa tidak ditangani, kode aplikasi Anda mungkin tidak berhenti dengan baik. Mungkin juga perlu waktu lebih lama untuk memulihkan ketersediaan penuh, atau mungkin secara tidak sengaja menjadwalkan pekerjaan ke node yang turun. NTH dapat beroperasi dalam dua mode yang berbeda: Instance Metadata Service (IMDS) atau Queue Processor. `aws-node-termination-handler` Untuk informasi lebih lanjut tentang dua mode, lihat file Readme.

Pola ini menggunakan AWS CodeCommit, dan mengotomatiskan penyebaran NTH dengan menggunakan Prosesor Antrian melalui pipeline continuous integration and continuous delivery (CI/CD).

> **Note**
>
> Jika Anda menggunakan grup simpul terkelola EKS, Anda tidak memerlukan`aws-node-termination-handler`.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Browser web yang didukung untuk digunakan dengan AWS Management Console. Lihat daftar browser yang didukung.
- AWS Cloud Development Kit (AWS CDK) diinstal.
- `kubectl`, alat baris perintah Kubernetes, diinstal.
- `eksctl`AWS Command Line Interface (AWS CLI), AWS Command Line Interface (AWS CLI) untuk Amazon Elastic Kubernetes Service (Amazon EKS), diinstal.
- Kluster EKS yang berjalan dengan versi 1.20 atau yang lebih baru.
- Grup node yang dikelola sendiri yang melekat pada cluster EKS. Untuk membuat klaster Amazon EKS dengan grup node yang dikelola sendiri, jalankan perintah berikut.

```
eksctl create cluster --managed=false --region <region> --name <cluster_name>
```

  Untuk informasi lebih lanjut tentang`eksctl`, lihat dokumentasi eksctl.
- Penyedia AWS Identity and Access Management (IAM) OpenID Connect (OIDC) untuk klaster Anda. Untuk informasi selengkapnya, lihat Membuat penyedia IAM OIDC untuk klaster Anda.

Batasan

- Anda harus menggunakan Wilayah AWS yang mendukung layanan Amazon EKS.

Versi produk

- Kubernetes versi 1.20 atau yang lebih baru
- `eksctl`versi 0.107.0 atau yang lebih baru
- AWS CDK versi 2.27.0 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- Awan pribadi virtual (VPC)

- Kluster EKS
- Amazon Simple Queue Service (Amazon SQS)
- IAM
- Kubernetes

Arsitektur target

Diagram berikut menunjukkan tampilan tingkat tinggi dari end-to-end langkah-langkah ketika penghentian node dimulai.

Alur kerja yang ditunjukkan dalam diagram terdiri dari langkah-langkah tingkat tinggi berikut:

1. Peristiwa penghentian EC2 instance penskalaan otomatis dikirim ke antrian SQS.
2. NTH Pod memonitor pesan baru dalam antrean SQS.
3. NTH Pod menerima pesan baru dan melakukan hal berikut:
   - Cordons node sehingga pod baru tidak berjalan pada node.
   - Menguras node, sehingga pod yang ada dievakuasi
   - Mengirim sinyal hook siklus hidup ke grup Auto Scaling sehingga node dapat dihentikan.

Otomatisasi dan skala

- Kode dikelola dan diterapkan oleh AWS CDK, didukung oleh tumpukan CloudFormation bersarang AWS.
- Pesawat kontrol Amazon EKS berjalan di beberapa Availability Zone untuk memastikan ketersediaan yang tinggi.
- Untuk penskalaan otomatis, Amazon EKS mendukung Kubernetes Cluster Autoscaler dan Karpenter.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.

- AWS CodeBuild adalah layanan build terkelola penuh yang membantu Anda mengompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodePipeline membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- EC2 Auto Scaling Amazon membantu Anda menjaga ketersediaan aplikasi dan memungkinkan Anda menambahkan atau menghapus EC2 instans Amazon secara otomatis sesuai dengan kondisi yang Anda tentukan.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.

Alat lainnya

- kubectl adalah alat baris perintah Kubernetes untuk menjalankan perintah terhadap klaster Kubernetes. Anda dapat menggunakan kubectl untuk menyebarkan aplikasi, memeriksa dan mengelola sumber daya klaster, dan melihat log.

Kode

Kode untuk pola ini tersedia di deploy-nth-to-eksrepo di GitHub .com. Kode repo berisi file dan folder berikut.

- `nth folder`— Bagan Helm, file nilai, dan skrip untuk memindai dan menerapkan CloudFormation template AWS untuk Node Termination Handler.
- `config/config.json`— File parameter konfigurasi untuk aplikasi. File ini berisi semua parameter yang diperlukan untuk CDK untuk digunakan.
- `cdk`— Kode sumber AWS CDK.
- `setup.sh`— Skrip yang digunakan untuk menyebarkan aplikasi AWS CDK untuk membuat CI/CD pipeline yang diperlukan dan sumber daya lain yang diperlukan.
- `uninstall.sh`— Script yang digunakan untuk membersihkan sumber daya.

Untuk menggunakan kode contoh, ikuti instruksi di bagian Epik.

Praktik terbaik

Untuk praktik terbaik saat mengotomatiskan AWS Node Termination Handler, lihat berikut ini:

- Panduan Praktik Terbaik EKS
- Handler Terminasi Node - Konfigurasi

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repo.	Untuk mengkloning repo dengan menggunakan SSH (Secure Shell), jalankan perintah berikut.
	`git clone git@github.com:aws-samples/deploy-nth-to-eks.git`
	Untuk mengkloning repo dengan menggunakan HTTPS, jalankan perintah berikut.
	`git clone https://github.com/aws-samples/deploy-nth-to-eks.git`
	Mengkloning repo membuat folder bernama. `deploy-nth-to-eks`
	Ubah ke direktori itu.
		Pengembang aplikasi, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`cd deploy-nth-to-eks`	
Atur file kubeconfig.	Tetapkan kredensi AWS Anda di terminal dan konfirmasikan bahwa Anda memiliki hak untuk mengambil peran klaster. Anda dapat menggunakan kode contoh berikut. `aws eks update-kubeconfig --name <Cluster_Name> --region <region>--role-arn <Role_ARN>`	AWS DevOps, DevOps insinyur, Pengembang aplikasi

Menyebarkan pipa CI/CD

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan parameternya.	Dalam `config/config.json` file, atur parameter yang diperlukan berikut. • `pipelineName` : Nama CI/CD pipeline yang akan dibuat oleh AWS CDK (misalnya,`deploy-nth-to-eks-pipeline`). AWS CodePipeline akan membuat pipeline yang memiliki nama ini.	Pengembang aplikasi, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `repositoryName` : CodeCommit Repo AWS yang akan dibuat (misalnya ,`deploy-nth-to-eks-repo`). AWS CDK akan membuat repo ini dan mengaturnya sebagai sumber untuk pipeline. CI/CD **Note** Solusi ini akan membuat CodeCommit repo ini dan cabang (disediakan dalam parameter cabang berikut). • `branch`: Nama cabang di repo (misalnya,`main`). Komit untuk cabang ini akan memulai CI/CD pipeline. • `cfn_scan_script` : Jalur skrip yang akan digunakan untuk memindai CloudFormation template AWS untuk NTH (`scan.sh`). Skrip ini ada di `nth` folder yang akan menjadi bagian dari CodeCommit repo AWS. • `cfn_deploy_scrip t`: Jalur skrip yang akan digunakan untuk menyebark	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an CloudFormation template AWS untuk NTH ()`installApp.sh` . • `stackName` : Nama CloudFormation tumpukan yang akan digunakan. • `eksClusterName` : Nama cluster EKS yang ada. • `eksClusterRole` : Peran IAM yang akan digunakan untuk mengakses kluster EKS untuk semua panggilan API Kubernetes (misalnya,). `clusteradmin` Biasanya, peran ini ditambahkan `aws-authConfigMap` . • `create_cluster_role` : Untuk membuat peran `eksClusterRole` IAM, masukkan ya. Jika Anda ingin memberikan peran cluster yang ada dalam `eksClusterRole` parameter, masukkan no. • `create_iam_oidc_provider` : Untuk membuat penyedia IAM OIDC untuk cluster Anda, masukkan ya. Jika penyedia IAM OIDC sudah ada, masukkan no. Untuk informasi selengkapnya, lihat Membuat penyedia IAM OIDC untuk klaster Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `AsgGroupName` : Daftar nama grup Auto Scaling yang dipisahkan koma yang merupakan bagian dari kluster EKS (misalnya,). `ASG_Group_1,ASG_Group_2` • `region`: Nama Wilayah AWS tempat cluster berada (misalnya,`us-east-2`). • `install_cdk` : Jika AWS CDK saat ini tidak diinstal pada mesin, masukkan ya. Jalankan `cdk --version` perintah untuk memeriksa apakah versi AWS CDK yang diinstal adalah 2.27.0 atau yang lebih baru. Dalam hal ini, masukkan no. Jika Anda memasukkan ya, skrip setup.sh akan menjalankan `sudo npm install -g cdk@2.27.0` perintah untuk menginstal AWS CDK pada mesin. Skrip memerlukan izin sudo, jadi berikan kata sandi akun saat diminta.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CI/CD pipeline untuk menyebarkan NTH.	Jalankan skrip setup.sh. `./setup.sh` Skrip akan menerapkan aplikasi AWS CDK yang akan membuat CodeCommit repo dengan kode contoh, pipeline, dan CodeBuild proyek berdasarkan parameter input pengguna dalam file `config/config.json`. Skrip ini akan meminta kata sandi saat menginstal paket npm dengan perintah sudo.	Pengembang aplikasi, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau CI/CD pipa.	Buka AWS Management Console, dan tinjau sumber daya berikut yang dibuat di tumpukan. • CodeCommit repo dengan isi folder `nth` • CodeBuild Proyek AWS`cfn-scan`, yang akan memindai CloudFormation template untuk kerentanan. • CodeBuild project`Nth-Deploy` , yang akan menerapkan CloudFormation template AWS dan bagan NTH Helm yang sesuai melalui pipeline AWS. CodePipeline • CodePipeline Pipa untuk menyebarkan NTH. Setelah pipeline berjalan dengan sukses, rilis Helm `aws-node-termination-handler` diinstal di cluster EKS. Juga, sebuah Pod bernama `aws-node-termination-handler` sedang berjalan di `kube-system` namespace di cluster.	Pengembang aplikasi, AWS DevOps, DevOps insinyur

Uji penyebaran NTH

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Simulasikan acara skala dalam grup Auto Scaling.	Untuk mensimulasikan event scale-in penskalaan otomatis, lakukan hal berikut: 1. Di konsol AWS, buka EC2 konsol, dan pilih Grup Auto Scaling. 2. Pilih grup Auto Scaling yang memiliki nama yang sama dengan yang disediaka n**`config/config.json`** , dan pilih Edit. 3. Kurangi Kapasitas yang Diinginkan dan Minimum sebesar 1. 4. Pilih Perbarui.	
Tinjau log.	Selama peristiwa scale-in, Pod NTH akan mengikat dan menguras node pekerja yang sesuai (EC2 instance yang akan dihentikan sebagai bagian dari peristiwa scale-in). Untuk memeriksa log, gunakan kode di bagian Informasi tambahan.	Pengembang aplikasi, AWS DevOps, DevOps insinyur

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan semua sumber daya AWS.	Untuk membersihkan sumber daya yang dibuat oleh pola ini, jalankan perintah berikut. `./uninstall.sh` Ini akan membersihkan semua sumber daya yang dibuat dalam pola ini dengan menghapus CloudFormation tumpukan.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Registri npm tidak disetel dengan benar.	Selama instalasi solusi ini, skrip menginstal npm install untuk mengunduh semua paket yang diperlukan. Jika, selama instalasi, Anda melihat pesan yang mengatakan “Tidak dapat menemukan modul,” registri npm mungkin tidak disetel dengan benar. Untuk melihat pengaturan registri saat ini, jalankan perintah berikut. `npm config get registry` Untuk mengatur registri dengan`https://registry.npmjs.org/`, jalankan perintah berikut.

Isu	Solusi
	`npm config set registry https://registry.npmjs.org`
Tunda pengiriman pesan SQS.	Sebagai bagian dari pemecahan masalah Anda, jika Anda ingin menunda pengiriman pesan SQS ke NTH Pod, Anda dapat menyesuaikan parameter penundaan pengiriman SQS. Untuk informasi selengkapnya, lihat antrian penundaan Amazon SQS.

Sumber daya terkait

- Kode sumber AWS Node Termination Handler
- EC2 lokakarya
- AWS CodePipeline
- Amazon Elastic Kubernetes Service (Amazon EKS)
- AWS Cloud Development Kit
- AWS CloudFormation

Informasi tambahan

1. Temukan nama NTH Pod.

```
kubectl get pods -n kube-system |grep aws-node-termination-handler
aws-node-termination-handler-65445555-kbqc7   1/1     Running   0          26m
kubectl get pods -n kube-system |grep aws-node-termination-handler
aws-node-termination-handler-65445555-kbqc7   1/1     Running   0          26m
```

2. Periksa log. Contoh log terlihat seperti berikut ini. Ini menunjukkan bahwa node telah dikepung dan dikeringkan sebelum mengirim sinyal penyelesaian kait siklus hidup grup Auto Scaling.

```
kubectl -n kube-system logs aws-node-termination-handler-65445555-kbqc7
022/07/17 20:20:43 INF Adding new event to the event store
 event={"AutoScalingGroupName":"eksctl-my-cluster-target-nodegroup-
```

```
ng-10d99c89-NodeGroup-ZME36IGAP7O1","Description":"ASG Lifecycle Termination
 event received. Instance will be interrupted at 2022-07-17 20:20:42.702
 +0000 UTC \n","EndTime":"0001-01-01T00:00:00Z","EventID":"asg-lifecycle-
term-33383831316538382d353564362d343332362d613931352d383430666165636334333564","InProgress":fal
east-2.compute.internal","NodeProcessed":false,"Pods":null,"ProviderID":"aws:///us-
east-2c/i-0409f2a9d3085b80e","StartTime":"2022-07-17T20:20:42.702Z","State":""}
2022/07/17 20:20:44 INF Requesting instance drain event-id=asg-lifecycle-
term-33383831316538382d353564362d343332362d613931352d383430666165636334333564
 instance-id=i-0409f2a9d3085b80e kind=SQS_TERMINATE node-name=ip-192-168-75-60.us-
east-2.compute.internal provider-id=aws:///us-east-2c/i-0409f2a9d3085b80e
2022/07/17 20:20:44 INF Pods on node node_name=ip-192-168-75-60.us-
east-2.compute.internal pod_names=["aws-node-qchsw","aws-node-termination-
handler-65445555-kbqc7","kube-proxy-mz5x5"]
2022/07/17 20:20:44 INF Draining the node
2022/07/17 20:20:44 ??? WARNING: ignoring DaemonSet-managed Pods: kube-system/aws-node-
qchsw, kube-system/kube-proxy-mz5x5
2022/07/17 20:20:44 INF Node successfully cordoned and drained
 node_name=ip-192-168-75-60.us-east-2.compute.internal reason="ASG Lifecycle
 Termination event received. Instance will be interrupted at 2022-07-17 20:20:42.702
 +0000 UTC \n"
2022/07/17 20:20:44 INF Completed ASG Lifecycle Hook (NTH-K8S-TERM-HOOK) for instance
 i-0409f2a9d3085b80e
```

Secara otomatis membangun dan menyebarkan aplikasi Java ke Amazon EKS menggunakan pipeline CI/CD

MAHESH RAGHUNANDANAN, Jomcy Pappachen, dan James Radtke, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara membuat pipeline continuous integration and continuous delivery (CI/CD) yang secara otomatis membangun dan menyebarkan aplikasi Java dengan DevSecOps praktik yang direkomendasikan ke cluster Amazon Elastic Kubernetes Service (Amazon EKS) di file. AWS Cloud Pola ini menggunakan aplikasi salam yang dikembangkan dengan kerangka Spring Boot Java dan yang menggunakan Apache Maven.

Anda dapat menggunakan pendekatan pola ini untuk membangun kode untuk aplikasi Java, mengemas artefak aplikasi sebagai gambar Docker, memindai keamanan gambar, dan mengunggah gambar sebagai wadah beban kerja di Amazon EKS. Pendekatan pola ini berguna jika Anda ingin bermigrasi dari arsitektur monolitik yang digabungkan erat ke arsitektur layanan mikro. Pendekatan ini juga membantu Anda memantau dan mengelola seluruh siklus hidup aplikasi Java, yang memastikan tingkat otomatisasi yang lebih tinggi dan membantu menghindari kesalahan atau bug.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang ini, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLI dokumentasi. AWS CLI

  AWS CLI versi 2 harus dikonfigurasi dengan peran AWS Identity and Access Management (IAM) yang sama yang membuat kluster Amazon EKS, karena hanya peran itu yang diizinkan untuk menambahkan peran IAM lainnya ke. `aws-auth` `ConfigMap` Untuk informasi dan langkah-langkah untuk mengkonfigurasi AWS CLI, lihat Mengkonfigurasi pengaturan dalam AWS CLI dokumentasi.
- Peran dan izin IAM dengan akses penuh ke. AWS CloudFormation Untuk informasi selengkapnya tentang ini, lihat Mengontrol akses dengan IAM dalam CloudFormation dokumentasi.
- Cluster Amazon EKS yang ada, dengan detail nama peran IAM dan Nama Sumber Daya Amazon (ARN) peran IAM untuk node pekerja di cluster EKS.

- Kubernetes Cluster Autoscaler, diinstal dan dikonfigurasi di klaster Amazon EKS Anda. Untuk informasi selengkapnya, lihat Komputasi klaster skala dengan Karpenter dan Cluster Autoscaler di dokumentasi Amazon EKS.
- Akses ke kode di GitHub repositori.

> **Important**
>
> AWS Security Hub CSPM diaktifkan sebagai bagian dari CloudFormation template yang disertakan dalam kode untuk pola ini. Secara default, setelah Security Hub CSPM diaktifkan, ia dilengkapi dengan uji coba gratis 30 hari. Setelah persidangan, ada biaya yang terkait dengan ini Layanan AWS. Untuk informasi selengkapnya tentang harga, silakan lihat harga AWS Security Hub CSPM.

Versi produk

- Helm versi 3.4.2 atau yang lebih baru
- Apache Maven versi 3.6.3 atau yang lebih baru
- BridgeCrew Checkov versi 2.2 atau yang lebih baru
- Aqua Security Trivy versi 0.37 atau yang lebih baru

Arsitektur

Tumpukan teknologi

- AWS CodeBuild
- AWS CodeCommit
- Amazon CodeGuru
- AWS CodePipeline
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon EKS
- Amazon EventBridge
- AWS Security Hub CSPM
- Amazon Simple Notification Service (Amazon SNS)

Arsitektur target

Diagram menunjukkan alur kerja berikut:

1. Pengembang memperbarui kode aplikasi Java di cabang dasar CodeCommit repositori, yang membuat permintaan tarik (PR).
2. Segera setelah PR dikirimkan, Amazon CodeGuru Reviewer secara otomatis meninjau kode, menganalisisnya berdasarkan praktik terbaik untuk Java, dan memberikan rekomendasi kepada pengembang.
3. Setelah PR digabungkan ke cabang dasar, EventBridge acara Amazon dibuat.
4. EventBridge Acara memulai CodePipeline pipa, yang dimulai.
5. CodePipeline menjalankan tahap CodeSecurity Scan (keamanan berkelanjutan).
6. AWS CodeBuild memulai proses pemindaian keamanan di mana file Helm penerapan Dockerfile dan Kubernetes dipindai dengan menggunakan Checkov, dan kode sumber aplikasi dipindai berdasarkan perubahan kode tambahan. Pemindaian kode sumber aplikasi dilakukan oleh pembungkus CodeGuru Reviewer Command Line Interface (CLI).

   > **Note**
   >
   > Mulai 7 November 2025, Anda tidak dapat membuat asosiasi repositori baru di Amazon Reviewer. CodeGuru Untuk mempelajari tentang layanan dengan kemampuan yang mirip dengan CodeGuru Reviewer, lihat Perubahan ketersediaan Amazon CodeGuru Reviewer di dokumentasi CodeGuru Reviewer.

7. Jika tahap pemindaian keamanan berhasil, tahap Build (integrasi berkelanjutan) dimulai.
8. Pada tahap CodeBuild Build, membangun artefak, mengemas artefak ke image Docker, memindai gambar untuk kerentanan keamanan dengan menggunakan Aqua Security Trivy, dan menyimpan gambar di Amazon ECR.
9. Kerentanan yang terdeteksi dari langkah 8 diunggah ke Security Hub CSPM untuk analisis lebih lanjut oleh pengembang atau insinyur. Security Hub CSPM memberikan gambaran umum dan rekomendasi untuk memulihkan kerentanan.
10. Pemberitahuan email tentang fase berurutan dalam CodePipeline pipeline dikirim melalui Amazon SNS.
11. Setelah fase integrasi berkelanjutan selesai, CodePipeline memasuki tahap Deploy (pengiriman berkelanjutan).

12. Image Docker di-deploy ke Amazon EKS sebagai beban kerja kontainer (pod) dengan menggunakan bagan Helm.
13. Pod aplikasi dikonfigurasi dengan agen Amazon CodeGuru Profiler, yang mengirimkan data profil aplikasi (CPU, penggunaan heap, dan latensi) ke CodeGuru Profiler, yang membantu pengembang memahami perilaku aplikasi.

Alat

Layanan AWS

- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS CodeBuildadalah layanan build yang dikelola sepenuhnya yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- Amazon CodeGuru Profiler mengumpulkan data performa runtime dari aplikasi live Anda, dan memberikan rekomendasi yang dapat membantu Anda menyempurnakan performa aplikasi.
- AWS CodePipelinemembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data real-time dari berbagai sumber, termasuk AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

- AWS Security Hub CSPMmemberikan pandangan komprehensif tentang keadaan keamanan Anda AWS. Ini juga membantu Anda memeriksa AWS lingkungan Anda terhadap standar industri keamanan dan praktik terbaik.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Layanan lainnya

- Helm adalah manajer paket open-source untuk Kubernetes.
- Apache Maven adalah alat manajemen dan pemahaman proyek perangkat lunak.
- BridgeCrew Checkov adalah alat analisis kode statis untuk memindai infrastruktur sebagai file kode (IAc) untuk kesalahan konfigurasi yang dapat menyebabkan masalah keamanan atau kepatuhan.
- Aqua Security Trivy adalah pemindai komprehensif untuk kerentanan dalam gambar kontainer, sistem file, dan repositori Git, selain masalah konfigurasi.

Kode

Kode untuk pola ini tersedia di GitHub aws-codepipeline-devsecops-amazoneksrepositori.

Praktik terbaik

- Pola ini mengikuti praktik terbaik keamanan IAM untuk menerapkan prinsip hak istimewa terkecil untuk entitas IAM di semua fase solusi. Jika Anda ingin memperluas solusi dengan alat tambahan Layanan AWS atau pihak ketiga, kami sarankan Anda meninjau bagian tentang penerapan izin hak istimewa paling sedikit dalam dokumentasi IAM.
- Jika Anda memiliki beberapa aplikasi Java, kami sarankan Anda membuat CI/CD pipeline terpisah untuk setiap aplikasi.
- Jika Anda memiliki aplikasi monolit, kami sarankan Anda memecah aplikasi menjadi layanan mikro jika memungkinkan. Layanan mikro lebih fleksibel, memudahkan penerapan aplikasi sebagai wadah, dan memberikan visibilitas yang lebih baik ke dalam keseluruhan pembuatan dan penyebaran aplikasi.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Untuk mengkloning repositori, jalankan perintah berikut. `git clone https://github.com/aws-samples/aws-codepipeline-devsecops-amazoneks`	Pengembang aplikasi, DevOps insinyur
Buat bucket S3 dan unggah kodenya.	1. Masuk ke Konsol Manajemen AWS, buka konsol Amazon S3, lalu buat bucket S3 di Wilayah AWS tempat Anda berencana untuk menerapkan solusi ini. Untuk informasi selengkapnya, lihat Membuat bucket di dokumentasi Amazon S3. 2. Di bucket S3, buat folder bernama `code`. 3. Arahkan ke tempat Anda mengkloning repositori. Untuk membuat versi terkompresi dari seluruh kode dengan ekstensi. zip (`cicdstack.zip`) dan memvalidasi file.zip, jalankan perintah berikut secara berurutan.	AWS DevOps, administrator Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>cd aws-codepipeline-devsecops-amazoneks
	python -m zipfile -c cicdstack.zip *
	python -m zipfile -t cicdstack.zip</pre>
	Note
	Jika `python` perintah gagal dan menyatakan bahwa Python tidak ditemukan, gunakan `python3` sebagai gantinya.
	4. Unggah `cicdstack.zip` file ke folder kode yang sebelumnya Anda buat di bucket S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan.	1. Buka konsol CloudFormation dan pilih Buat tumpukan. 2. Di Tentukan templat, pilih Unggah file templat, unggah `cf_templates/codecommit_ecr.yaml` file, lalu pilih Berikutnya. 3. Di Tentukan detail tumpukan, masukkan nama tumpukan, lalu berikan nilai parameter input berikut: • `CodeCommitRepositoryBranchName` : Nama cabang tempat kode Anda akan berada (defaultnya adalah`main`) • `CodeCommitRepositoryName` : Nama yang CodeCommitrepository akan dibuat • `CodeCommitRepositoryS3Bucket` : Nama bucket S3 tempat Anda membuat folder kode • `CodeCommitRepositoryS3BucketObjKey` : `code/cicdstack.zip` • `ECRRepositoryName` : Nama repositori Amazon ECR untuk dibuat	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pilih Berikutnya, gunakan pengaturan default untuk opsi Konfigurasi tumpukan, lalu pilih Berikutnya. 5. Di bagian Review, verifikasi detail template dan stack, lalu pilih Create stack. Tumpukan kemudian dibuat, termasuk repositori ECR CodeCommit dan Amazon. 6. Perhatikan nama repositori ECR CodeCommit dan Amazon, yang akan diperlukan untuk mengatur pipeline Java. CI/CD	
Validasi penyebaran CloudFormation tumpukan.	1. Di bawah Tumpukan di CloudFormation konsol, verifikasi status CloudFormation tumpukan yang Anda gunakan. Status tumpukan harus CREATE COMPLETE. 2. Dari konsol, validasi bahwa repositori ECR CloudFormation dan Amazon telah disediakan dan siap.	AWS DevOps, DevOps insinyur
Hapus bucket S3.	Kosongkan dan hapus bucket S3 yang Anda buat sebelumnya. Untuk informasi selengkapnya, lihat Menghapus bucket di dokumentasi Amazon S3.	AWS DevOps, DevOps insinyur

Konfigurasikan bagan Helm

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan bagan Helm aplikasi Java Anda.	1. Di lokasi tempat Anda mengkloning GitHub repositori, navigasikan ke folder. `helm_charts/aws-proserve-java-greeting` Di folder ini, `values.dev.yaml` file berisi informasi tentang konfigurasi sumber daya Kubernetes yang dapat Anda modifikasi untuk penerapan container Anda ke Amazon EKS. Perbarui parameter repositori Docker dengan memberikan Akun AWS ID Anda,, Wilayah AWS dan nama repositori Amazon ECR.
	`image:`
	`  repository:`
	`  <account-id>.dkr.ecr.<region>.amazonaws.com/<app-ecr-repo-name>`
	2. Jenis layanan pod Java diatur ke`LoadBalancer` .
	`service:`
	`  type: LoadBalancer`
	`  port: 80`
	`  targetPort: 8080`
	`  path: /hello`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> initialDelaySecond
	s: 60
	  periodSeconds: 30</pre>
	Untuk menggunakan layanan yang berbeda (misalnya,`NodePort`), Anda dapat mengubah parameter ini. Untuk informasi selengkap nya, lihat dokumentasi Kubernetes.
	3. Anda dapat mengaktifkan Kubernetes Horizontal Pod Autoscaler dengan mengubah parameter menjadi. `autoscaling enabled: true`
	<pre>autoscaling:
	  enabled: true
	  minReplicas: 1
	  maxReplicas: 100
	  targetCPUUtilizati
	onPercentage: 80
	  # targetMem
	oryUtilizationPerc
	entage: 80</pre>
	4. Anda dapat mengaktifkan berbagai fitur untuk beban kerja Kubernetes dengan mengubah nilai dalam `values.<ENV>.yaml` file, di mana `<ENV>` lingkungan pengembangan,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	produksi, UAT, atau QA Anda.	

Validasi bagan Helm untuk kesalahan sintaks.	1. Dari terminal, verifikasi bahwa Helm v3 diinstal di workstation lokal Anda dengan menjalankan perintah berikut.
	`helm --version`
	Jika Helm v3 tidak diinstal, instal.
	2. Di terminal, arahkan ke direktori bagan Helm (`helm_charts/aws-proserve-java-greeting`), dan jalankan perintah berikut.
	`helm lint . -f values.dev.yaml`
	Ini akan memeriksa bagan Helm untuk kesalahan sintaks apa pun.
		DevOps insinyur

Siapkan CI/CD pipeline Java

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CI/CD pipa.	1. Buka konsol CloudFormation, lalu pilih Buat tumpukan.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di Tentukan templat, pilih Unggah file templat, unggah `cf_templates/build_deployment.yaml` templat, lalu pilih Berikutnya. 3. Di Tentukan detail tumpukan, tentukan nama Stack, lalu berikan nilai berikut untuk parameter input: • `CodeBranchName` : Nama cabang CodeCommit repositori tempat kode Anda berada • `EKSClusterName` : Nama cluster EKS Anda (bukan `EKSCluster` ID) • `EKSCodeBuildAppName` : Nama aplikasi Helm chart () `aws-proserve-java-greeting` • `EKSWorkerNodeRoleARN` : ARN dari peran IAM yang ditetapkan ke node pekerja Amazon EKS • `EKSWorkerNodeRoleName` : Nama peran IAM yang ditetapkan ke node pekerja Amazon EKS • `EcrDockerRepository` : Nama repositori	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon ECR tempat gambar Docker kode Anda akan disimpan • `EmailRecipient` : Alamat email tempat pemberitahuan build harus dikirim • `EnvType`: Lingkungan (misalnya, dev, test, atau prod) • `SourceRepoName` : Nama CodeCommit repositori tempat kode Anda berada 4. Pilih Berikutnya. Gunakan pengaturan default di Configure stack options, lalu pilih Next. 5. Di bagian Review, verifikasi CloudFormation template dan detail stack, lalu pilih Next. 6. Pilih Buat tumpukan. 7. Selama penyebaran CloudFormation tumpukan, pemilik alamat email yang Anda berikan dalam parameter akan menerima pesan untuk berlangganan topik SNS. Untuk berlangganan Amazon SNS, pemilik harus memilih tautan dalam pesan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. Setelah tumpukan dibuat, buka tab Output dari tumpukan, lalu rekam nilai ARN untuk `EksCodeBuildkubeRoleARN` kunci keluaran. Nilai IAM ARN ini akan diperlukan nanti saat Anda memberikan izin untuk peran IAM untuk menerapkan beban kerja CodeBuild di klaster Amazon EKS.	

Aktifkan integrasi antara Security Hub CSPM dan Aqua Security

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan integrasi Aqua Security.	Langkah ini diperlukan untuk mengunggah temuan kerentanan gambar Docker yang dilaporkan oleh Trivy ke Security Hub CSPM. Karena CloudFormation tidak mendukung integrasi CSPM Security Hub, proses ini harus dilakukan secara manual. 1. Buka AWS Security Hub CSPM konsol, dan arahkan ke Integrasi. 2. Cari Aqua Security, dan pilih Aqua Security: Aqua Security. 3. Pilih Terima temuan.	Administrator AWS, DevOps insinyur

Konfigurasi CodeBuild untuk menjalankan perintah Helm atau kubectl

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Izinkan CodeBuild untuk menjalankan perintah Helm atau kubectl di cluster Amazon EKS.	CodeBuild Agar dapat diautentikasi untuk menggunakan Helm atau `kubectl` perintah dengan kluster Amazon EKS, Anda harus menambahkan peran IAM ke. `aws-auth ConfigMap` Dalam hal ini, tambahkan ARN dari peran IAM`EksCodeBuildkubeRoleARN` , yang merupakan peran IAM yang dibuat untuk CodeBuild layanan untuk mengakses kluster Amazon EKS dan menyebarkan beban kerja di dalamnya. Ini adalah kegiatan satu kali. ⚠ Important Prosedur berikut harus diselesaikan sebelum tahap persetuju an penyebaran di CodePipeline. 1. Buka skrip `cf_templates/kube_aws_auth_configmap_patch.sh` shell di lingkungan Amazon Linux atau macOS Anda.	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Otentikasi ke cluster Amazon EKS dengan menjalankan perintah berikut.
	
	`aws eks --region <aws-region> update-kubeconfig --name <eks-cluster-name>`
	
	3. Jalankan skrip shell dengan menggunakan perintah berikut, ganti `<rolearn-eks-codebuild-kubectl>` dengan nilai ARN `EksCodeBuildkubeRoleARN` yang Anda rekam sebelumnya.
	
	`bash cf_templates/kube_aws_auth_configmap_patch.sh <rolearn-eks-codebuild-kubectl>`
	
	`aws_authConfigMap` Ini dikonfigurasi, dan akses diberikan.	

Validasi pipa CI/CD

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi bahwa CI/CD pipeline dimulai secara otomatis.	1. Tahap CodeSecurity Pemindaian dalam pipeline biasanya akan gagal jika Checkov mendeteksi kerentanan di bagan Dockerfile atau Helm. Namun, tujuan dari contoh ini adalah untuk menetapkan proses mengidentifikasi potensi kerentanan keamanan daripada memperbaikinya melalui CI/CD pipeline, biasanya sebuah DevSecOps proses. Dalam file`buildspec/buildspec_secscan.yaml` , `checkov` perintah menggunakan `--soft-fail` bendera untuk menghindari kegagalan pipeline.
	`- echo -e "\n Running Dockerfile Scan"`
	`- checkov -f code/app/Dockerfile --framework dockerfile --soft-fail --summary-position bottom`
	`- echo -e "\n Running Scan of Helm Chart files"`
	`- cp -pv helm_charts/$EKS_CODEBUILD_APP_NAME/values.de`
		DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	v.yaml helm_char ts/$EKS_CODEBUILD_ APP_NAME/values.ya ml - checkov -d helm_char ts/$EKS_CODEBUILD_ APP_NAME --framewo rk helm --soft-fail --summary-position bottom - rm -rfv helm_char ts/$EKS_CODEBUILD_ APP_NAME/values.ya ml Agar pipeline gagal saat kerentanan dilaporkan untuk bagan Dockerfile dan Helm, `--soft-fail` opsi harus dihapus dari perintah. `checkov` Pengembang atau insinyur kemudian dapat memperbaiki kerentanan dan melakukan perubahan ke repositori kode CodeCommit sumber. 2. Mirip dengan CodeSecur ity Scan, tahap Build menggunakan Aqua Security Trivy untuk mengidentifikasi `HIGH` dan kerentanan gambar `CRITICAL` Docker sebelum mendorong aplikasi ke Amazon ECR.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>- AWS_REGION=$AWS_DEFAULT_REGION
	 AWS_ACCOUNT_ID=$AWS_ACCOUNT_ID trivy -d image --no-progress --ignore-unfixed --exit-code 0 --severity HIGH,CRITICAL --format template --template "@securityhub/asff.tpl" -o securityhub/report.asff $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$CODEBUILD_RESOLVED_SOURCE_VERSION</pre>
	Dalam contoh ini, pipeline tidak gagal saat kerentanan image Docker dilaporkan, karena `trivy` perintah dalam `buildspec/buildspec.yml` file menyertakan flag **`--exit-code`** dengan nilai. `0` Agar pipeline gagal saat `HIGH` dan `CRTICAL` kerentanan dilaporkan, ubah nilai `--exit-code` to`1`. Pengembang atau insinyur kemudian dapat memperbaiki kerentanan dan melakukan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	perubahan ke repositori kode CodeCommit sumber. 3. Kerentanan gambar Docker yang dilaporkan oleh Aqua Security Trivy diunggah ke Security Hub CSPM. Di konsol CSPM Security Hub, navigasikan ke Temuan. Filter temuan dengan Record State = Active dan Product = Aqua Security. Ini mencantumkan kerentanan gambar Docker di Security Hub CSPM. Diperlukan waktu 15 menit hingga satu jam agar kerentana n muncul di Security Hub CSPM. Untuk informasi selengkap nya tentang memulai pipeline dengan menggunakan CodePipeline, lihat Memulai pipelineCodePipeline, Memulai pipeline secara manual, dan Memulai pipeline sesuai jadwal dalam CodePipeline dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyetujui penyebaran.	1. Setelah fase build selesai, ada gerbang persetuju an penerapan. Peninjau atau manajer rilis harus memeriksa build, dan, jika semua persyarat an terpenuhi, setujui. Ini adalah pendekatan yang direkomendasikan untuk tim yang menggunakan pengiriman berkelanjutan untuk penerapan aplikasi. 2. Setelah disetujui, pipeline memulai tahap Deploy. 3. Setelah tahap Deploy berhasil, CodeBuild log untuk tahap ini menyediak an URL aplikasi. Gunakan URL untuk memvalidasi kesiapan aplikasi.	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi pembuatan profil aplikasi.	Setelah penerapan selesai dan pod aplikasi dikerahkan di Amazon EKS, agen Amazon CodeGuru Profiler yang dikonfigurasi dalam aplikasi akan mencoba mengirim data profil aplikasi (CPU, ringkasan heap, latensi, dan bottleneck) ke Profiler. CodeGuru Untuk penerapan awal aplikasi, CodeGuru Profiler membutuhkan waktu sekitar 15 menit untuk memvisual isasikan data profil.	AWS DevOps

Sumber daya terkait

- AWS CodePipeline dokumentasi
- Memindai gambar dengan Trivy di AWS CodePipeline (posting AWS blog)
- Meningkatkan aplikasi Java Anda menggunakan Amazon CodeGuru Profiler (posting AWS blog)
- AWS Sintaks Security Finding Format (ASFF)
- Pola EventBridge acara Amazon
- Peningkatan helm

Informasi tambahan

- CodeGuru Profiler tidak harus bingung dengan AWS X-Ray layanan dalam hal fungsionalitas. Kami menyarankan Anda menggunakan CodeGuru Profiler untuk mengidentifikasi baris kode paling mahal yang dapat menyebabkan kemacetan atau masalah keamanan, dan memperbaikinya sebelum menjadi risiko potensial. Layanan X-Ray adalah untuk pemantauan kinerja aplikasi.
- Dalam pola ini, aturan acara dikaitkan dengan bus acara default. Jika diperlukan, Anda dapat memperluas pola untuk menggunakan bus acara khusus.

- Pola ini menggunakan CodeGuru Reviewer sebagai alat pengujian keamanan aplikasi statis (SAST) untuk kode aplikasi. Anda juga dapat menggunakan pipeline ini untuk alat lain, seperti SonarQube atau Checkmarx. Anda dapat menambahkan petunjuk pengaturan pemindaian untuk salah satu alat ini `buildspec/buildspec_secscan.yaml` untuk mengganti instruksi CodeGuru pemindaian.

> **Note**
>
> Mulai 7 November 2025, Anda tidak dapat membuat asosiasi repositori baru di Amazon Reviewer. CodeGuru Untuk mempelajari tentang layanan dengan kemampuan yang mirip dengan CodeGuru Reviewer, lihat Perubahan ketersediaan Amazon CodeGuru Reviewer di dokumentasi CodeGuru Reviewer.

Salin gambar wadah Amazon ECR di seluruh Akun AWS dan Wilayah AWS

Faisal Shahdad, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara menggunakan pendekatan tanpa server untuk mereplikasi gambar yang ditandai dari repositori Amazon Elastic Container Registry (Amazon ECR) yang ada ke repositori lain dan. Akun AWS Wilayah AWS Solusinya digunakan AWS Step Functions untuk mengelola alur kerja replikasi dan AWS Lambda fungsi untuk menyalin gambar kontainer besar.

Amazon ECR menggunakan fitur replikasi lintas wilayah dan lintas akun asli yang mereplikasi gambar kontainer di seluruh Wilayah dan akun. Tetapi fitur-fitur ini mereplikasi gambar hanya dari saat replikasi dihidupkan. Tidak ada mekanisme untuk mereplikasi gambar yang ada di Wilayah dan akun yang berbeda.

Pola ini membantu tim kecerdasan buatan (AI) mendistribusikan model pembelajaran mesin kontainerisasi (ML), kerangka kerja (misalnya, PyTorch, dan Hugging Face) TensorFlow, dan dependensi ke akun dan Wilayah lain. Ini dapat membantu Anda mengatasi batas layanan dan mengoptimalkan sumber daya komputasi GPU. Anda juga dapat secara selektif mereplikasi repositori Amazon ECR dari akun sumber dan Wilayah tertentu. Untuk informasi selengkapnya, lihat Replikasi Lintas Wilayah di Amazon ECR telah mendarat.

Prasyarat dan batasan

Prasyarat

- Dua atau lebih aktif Akun AWS (satu akun sumber dan satu akun tujuan, minimal)
- Izin yang sesuai AWS Identity and Access Management (IAM) di semua akun
- Docker untuk membangun gambar wadah Lambda
- AWS Command Line Interface (AWS CLI) dikonfigurasi untuk semua akun

Batasan

- Pengecualian gambar yang tidak ditandai - Solusinya hanya menyalin gambar kontainer yang memiliki tag eksplisit. Ini melewatkan gambar yang tidak ditandai yang ada dengan intisari. `SHA256`
- Batasan batas waktu eksekusi Lambda — AWS Lambda dibatasi hingga batas waktu eksekusi maksimum 15 menit, yang mungkin tidak cukup untuk menyalin gambar atau repositori kontainer besar.

- Manajemen gambar kontainer manual - Kode `crane-app.py` Python memerlukan pembangunan kembali dan penempatan kembali gambar kontainer Lambda.
- Kapasitas pemrosesan paralel terbatas - Pengaturan `MaxConcurrency` status membatasi berapa banyak repositori yang dapat Anda salin pada saat yang bersamaan. Namun, Anda dapat mengubah pengaturan ini di AWS CloudFormation template akun sumber. Perhatikan bahwa nilai konkurensi yang lebih tinggi dapat menyebabkan Anda melebihi batas tingkat layanan dan kuota eksekusi Lambda tingkat akun.

Arsitektur

Target tumpukan

Pola ini memiliki empat komponen utama:

- Infrastruktur akun sumber — CloudFormation template yang membuat komponen orkestrasi
- Infrastruktur akun tujuan - CloudFormation templat yang membuat peran akses lintas akun
- Fungsi Lambda - Fungsi berbasis Python yang menggunakan Crane untuk penyalinan gambar yang efisien
- Gambar kontainer - Wadah Docker yang mengemas fungsi Lambda dengan alat yang diperlukan

Arsitektur target

Alur kerja Step Functions

Mesin status Step Functions mengatur hal berikut, seperti yang ditunjukkan pada diagram berikut:

- `PopulateRepositoryList`— Memindai repositori Amazon ECR dan mengisi Amazon DynamoDB
- `GetRepositoryList`- Mengambil daftar repositori unik dari DynamoDB
- `DeduplicateRepositories`— Memastikan bahwa tidak ada pemrosesan duplikat
- `CopyRepositories`— Menangani penyalinan paralel repositori
- `NotifySuccess`/`NotifyFailure`— Pemberitahuan Amazon Simple Notification Service (Amazon SNS) berdasarkan hasil eksekusi

Alat

Alat Amazon

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi Lambda dan lainnya untuk membangun aplikasi bisnis yang penting. Layanan AWS

Alat-alat lainnya

- Crane adalah alat orkestrasi Docker. Ini mirip dengan Docker Compose tetapi memiliki fitur tambahan.
- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi di tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.

Repositori kode

- Kode untuk pola ini tersedia di GitHub sample-ecr-copy repositori. Anda dapat menggunakan CloudFormation template dari repositori untuk membuat sumber daya yang mendasarinya.

Praktik terbaik

Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Persiapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan AWS CLI profil.	1. Konfigurasikan profil akun sumber:
	`aws configure --profile source-account`
	`# Enter: Access Key ID, Secret Access Key, Default region, Output format (json)`
	2. Konfigurasikan profil akun tujuan:
	`aws configure --profile destination-account`
	`# Enter: Access Key ID, Secret Access Key, Default region, Output format (json)`
	3. Verifikasi konfigurasi:
	`aws sts get-caller-identity --profile source-account`
		DevOps insinyur, insinyur data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws sts get-caller-identity --profile destination-account`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kumpulkan informasi yang diperlukan.	1. Dapatkan ID akun sumber:
	`export SOURCE_ACCOUNT_ID=$(aws sts get-caller-identity --profile source-account --query Account --output text)`
	`echo "Source Account ID: $SOURCE_ACCOUNT_ID"`
	2. Dapatkan ID akun tujuan:
	`export DEST_ACCOUNT_ID=$(aws sts get-caller-identity --profile destination-account --query Account --output text)`
	`echo "Destination Account ID: $DEST_ACCOUNT_ID"`
	3. Mengatur Wilayah AWS. Ubah perintah ini untuk Wilayah Anda:
	`export SOURCE_REGION="us-east-1"`
	`export DEST_REGION="us-east-2"`
	4. Buat daftar repositori Amazon ECR yang ada di akun sumber:
	`aws ecr describe-repositories --`
		DevOps insinyur, insinyur data, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`profile source-account --region $SOURCE_REGION --query 'repositories[].repositoryName' --output table`	
Kloning repositori.	Kloning repositori pola ke workstation lokal Anda: `git clone https://github.com/aws-samples/sample-ecr-copy`	DevOps insinyur, insinyur data, insinyur ML

Menyebarkan infrastruktur untuk akun tujuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi template.	Validasi CloudFormation template: `aws cloudformation validate-template \ --template-body file://"Destination Account cf_template.yml" \ --profile destination-account`	DevOps insinyur, insinyur ML, Insinyur data
Menyebarkan infrastruktur tujuan.	1. Terapkan tumpukan akun tujuan: `aws cloudformation deploy \`	Insinyur data, insinyur ML, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--template-file "Destination Account cf_template.yml" \
	--stack-name ecr-copy-destination \
	--parameter-overrides \
	SourceAccountId=$SOURCE_ACCOUNT_ID \
	SourceRoleName=ECRContainerLambdaRole \
	--capabilities CAPABILITY_NAMED_IAM \
	--profile destination-account \
	--region $DEST_REGION</pre>
	2. Tunggu hingga tumpukan selesai:
	<pre>aws cloudformation wait stack-create-complete \
	--stack-name ecr-copy-destination \
	--profile destination-account \
	--region $DEST_REGION</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi penyebaran.	1. Dapatkan output tumpukan:
	`aws cloudformation describe-stacks \ --stack-name ecr-copy-destination \ --profile destination-account \ --region $DEST_REGION \ --query 'Stacks[0].Outputs' \ --output table`
	2. Simpan peran IAM lintas akun:
	`export CROSS_ACCOUNT_ROLE_ARN=$(aws cloudformation describe-stacks \ --stack-name ecr-copy-destination \ --profile destination-account \ --region $DEST_REGION \ --query 'Stacks[0].Outputs[?OutputKey==`CrossAccountRoleArn`].OutputValue' \ --output text)`
	`echo "Cross-Account Role ARN: $CROSS_ACCOUNT_ROLE_ARN"`
		DevOps insinyur, insinyur ML, Insinyur data

Bangun dan terapkan gambar kontainer Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan kontainer build.	1. Verifikasi bahwa Docker sedang berjalan:
	`docker --version`
	`docker info`
	2. Pastikan itu `crane-app.py` dan `Dockerfile` berada di direktori saat ini:
	`ls -la crane-app.py Dockerfile`
		Insinyur data, insinyur ML, DevOps insinyur

Bangun gambar kontainer.	1. Bangun gambar wadah Lambda:
	`docker build -t ecr-copy-lambda . --no-cache`
	2. Verifikasi bahwa gambar telah dibuat:
	`docker images ecr-copy-lambda`
	3. (Opsional) Uji wadah secara lokal:
	`docker run --rm --entrypoint python ecr-copy-lambda -c "import boto3; print('Container working')"`
		Insinyur data, insinyur ML, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat repositori dan unggah gambar.	1. Buat repositori Amazon ECR di akun sumber:
	`aws ecr create-repository \ --repository-name ecr-copy-lambda \ --profile source-account \ --region $SOURCE_REGION`
	2. Dapatkan token login Amazon ECR dan autentikasi Docker:
	`aws ecr get-login-password \ --profile source-account \ --region $SOURCE_REGION | \ docker login --username AWS --password-stdin \ $SOURCE_ACCOUNT_ID.dkr.ecr.$SOURCE_REGION.amazonaws.com`
	3. Tandai gambar untuk Amazon ECR:
	`docker tag ecr-copy-lambda:latest \ $SOURCE_ACCOUNT_ID.dkr.ecr.$SOURCE_REGION.amazonaws.com/ecr-copy-lambda:latest`
		Insinyur data, insinyur ML, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Unggah gambar ke Amazon ECR:
	`docker push $SOURCE_ACCOUNT_ID.dkr.ecr.$SOURCE_REGION.amazonaws.com/ecr-copy-lambda:latest`
	5. Simpan URI gambar untuk digunakan nanti:
	`export LAMBDA_IMAGE_URI="$SOURCE_ACCOUNT_ID.dkr.ecr.$SOURCE_REGION.amazonaws.com/ecr-copy-lambda:latest"`
	`echo "Lambda Image URI: $LAMBDA_IMAGE_URI"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi gambar.	1. Daftar gambar di repositori:
	`aws ecr list-images \ --repository-name ecr-copy-lambda \ --profile source-account \ --region $SOURCE_REGION`
	2. Dapatkan detail gambar:
	`aws ecr describe-images \ --repository-name ecr-copy-lambda \ --profile source-account \ --region $SOURCE_REGION`
		Insinyur data, insinyur ML, DevOps insinyur

Menyebarkan infrastruktur akun sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan parameter penerapan.	1. Mengatur email notifikasi:
	`export NOTIFICATION_EMAIL="your-email@company.com"`
	2. Tentukan repositori yang akan disalin (dipisahkan koma):
	`export REPOSITORY_LIST="app-fronte`
		Insinyur data, DevOps insinyur, insinyur ML

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`nd,app-backend,dat abase-migrations"`
	3. Mengatur lingkungan:
	`export ENVIRONME NT="dev"`
	
	`echo "Deployment Parameters:" echo "Source Account: $SOURCE_ACCOUNT_ID" echo "Destination Account: $DEST_ACC OUNT_ID" echo "Source Region: $SOURCE_REGION" echo "Destination Region: $DEST_REG ION" echo "Lambda Image: $LAMBDA_IMAGE_URI" echo "Notification Email: $NOTIFICA TION_EMAIL" echo "Repositories: $REPOSITORY_LIST"`	

Validasi template sumber.	Validasi CloudFormation template sumber: `aws cloudformation validate-template \ --template-body file://"Source Account Cf template.yml" \ --profile source-ac count`	Insinyur data, insinyur ML, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan infrastruktur sumber.	1. Terapkan tumpukan akun sumber: `aws cloudformation deploy \ --template-file "Source Account Cf template.yml" \ --stack-name ecr-copy-source \ --parameter-overrides \ SourceAccountId=$SOURCE_ACCOUNT_ID \ DestinationAccountId=$DEST_ACCOUNT_ID \ DestinationRegion=$DEST_REGION \ SourceRegion=$SOURCE_REGION \ NotificationEmail=$NOTIFICATION_EMAIL \ RepositoryList="$REPOSITORY_LIST" \ LambdaImageUri=$LAMBDA_IMAGE_URI \ Environment=$ENVIRONMENT \ --capabilities CAPABILITY_NAMED_IAM \ --profile source-account \ --region $SOURCE_REGION`	Insinyur data, insinyur ML, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Tunggu hingga tumpukan selesai (ini mungkin memakan waktu hingga 10 menit): `aws cloudformation wait stack-create-complete \ --stack-name ecr-copy-source \ --profile source-account \ --region $SOURCE_REGION`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi penerapan dan kumpulkan output.	1. Dapatkan output tumpukan:
	
	`aws cloudformation describe-stacks \ --stack-name ecr-copy-source \ --profile source-account \ --region $SOURCE_REGION \ --query 'Stacks[0].Outputs' \ --output table`
	
	2. Simpan Amazon Resource Names (ARNs) untuk mesin status dan topik SNS:
	
	`export STATE_MACHINE_ARN=$(aws cloudformation describe-stacks \ --stack-name ecr-copy-source \ --profile source-account \ --region $SOURCE_REGION \ --query 'Stacks[0].Outputs[?OutputKey==`StateMachineArn`].OutputValue' \ --output text)`
	
	`export SNS_TOPIC_ARN=$(aws cloudformation describe-stacks \ --stack-name ecr-copy-source \`
		DevOps insinyur, insinyur ML, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> --profile source-account \
	  --region $SOURCE_REGION \
	  --query 'Stacks[0].Outputs[?OutputKey==`SNSTopicArn`].OutputValue' \
	  --output text)
	
	echo "State Machine ARN: $STATE_MACHINE_ARN"
	echo "SNS Topic ARN: $SNS_TOPIC_ARN"</pre>	

Konfirmasikan langganan email Anda.	1. Periksa email Anda untuk konfirmasi langganan SNS Anda.
	2. Pilih tautan konfirmasi di email.
	3. Verifikasi status langganan.
	<pre>aws sns list-subscriptions-by-topic \
	  --topic-arn $SNS_TOPIC_ARN \
	  --profile source-account \
	  --region $SOURCE_REGION</pre>
		Insinyur data, insinyur ML, DevOps insinyur

Jalankan dan pantau proses penyalinan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan dan pantau proses penyalinan.	1. Masuk ke Konsol Manajemen AWS, dan buka konsol Step Functions. 2. Temukan mesin negara. 3. Pilih Mulai Eksekusi. Setelah selesai, hasil ditampilkan pada tab input dan output Eksekusi. 4. (Opsional) Jika Anda ingin terus menjalankan Step Functions dengan menggunakan AWS CLI, ikuti langkah-langkah yang tersisa dalam epik ini.	DevOps insinyur, insinyur ML, Insinyur data

Jalankan fungsi langkah.	1. Hasilkan nama unik:
	`export EXECUTION_NAME="ecr-copy-$(date +%Y%m%d-%H%M%S)"`
	2. Jalankan fungsi langkah.
	`export EXECUTION_ARN=$(aws stepfunctions start-execution \ --state-machine-arn $STATE_MACHINE_ARN \ --name $EXECUTION_NAME \ --profile source-account \`
		DevOps insinyur, insinyur ML, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> --region $SOURCE_R
	EGION \
	  --query 'executio
	nArn' \
	  --output text)
	
	echo "Execution
	 started: $EXECUTIO
	N_ARN"
	echo "Execution
	 Name: $EXECUTIO
	N_NAME"</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Pantau kemajuan.	1. Periksa statusnya:
	`aws stepfunctions describe-execution \ --execution-arn $EXECUTION_ARN \ --profile source-account \ --region $SOURCE_REGION \ --query '{Status:status,StartDate:startDate,StopDate:stopDate}' \ --output table`
	2. Dapatkan sejarahnya:
	`aws stepfunctions get-execution-history \ --execution-arn $EXECUTION_ARN \ --profile source-account \ --region $SOURCE_REGION \ --query 'events[?type==`TaskStateEntered` || type==`TaskSucceeded` || type==`TaskFailed`].{Type:type,Timestamp:timestamp,Details:stateEnteredEventDetails.name}' \ --output table`
		DevOps insinyur, insinyur ML, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa hasilnya.	Tunggu proses selesai (diperbarui setiap 30 detik):
	
	`while true; do`
	`  STATUS=$(aws`
	` stepfunctions`
	` describe-execution \`
	`  --execution-arn`
	` $EXECUTION_ARN \`
	`  --profile source-ac`
	`count \`
	`  --region $SOURCE_R`
	`EGION \`
	`  --query 'status' \`
	`  --output text)`
	
	`  echo "Current status:`
	` $STATUS"`
	
	`  if [[ "$STATUS"`
	` == "SUCCEEDED"`
	` || "$STATUS" ==`
	` "FAILED" || "$STATUS"`
	` == "TIMED_OUT"`
	` || "$STATUS" ==`
	` "ABORTED" ]]; then`
	`  break`
	`  fi`
	
	`  sleep 30`
	`done`
	
	`echo "Final execution`
	` status: $STATUS"`
		DevOps insinyur, insinyur ML, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi gambar.	1. Buat daftar repositori di akun tujuan:
	`aws ecr describe-repositories \ --profile destination-account \ --region $DEST_REGION \ --query 'repositories[].repositoryName' \ --output table`
	2. Periksa gambar repositori:
	`for repo in $(echo $REPOSITORY_LIST | tr ',' ' '); do echo "\nImages in repository: $repo" aws ecr list-images \ --repository-name $repo \ --profile destination-account \ --region $DEST_REGION \ --query 'imageIds[].imageTag' \ --output table 2>/dev/null || echo "Repository $repo not found or no images" done`
		DevOps insinyur, insinyur data, insinyur ML

Pemecahan masalah

Isu	Solusi

Fungsi langkah gagal dijalankan.	1. Untuk mengambil peristiwa kegagalan terperinci dari riwayat, jalankan AWS CLI perintah berikut:
	`if [[ "$STATUS" == "FAILED" ]]; then`
	`  echo "Getting failure details..."`
	`  aws stepfunctions get-execution-history \`
	`  --execution-arn $EXECUTION_ARN \`
	`  --profile source-account \`
	`  --region $SOURCE_REGION \`
	`  --query 'events[?type==`TaskFailed`]' \`
	`  --output json`
	`fi`
	2. Untuk mengambil log untuk fungsi Lambda yang gagal, jalankan perintah berikut AWS CLI :
	`# Check Lambda function logs`
	`echo "\nLambda function logs:"`
	`aws logs describe-log-groups \`
	`  --log-group-name-prefix "/aws/lambda/ecr-copy-source" \`
	`  --profile source-account \`
	`  --region $SOURCE_REGION \`
	`  --query 'logGroups[].logGroupName' \`
	`  --output table`

Sumber daya terkait

- Dokumentasi derek

- Apa itu Amazon Elastic Container Registry?
- Apa itu AWS Lambda?
- Apa itu Step Functions?

Informasi tambahan

Parameter konfigurasi

Parameter	Deskripsi	Contoh
`SourceAccountId`	Akun AWS ID Sumber	`11111111111`
`DestinationAccountId`	Akun AWS ID Tujuan	`22222222222`
`DestinationRegion`	Target Wilayah AWS	`us-east-2`
`SourceRegion`	Sumber Wilayah AWS	`us-east-1`
`NotificationEmail`	Email untuk notifikasi	`abc@xyz.com`
`RepositoryList`	Repositori untuk disalin	`repo1,repo2,repo3`
`LambdaImageUri`	URI gambar kontainer Lambda	`${ACCOUNT}.dkr.ecr.${REGION}.amazonaws.com/ecr-copy-lambda:latest`

Buat definisi tugas Amazon ECS dan pasang sistem file pada EC2 instans menggunakan Amazon EFS

Durga Prasad Cheepuri, Amazon Web Services

Ringkasan

Pola ini menyediakan contoh kode dan langkah-langkah untuk membuat definisi tugas Amazon Elastic Container Service (Amazon ECS) yang berjalan pada instans Amazon Elastic Compute Cloud (EC2Amazon) di Amazon Web Services (AWS) Cloud, saat menggunakan Amazon Elastic File System (Amazon EFS) untuk memasang sistem file pada instance tersebut. EC2 Tugas Amazon ECS yang menggunakan Amazon EFS secara otomatis memasang sistem file yang Anda tentukan dalam definisi tugas dan membuat sistem file ini tersedia untuk penampung tugas di semua Availability Zone di Wilayah AWS.

Untuk memenuhi persyaratan penyimpanan persisten dan penyimpanan bersama, Anda dapat menggunakan Amazon ECS dan Amazon EFS bersama-sama. Misalnya, Anda dapat menggunakan Amazon EFS untuk menyimpan data pengguna dan data aplikasi persisten untuk aplikasi Anda dengan pasangan kontainer ECS aktif dan siaga yang berjalan di Availability Zone yang berbeda untuk ketersediaan tinggi. Anda juga dapat menggunakan Amazon EFS untuk menyimpan data bersama yang dapat diakses secara paralel oleh kontainer ECS dan beban kerja terdistribusi.

Untuk menggunakan Amazon EFS dengan Amazon ECS, Anda dapat menambahkan satu atau beberapa definisi volume ke definisi tugas. Definisi volume mencakup ID sistem file Amazon EFS, ID titik akses, dan konfigurasi untuk otorisasi AWS Identity and Access Management (IAM) atau enkripsi Transport Layer Security (TLS) dalam perjalanan. Anda dapat menggunakan definisi kontainer dalam definisi tugas untuk menentukan volume definisi tugas yang dipasang saat penampung berjalan. Saat tugas yang menggunakan sistem file Amazon EFS berjalan, Amazon ECS memastikan bahwa sistem file terpasang dan tersedia untuk wadah yang memerlukan akses ke sana.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Virtual Private Cloud (VPC) dengan titik akhir virtual private network (VPN) atau router
- (Direkomendasikan) Agen kontainer Amazon ECS 1.38.0 atau yang lebih baru untuk kompatibilitas dengan jalur akses Amazon EFS dan fitur otorisasi IAM (Untuk informasi selengkapnya, lihat postingan blog AWS Baru untuk Amazon EFS — Otorisasi IAM dan Titik Akses.)

Batasan

- Versi agen penampung Amazon ECS lebih awal dari 1.35.0 tidak mendukung sistem file Amazon EFS untuk tugas yang menggunakan jenis peluncuran. EC2

Arsitektur

Diagram berikut menunjukkan contoh aplikasi yang menggunakan Amazon ECS untuk membuat definisi tugas dan memasang sistem file Amazon EFS pada EC2 instance dalam wadah ECS.

Diagram menunjukkan alur kerja berikut:

1. Buat sistem file Amazon EFS.
2. Buat definisi tugas dengan wadah.
3. Konfigurasikan instance kontainer untuk memasang sistem file Amazon EFS. Definisi tugas mereferensikan pemasangan volume, sehingga instance container dapat menggunakan sistem file Amazon EFS. Tugas ECS memiliki akses ke sistem file Amazon EFS yang sama, terlepas dari instance container mana tugas tersebut dibuat.
4. Buat layanan Amazon ECS dengan tiga contoh definisi tugas.

Tumpukan teknologi

- Amazon EC2
- Amazon ECS
- Amazon EFS

Alat

- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, dan Anda dapat meningkatkan skala atau meningkatkan skala.
- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat untuk menjalankan, menghentikan, dan mengelola kontainer di cluster. Anda dapat menjalankan tugas dan layanan Anda pada infrastruktur tanpa

server yang dikelola oleh AWS Fargate. Atau, untuk kontrol lebih besar atas infrastruktur Anda, Anda dapat menjalankan tugas dan layanan Anda pada sekelompok EC2 instance yang Anda kelola.

- Amazon EFS — Amazon Elastic File System (Amazon EFS) menyediakan sistem file NFS elastis yang sederhana, dapat diskalakan, dan dikelola sepenuhnya untuk digunakan dengan layanan AWS Cloud dan sumber daya lokal.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat sumber terbuka untuk berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda. Dengan konfigurasi minimal, Anda dapat menjalankan perintah AWS CLI yang mengimplementasikan fungsionalitas yang setara dengan yang disediakan oleh AWS Management Console berbasis browser dari prompt perintah.

Epik

Buat sistem file Amazon EFS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file Amazon EFS menggunakan AWS Management Console.	1. **Note** Buat sistem file Amazon EFS dan pilih VPC yang menyertakan kontainer Anda. : Jika Anda menggunakan VPC yang berbeda, siapkan koneksi peering VPC. 2. Catat ID sistem file.	AWS DevOps

Buat definisi tugas Amazon ECS dengan menggunakan sistem file Amazon EFS atau AWS CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat definisi tugas menggunakan sistem file Amazon EFS.	Buat definisi tugas dengan menggunakan konsol Amazon ECS baru atau konsol Amazon ECS klasik dengan konfigurasi berikut: • Jika Anda menggunakan konsol baru, pilih EC2 instans Amazon untuk lingkungan Aplikasi. Jika Anda menggunakan konsol klasik, pilih EC2sebagai jenis peluncuran. • Tambahkan volume. Masukkan nama untuk volume, pilih EFS untuk jenis volume, lalu pilih ID sistem file yang Anda catat sebelumnya. Untuk direktori root, pilih jalur sistem file Amazon EFS yang ingin Anda host di host penampung Amazon ECS.	AWS DevOps
Buat definisi tugas menggunakan AWS CLI.	1. Untuk membuat template JSON dengan placeholder parameter input untuk definisi tugas Anda, jalankan perintah berikut: `aws ecs register-task-definition`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--generate-cli-skeleton`
	2. Untuk membuat definisi tugas dengan template JSON, jalankan perintah berikut:
	`aws ecs register-task-definition --cli-input-json file://<path_to_your_json_file>`
	3. **Note**
	Masukkan parameter input dalam template JSON Anda berdasarkan `task_definition_parameters.json` file (terlampir). : Untuk informasi selengkapnya tentang parameter input, lihat Parameter definisi tugas (dokumentasi Amazon ECS) dan register-task-definition(AWS CLI Command Reference).	

Sumber daya terkait

- Definisi tugas Amazon ECS
- Volume Amazon EFS

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menyebarkan fungsi Lambda dengan gambar kontainer

Ram Kandaswamy, Amazon Web Services

Ringkasan

AWS Lambda mendukung gambar kontainer sebagai model penerapan. Pola ini menunjukkan cara menerapkan fungsi Lambda melalui gambar kontainer.

Lambda adalah layanan komputasi tanpa server dan berbasis peristiwa yang dapat Anda gunakan untuk menjalankan kode untuk hampir semua jenis aplikasi atau layanan backend tanpa menyediakan atau mengelola server. Dengan dukungan gambar kontainer untuk fungsi Lambda, Anda mendapatkan manfaat penyimpanan hingga 10 GB untuk artefak aplikasi Anda dan kemampuan untuk menggunakan alat pengembangan gambar kontainer yang sudah dikenal.

Contoh dalam pola ini menggunakan Python sebagai bahasa pemrograman yang mendasarinya, tetapi Anda dapat menggunakan bahasa lain, seperti Java, Node.js, atau Go. Untuk sumbernya, pertimbangkan sistem berbasis Git seperti GitHub,, atau Bitbucket GitLab, atau gunakan Amazon Simple Storage Service (Amazon S3).

Prasyarat dan batasan

Prasyarat

- Amazon Elastic Container Registry (Amazon ECR) diaktifkan
- Kode aplikasi
- Gambar Docker dengan klien antarmuka runtime dan versi terbaru Python
- Pengetahuan tentang Git

Batasan

- Ukuran gambar maksimum yang didukung adalah 10 GB.
- Runtime maksimum untuk penerapan kontainer berbasis Lambda adalah 15 menit.

Arsitektur

Arsitektur target

1. Anda membuat repositori Git dan mengkomit kode aplikasi ke repositori.
2. AWS CodeBuild Proyek ini dipicu oleh perubahan komit.
3. CodeBuild Proyek ini membuat gambar Docker dan menerbitkan gambar yang dibangun ke Amazon ECR.
4. Anda membuat fungsi Lambda menggunakan gambar di Amazon ECR.

Otomatisasi dan skala

Pola ini dapat diotomatisasi dengan menggunakan AWS CloudFormation AWS Cloud Development Kit (AWS CDK),, atau operasi API dari SDK. Lambda dapat secara otomatis menskalakan berdasarkan jumlah permintaan, dan Anda dapat menyetelnya dengan menggunakan parameter konkurensi. Untuk informasi selengkapnya, lihat dokumentasi Lambda.

Alat

Layanan AWS

- AWS CloudFormationAWS CloudFormationhelps Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS Pola ini menggunakan AWS CloudFormation Application Composer, yang membantu Anda melihat dan mengedit CloudFormation template secara visual.
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- GitHub, GitLab, dan Bitbucket adalah beberapa sistem kontrol sumber berbasis Git yang umum digunakan untuk melacak perubahan kode sumber.

Praktik terbaik

- Jadikan fungsi Anda seefisien dan sekecil mungkin untuk menghindari memuat file yang tidak perlu.
- Berusahalah untuk memiliki lapisan statis yang lebih tinggi di daftar file Docker Anda, dan tempatkan lapisan yang lebih sering berubah di bawah. Ini meningkatkan caching, yang meningkatkan kinerja.
- Pemilik gambar bertanggung jawab untuk memperbarui dan menambal gambar. Tambahkan irama pembaruan itu ke proses operasional Anda. Lihat informasi yang lebih lengkap dalam dokumentasi AWS Lambda.

Epik

Buat proyek di CodeBuild

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Git.	Buat repositori Git yang akan berisi kode sumber aplikasi, Dockerfile, dan file. `buildspec.yaml`	Developer
Buat CodeBuild proyek.	Untuk menggunakan CodeBuild proyek untuk membuat gambar Lambda kustom, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan buka CodeBuild konsol di https://console.aws.amazon.com/codesuite/ codebuild/. 2. Buat proyek baru. Untuk sumber, pilih repositori Git yang Anda buat. Untuk informasi tentang berbagai jenis integrasi repositori Git,	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lihat dokumentasi Bekerja dengan koneksi. 3. Konfirmasikan bahwa mode istimewa diaktifkan. Untuk membuat gambar Docker, ini diperlukan. Jika tidak, gambar tidak akan berhasil dibangun. 4. Berikan nilai untuk nama dan deskripsi proyek.	
Edit Dockerfile.	Dockerfile harus ditempatkan di direktori tingkat atas tempat Anda mengembangkan aplikasi. Kode Python harus ada di folder. `src` Saat Anda membuat gambar, gunakan gambar resmi yang didukung Lambda. Jika tidak, kesalahan bootstrap akan terjadi, membuat proses pengepakan lebih sulit. Untuk detailnya, lihat bagian Informasi tambahan.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori di Amazon ECR.	Buat repositori kontainer di Amazon ECR. Dalam contoh perintah berikut, nama repositori yang dibuat adalah: `cf-demo` `aws ecr create-repository --cf-demo` Repositori akan direferensikan dalam file. `buildspec.yaml`	Administrator AWS, Pengembang
Dorong gambar ke Amazon ECR.	Anda dapat menggunakannya CodeBuild untuk melakukan proses pembuatan gambar. CodeBuild memerlukan izin untuk berinteraksi dengan Amazon ECR dan bekerja dengan S3. Sebagai bagian dari proses, image Docker dibuat dan didorong ke registri Amazon ECR. Untuk detail tentang template dan kode, lihat bagian Informasi tambahan.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa gambar ada di repositori.	Untuk memverifikasi bahwa gambar ada di repositor i, di konsol Amazon ECR, pilih Repositori. Gambar harus terdaftar, dengan tag dan dengan hasil laporan pemindaian kerentanan jika fitur itu diaktifkan di pengatura n Amazon ECR. Untuk informasi selengkapnya, lihat dokumentasi AWS.	Developer

Buat fungsi Lambda untuk menjalankan gambar

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda.	Di konsol Lambda, pilih Create function, lalu pilih Container image. Masukkan nama fungsi dan URI untuk gambar yang ada di repositori Amazon ECR, lalu pilih Create function. Untuk informasi selengkap nya, lihat dokumentasi AWS Lambda.	Pengembang aplikasi
Uji fungsi Lambda.	Untuk memanggil dan menguji fungsi, pilih Uji. Untuk informasi selengkapnya, lihat dokumentasi AWS Lambda.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Membangun tidak berhasil.	1. Periksa apakah mode istimewa dihidupkan untuk CodeBuild proyek. 2. Pastikan bahwa perintah terkait Docker memiliki izin yang diperlukan. `sudo`Mencoba menambahkan perintah. 3. Verifikasi bahwa peran IAM yang terkait dengan CodeBuild memiliki kebijakan dengan tindakan yang sesuai untuk berinteraksi dengan Amazon ECR, Amazon S3, dan log. CloudWatch

Sumber daya terkait

- Gambar dasar untuk Lambda
- Sampel Docker untuk CodeBuild
- Lulus kredensyal sementara

Informasi tambahan

Edit Dockerfile

Kode berikut menunjukkan perintah yang Anda edit di Dockerfile:

```
FROM public.ecr.aws/lambda/python:3.xx

# Copy function code
COPY app.py ${LAMBDA_TASK_ROOT}
COPY requirements.txt  ${LAMBDA_TASK_ROOT}

# install dependencies
RUN pip3 install --user -r requirements.txt

# Set the CMD to your handler (could also be done as a parameter override outside of
 the Dockerfile)
```

```
CMD [ "app.lambda_handler" ]
```

Dalam `FROM` perintah, gunakan nilai yang sesuai untuk versi Python yang didukung oleh Lambda (misalnya,). `3.12` Ini akan menjadi gambar dasar yang tersedia di repositori gambar Amazon ECR publik.

`COPY app.py ${LAMBDA_TASK_ROOT}`Perintah menyalin kode ke direktori root tugas, yang akan digunakan fungsi Lambda. Perintah ini menggunakan variabel lingkungan sehingga kita tidak perlu khawatir tentang jalur yang sebenarnya. Fungsi yang akan dijalankan diteruskan sebagai argumen ke `CMD [ "app.lambda_handler" ]` perintah.

`COPY requirements.txt`Perintah menangkap dependensi yang diperlukan untuk kode.

`RUN pip install --user -r requirements.txt`Perintah menginstal dependensi ke direktori pengguna lokal.

Untuk membangun gambar Anda, jalankan perintah berikut.

```
docker build -t <image name> .
```

Tambahkan gambar di Amazon ECR

Dalam kode berikut, ganti `aws_account_id` dengan nomor akun, dan ganti `us-east-1` jika Anda menggunakan Wilayah yang berbeda. `buildspec`File menggunakan nomor CodeBuild build untuk mengidentifikasi versi gambar secara unik sebagai nilai tag. Anda dapat mengubah ini agar sesuai dengan kebutuhan Anda.

Kode kustom buildspec

```
phases:
  install:
    runtime-versions:
       python: 3.xx
  pre_build:
    commands:
      - python3 --version
      - pip3 install --upgrade pip
      - pip3 install --upgrade awscli
      - sudo docker info
  build:
    commands:
```

```
      - echo Build started on `date`
      - echo Building the Docker image...
      - ls
      - cd app
      - docker build -t cf-demo:$CODEBUILD_BUILD_NUMBER .
      - docker container ls
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin aws_account_id.dkr.ecr.us-east-1.amazonaws.com
      - docker tag cf-demo:$CODEBUILD_BUILD_NUMBER aws_account_id.dkr.ecr.us-
east-1.amazonaws.com/cf-demo:$CODEBUILD_BUILD_NUMBER
      - docker push aws_account_id.dkr.ecr.us-east-1.amazonaws.com/cf-demo:
$CODEBUILD_BUILD_NUMBER
```

Menerapkan layanan mikro Java di Amazon ECS menggunakan AWS Fargate

Vijay Thompson dan Sandeep Bondugula, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk menerapkan layanan mikro Java dalam peti kemas di Amazon Elastic Container Service (Amazon ECS) Container Service (Amazon ECS) dengan menggunakan AWS Fargate. Polanya tidak menggunakan Amazon Elastic Container Registry (Amazon ECR) Registry (Amazon ECR) untuk manajemen kontainer; sebagai gantinya, gambar Docker ditarik dari hub Docker.

Prasyarat dan batasan

Prasyarat

- Aplikasi microservices Java yang ada di hub Docker
- Sebuah repositori Docker publik
- Akun AWS yang aktif
- Keakraban dengan layanan AWS, termasuk Amazon ECS dan Fargate
- Kerangka kerja Docker, Java, dan Spring Boot
- Amazon Relational Database Service (Amazon RDS) aktif dan berjalan (opsional)
- Virtual private cloud (VPC) jika aplikasi memerlukan Amazon RDS (opsional)

Arsitektur

Tumpukan teknologi sumber

- Layanan mikro Java (misalnya, diimplementasikan di Spring Boot) dan digunakan di Docker

Arsitektur sumber

Tumpukan teknologi target

- Cluster Amazon ECS yang menampung setiap layanan mikro dengan menggunakan Fargate

- Jaringan VPC untuk meng-host cluster Amazon ECS dan grup keamanan terkait
- cluster/task Definisi untuk setiap layanan mikro yang memutar kontainer dengan menggunakan Fargate

Arsitektur target

Alat

Alat

- Amazon ECS menghilangkan kebutuhan untuk menginstal dan mengoperasikan perangkat lunak orkestrasi kontainer Anda sendiri, mengelola dan menskalakan sekelompok mesin virtual, atau menjadwalkan kontainer pada mesin virtual tersebut.
- AWS Fargate membantu Anda menjalankan container tanpa perlu mengelola server atau instans Amazon Elastic Compute Cloud (Amazon). EC2 Ini digunakan bersama dengan Amazon Elastic Container Service (Amazon ECS).
- Docker adalah platform perangkat lunak yang memungkinkan Anda membangun, menguji, dan menyebarkan aplikasi dengan cepat. Docker mengemas perangkat lunak ke dalam unit standar yang disebut kontainer yang memiliki semua yang dibutuhkan perangkat lunak untuk dijalankan, termasuk perpustakaan, alat sistem, kode, dan runtime.

Kode Docker

Dockerfile berikut menentukan versi Java Development Kit (JDK) yang digunakan, di mana file arsip Java (JAR) ada, nomor port yang terbuka, dan titik masuk untuk aplikasi.

```
FROM openjdk:11
ADD target/Spring-docker.jar Spring-docker.jar
EXPOSE 8080
ENTRYPOINT ["java","-jar","Spring-docker.jar"]
```

Epik

Buat definisi tugas baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat definisi tugas.	Menjalankan wadah Docker di Amazon ECS memerluka n definisi tugas. Buka konsol Amazon ECS di https:// console.aws.amazon.com/ ecs/, pilih Definisi tugas, lalu buat definisi tugas baru. Untuk informasi selengkapnya, lihat dokumentasi Amazon ECS.	Administrator sistem AWS, Pengembang aplikasi
Pilih jenis peluncuran.	Pilih Fargate sebagai tipe peluncuran.	Administrator sistem AWS, Pengembang aplikasi
Konfigurasikan tugas.	Tentukan nama tugas dan konfigurasikan aplikasi dengan jumlah memori tugas dan CPU yang sesuai.	Administrator sistem AWS, Pengembang aplikasi
Tentukan wadahnya.	Tentukan nama wadah. Untuk gambar, masukkan nama situs Docker, nama repositori, dan nama tag image Docker (). `docker.io/sample-r epo/sample-applica tion:sample-tag-na me` Tetapkan batas memori untuk aplikasi, dan atur pemetaan port (`8080, 80`) untuk port yang diizinkan.	Administrator sistem AWS, Pengembang aplikasi
Buat tugas.	Saat konfigurasi tugas dan wadah sudah ada, buat tugas.	Administrator sistem AWS, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk petunjuk terperinci, lihat tautan di bagian Sumber daya terkait.	

Konfigurasikan cluster

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan cluster.	Pilih Jaringan hanya sebagai tipe cluster, konfigurasikan nama, lalu buat cluster atau gunakan cluster yang ada jika tersedia. Untuk informasi selengkapnya, lihat dokumentasi Amazon ECS.	Administrator sistem AWS, Pengembang aplikasi

Konfigurasikan Tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas.	Di dalam cluster, pilih Jalankan tugas baru.	Administrator sistem AWS, Pengembang aplikasi
Pilih jenis peluncuran.	Pilih Fargate sebagai tipe peluncuran.	Administrator sistem AWS, Pengembang aplikasi
Pilih definisi tugas, revisi, dan versi platform.	Pilih tugas yang ingin Anda jalankan, revisi definisi tugas, dan versi platform.	Administrator sistem AWS, Pengembang aplikasi
Pilih cluster.	Pilih cluster tempat Anda ingin menjalankan tugas.	Administrator sistem AWS, Pengembang aplikasi
Tentukan jumlah tugas.	Konfigurasikan jumlah tugas yang harus dijalanka	Administrator sistem AWS, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n. Jika Anda meluncurkan dengan dua tugas atau lebih, penyeimbang beban diperluka n untuk mendistribusikan lalu lintas di antara tugas-tugas.	
Tentukan kelompok tugas.	(Opsional) Tentukan nama grup tugas untuk mengident ifikasi serangkaian tugas terkait sebagai kelompok tugas.	Administrator sistem AWS, Pengembang aplikasi
Konfigurasikan VPC cluster, subnet, dan grup keamanan.	Konfigurasikan VPC cluster dan subnet tempat Anda ingin menyebarkan aplikasi. Buat atau perbarui grup keamanan (HTTP, HTTPS, dan port 8080) untuk menyediakan akses ke koneksi masuk dan keluar.	Administrator sistem AWS, Pengembang aplikasi
Konfigurasikan pengaturan IP publik.	Aktifkan atau nonaktifk an IP publik, tergantung pada apakah Anda ingin menggunakan alamat IP publik untuk tugas Fargate. Opsi default yang direkomen dasikan adalah Diaktifkan.	Administrator sistem AWS, Pengembang aplikasi
Tinjau pengaturan dan buat tugas	Tinjau pengaturan Anda, lalu pilih Jalankan Tugas.	Administrator sistem AWS, Pengembang aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin URL aplikasi.	Ketika status tugas telah diperbarui ke Running, pilih tugas. Di bagian Jaringan, salin IP publik.	Administrator sistem AWS, Pengembang aplikasi
Uji aplikasi Anda.	Di browser Anda, masukkan IP publik untuk menguji aplikasi.	Administrator sistem AWS, Pengembang aplikasi

Sumber daya terkait

- Dasar-Dasar Docker untuk Amazon ECS (dokumentasi Amazon ECS)
- Amazon ECS di AWS Fargate (dokumentasi Amazon ECS)
- Membuat Definisi Tugas (dokumentasi Amazon ECS)
- Membuat Cluster (dokumentasi Amazon ECS)
- Mengkonfigurasi Parameter Layanan Dasar (dokumentasi Amazon ECS)
- Mengkonfigurasi Jaringan (dokumentasi Amazon ECS)
- Menerapkan Java Microservices di Amazon ECS (posting blog)

Menerapkan resource dan paket Kubernetes menggunakan Amazon EKS dan repositori bagan Helm di Amazon S3

Sagar Panigrahi, Amazon Web Services

Ringkasan

Pola ini membantu Anda mengelola aplikasi Kubernetes secara efisien, terlepas dari kompleksitasnya. Pola ini mengintegrasikan Helm ke dalam pipeline continuous integration dan continuous delivery (CI/CD) yang ada untuk menyebarkan aplikasi ke dalam klaster Kubernetes. Helm adalah manajer paket Kubernetes yang membantu Anda mengelola aplikasi Kubernetes. Bagan helm membantu mendefinisikan, menginstal, dan meningkatkan aplikasi Kubernetes yang kompleks. Bagan dapat dibuat versi dan disimpan di repositori Helm, yang meningkatkan mean time to restore (MTTR) selama pemadaman.

Pola ini menggunakan Amazon Elastic Kubernetes Service (Amazon EKS) untuk klaster Kubernetes. Ini menggunakan Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) sebagai repositori bagan Helm, sehingga bagan dapat dikelola dan diakses secara terpusat oleh pengembang di seluruh organisasi.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) aktif dengan virtual private cloud (VPC)
- Klaster Amazon EKS
- Node pekerja disiapkan dalam klaster Amazon EKS dan siap menerima beban kerja
- Kubectl untuk mengonfigurasi file Amazon EKS kubeconfig untuk cluster target di mesin klien
- Akses AWS Identity and Access Management (IAM) untuk membuat bucket S3
- Akses IAM (terprogram atau peran) ke Amazon S3 dari mesin klien
- Manajemen kode sumber dan CI/CD pipa

Batasan

- Tidak ada dukungan saat ini untuk meningkatkan, menghapus, atau mengelola definisi sumber daya khusus ()CRDs.

- Jika Anda menggunakan sumber daya yang mengacu pada CRD, CRD harus diinstal secara terpisah (di luar bagan).

Versi produk

- Helm v3.6.3

Arsitektur

Tumpukan teknologi target

- Amazon EKS
- Amazon VPC
- Amazon S3
- Manajemen kode sumber
- Helm
- Kubectl

Arsitektur target

Otomatisasi dan skala

- AWS CloudFormation dapat digunakan untuk mengotomatiskan pembuatan infrastruktur. Untuk informasi selengkapnya, lihat Membuat resource Amazon EKS dengan AWS CloudFormation di dokumentasi Amazon EKS.
- Helm akan dimasukkan ke dalam alat CI/CD otomatisasi Anda yang ada untuk mengotomatiskan pengemasan dan pembuatan versi bagan Helm (di luar cakupan pola ini).
- GitVersion atau nomor build Jenkins dapat digunakan untuk mengotomatiskan pembuatan versi bagan.

Alat

Alat

- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) adalah layanan terkelola untuk menjalankan Kubernetes di AWS tanpa perlu berdiri atau memelihara pesawat kontrol Kubernetes Anda sendiri. Kubernetes adalah sebuah sistem sumber terbuka untuk melakukan otomatisasi terhadap deployment, penskalaan, dan pengelolaan aplikasi terkontainer.
- Helm - Helm adalah manajer paket untuk Kubernetes yang membantu Anda menginstal dan mengelola aplikasi di klaster Kubernetes Anda.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web.
- Kubectl — Kubectl adalah utilitas baris perintah untuk menjalankan perintah terhadap klaster Kubernetes.

Kode

Kode contoh terlampir.

Epik

Konfigurasikan dan inisialisasi Helm

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal klien Helm.	Untuk mengunduh dan menginstal klien Helm di sistem lokal Anda, gunakan perintah berikut. `sudo curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash`	DevOps insinyur
Validasi instalasi Helm.	Untuk memvalidasi bahwa Helm dapat berkomunikasi dengan server API Kubernetes	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di dalam klaster Amazon EKS, jalankan. `helm version`	

Membuat dan menginstal bagan Helm di kluster Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bagan Helm untuk NGINX.	Untuk membuat bagan helm bernama `my-nginx` pada mesin klien, jalankan`helm create my-nginx`.	DevOps insinyur
Tinjau struktur bagan.	Untuk meninjau struktur bagan, jalankan perintah pohon`tree my-nginx/` .	DevOps insinyur
Nonaktifkan pembuatan akun layanan di bagan.	Di`values.yaml` , di bawah `serviceAccount` bagian, atur `create` kunci ke`false`. Ini dimatikan karena tidak ada persyaratan untuk membuat akun layanan untuk pola ini.	DevOps insinyur
Validasi (lint) bagan yang dimodifikasi untuk kesalahan sintaksis.	Untuk memvalidasi bagan untuk kesalahan sintaksis apa pun sebelum menginstalnya di cluster target, jalankan. `helm lint my-nginx/`	DevOps insinyur
Instal bagan untuk menerapkan sumber daya Kubernetes.	Untuk menjalankan instalasi bagan Helm, gunakan perintah berikut. `helm install --name my-nginx-release --debug`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`my-nginx/ --namespace helm-space` `debug`Bendera opsional menampilkan semua pesan debug selama penginsta lan. `namespace` Bendera menentukan namespace di mana bagian sumber daya dari bagan ini akan dibuat.	
Tinjau sumber daya di cluster Amazon EKS.	Untuk meninjau sumber daya yang dibuat sebagai bagian dari bagan Helm di `helm-space` namespace, gunakan perintah berikut. `kubectl get all -n helm-space`	DevOps insinyur

Kembalikan ke versi aplikasi Kubernetes sebelumnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah dan tingkatkan rilis.	Untuk memodifikasi bagan, di`values.yaml` , ubah `replicaCount` nilainya menjadi2. Kemudian tingkatka n rilis yang sudah diinstal dengan menjalankan perintah berikut.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`helm upgrade my-nginx-release my-nginx/ --namespace helm-space`	
Tinjau sejarah rilis Helm.	Untuk membuat daftar semua revisi untuk rilis tertentu yang telah diinstal menggunak an Helm, jalankan perintah berikut. `helm history my-nginx-release`	DevOps insinyur
Tinjau detail untuk revisi tertentu.	Sebelum beralih atau memutar kembali ke versi kerja, dan untuk lapisan validasi tambahan sebelum menginsta l revisi, lihat nilai mana yang diteruskan ke setiap revisi dengan menggunakan perintah berikut. `helm get --revision=2 my-nginx-release`	DevOps insinyur
Gulung kembali ke versi sebelumnya.	Untuk memutar kembali ke revisi sebelumnya, gunakan perintah berikut. `helm rollback my-nginx-release 1` Contoh ini bergulir kembali ke nomor revisi 1.	DevOps insinyur

Inisialisasi bucket S3 sebagai repositori Helm

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 untuk bagan Helm.	Buat bucket S3 yang unik. Di ember, buat folder bernama`charts`. Contoh dalam pola ini digunakan `s3://my-helm-charts/charts` sebagai repositori grafik target.	Administrator awan
Instal plugin Helm untuk Amazon S3.	Untuk menginstal plugin helm-s3 di mesin klien Anda, gunakan perintah berikut. `helm plugin install https://github.com/hypnoglow/helm-s3.git --version 0.10.0` Catatan: Dukungan Helm V3 tersedia dengan plugin versi 0.9.0 dan di atas.	DevOps insinyur
Inisialisasi repositori Amazon S3 Helm.	Untuk menginisialisasi folder target sebagai repositori Helm, gunakan perintah berikut. `helm S3 init s3://my-helm-charts/charts` Perintah membuat `index.yaml` file di target untuk melacak semua informasi bagan yang disimpan di lokasi itu.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan repositori Amazon S3 ke Helm.	Untuk menambahkan repositori di mesin klien, gunakan perintah berikut. `helm repo add my-helm-charts s3://my-helm-charts/charts` Perintah ini menambahkan alias ke repositori target di mesin klien Helm.	DevOps insinyur
Tinjau daftar repositori.	Untuk melihat daftar repositori di mesin klien Helm, jalankan. `helm repo list`	DevOps insinyur

Package and store chart di repositori Amazon S3 Helm

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Package grafik.	Untuk mengemas `my-nginx` bagan yang Anda buat, jalankan`helm package ./my-nginx/` . Perintah mengemas semua isi folder `my-nginx` bagan ke dalam file arsip, yang diberi nama menggunakan nomor versi yang disebutkan dalam `Chart.yaml` file.	DevOps insinyur
Simpan paket di repositori Amazon S3 Helm.	Untuk mengunggah paket ke repositori Helm di Amazon S3, jalankan perintah berikut,	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggunakan nama file yang benar. `.tgz` `helm s3 push ./my-nginx-0.1.0.tgz my-helm-charts`	
Cari bagan Helm.	Untuk mengonfirmasi bahwa bagan muncul baik secara lokal maupun di repositori Helm di Amazon S3, jalankan perintah berikut. `helm search repo my-nginx`	DevOps insinyur

Ubah, versi, dan paket bagan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memodifikasi dan mengemas bagan.	Dalam`values.yaml` , atur `replicaCount` nilainya ke`1`. Kemudian kemas bagan dengan menjalank anhelm package ./ my-nginx/ , kali ini mengubah versi `Chart.yaml` menjadi`0.1.1`. Pembuatan versi idealnya diperbarui melalui otomatisasi menggunakan alat seperti GitVersion atau Jenkins membangun nomor dalam pipeline. CI/CD Mengotoma	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tiskan nomor versi berada di luar cakupan pola ini.	
Dorong versi baru ke repositori Helm di Amazon S3.	Untuk mendorong paket baru dengan versi 0.1.1 ke repositori `my-helm-charts` Helm di Amazon S3, jalankan perintah berikut. `helm s3 push ./my-nginx-0.1.1.tgz my-helm-charts`	DevOps insinyur

Cari dan instal bagan dari repositori Amazon S3 Helm

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Cari semua versi bagan my-nginx.	Untuk melihat semua versi bagan yang tersedia, jalankan perintah berikut dengan `--versions` bendera. `helm search repo my-nginx --versions` Tanpa tanda, Helm secara default menampilkan versi bagan terbaru yang diunggah.	DevOps insinyur
Instal bagan dari repositori Amazon S3 Helm.	Hasil pencarian dari tugas sebelumnya menunjukkan beberapa versi `my-nginx` bagan. Untuk menginstal versi baru (0.1.1) dari repositori	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon S3 Helm, gunakan perintah berikut. `helm upgrade my-nginx-release my-helm-charts/my-nginx --version 0.1.1 --namespace helm-space`	

Sumber daya terkait

- Dokumentasi HELM
- plugin helm-s3 (Lisensi MIT)
- Biner klien HELM
- Dokumentasi Amazon EKS

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menerapkan cluster CockroachDB di Amazon EKS dengan menggunakan Terraform

Sandip Gangapadhyay dan Kalyan Senthilnathan, Amazon Web Services

Ringkasan

Pola ini menyediakan modul HashiCorp Terraform untuk menerapkan cluster CockroachDB multi-node di Amazon Elastic Kubernetes Service (Amazon EKS) dengan menggunakan operator CockroachDB. CockroachDB adalah database SQL terdistribusi yang menyediakan sharding horizontal otomatis, ketersediaan tinggi, dan kinerja yang konsisten di seluruh cluster yang didistribusikan secara geografis. Pola ini menggunakan Amazon EKS sebagai platform Kubernetes yang dikelola dan mengimplementasikan cert-manager untuk komunikasi node yang diamankan TLS. Ini juga menggunakan Network Load Balancer untuk distribusi lalu lintas dan menciptakan CockroachDB StatefulSetsdengan pod yang secara otomatis mereplikasi data untuk toleransi kesalahan dan kinerja.

Audiens yang dituju

Untuk menerapkan pola ini, kami sarankan Anda terbiasa dengan hal-hal berikut:

- HashiCorp Konsep dan infrastruktur Terraform sebagai praktik kode (IAc)
- Layanan AWS, khususnya Amazon EKS
- Dasar-dasar Kubernetes, termasuk, operator StatefulSets, dan konfigurasi layanan
- Database SQL terdistribusi
- Konsep keamanan, seperti manajemen sertifikat TLS.
- DevOps praktik, CI/CD alur kerja, dan otomatisasi infrastruktur

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Izin untuk menyebarkan sumber daya di klaster Amazon EKS
- Cluster Amazon EKS versi v1.23 atau yang lebih baru, dengan node berlabel `node=cockroachdb`
- Amazon Elastic Block Store Container Storage Interface (CSI) Driver versi 1.19.0 atau yang lebih baru, diinstal di cluster Amazon EKS

- Terraform CLI versi 1.0.0 atau yang lebih baru, diinstal
- kubectl, diinstal
- Git, diinstal
- AWS Command Line Interface (AWS CLI) versi 2.9.18 atau yang lebih baru, diinstal dan dikonfigurasi

Batasan

- Operator CockroachDB Kubernetes tidak mendukung beberapa klaster Kubernetes untuk penerapan Multi-region. Untuk batasan lebih lanjut, lihat Orchestrate CockroachDB Across Multiple Kubernetes Clusters (dokumentasi CockroachDB) dan CockroachDB Kubernetes Operator (). GitHub
- Pemangkasan otomatis klaim volume persisten (PVCs) saat ini dinonaktifkan secara default. Ini berarti bahwa setelah menonaktifkan dan menghapus node, operator tidak akan menghapus volume persisten yang dipasang ke podnya. Untuk informasi lebih lanjut, lihat Pemangkasan PVC otomatis dalam dokumentasi CockroachDB.

Versi produk

- CockroachDB versi 22.2.2

Arsitektur

Arsitektur target

Diagram berikut menunjukkan penyebaran CockroachDB yang sangat tersedia di tiga AWS Availability Zone dalam virtual private cloud (VPC). Pod CockroachDB dikelola melalui Amazon EKS. Arsitektur menggambarkan bagaimana pengguna mengakses database melalui Network Load Balancer, yang mendistribusikan lalu lintas ke pod CockroachDB. Pod berjalan di instans Amazon Elastic Compute Cloud (Amazon EC2) di setiap Availability Zone, yang memberikan ketahanan dan toleransi kesalahan.

Sumber daya dibuat

Menerapkan modul Terraform yang digunakan dalam pola ini menciptakan sumber daya berikut:

1. Network Load Balancer - Sumber daya ini berfungsi sebagai titik masuk untuk permintaan klien dan mendistribusikan lalu lintas secara merata di seluruh instance CockroachDB.
2. CockroachDB StatefulSet — StatefulSet Mendefinisikan keadaan yang diinginkan dari penyebaran CockroachDB dalam cluster Amazon EKS. Ini mengelola penyebaran, penskalaan, dan pembaruan pod CockroachDB yang dipesan.
3. Pod CockroachDB — Pod ini adalah contoh dari CockroachDB yang berjalan sebagai kontainer di dalam pod Kubernetes. Pod ini menyimpan dan mengelola data di seluruh cluster terdistribusi.
4. Database CockroachDB — Ini adalah database terdistribusi yang dikelola oleh CockroachDB, mencakup beberapa pod. Ini mereplikasi data untuk ketersediaan tinggi, toleransi kesalahan, dan kinerja.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.

Alat-alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- kubectl adalah antarmuka baris perintah yang membantu Anda menjalankan perintah terhadap klaster Kubernetes.

Repositori kode

Kode untuk pola ini tersedia di cluster GitHub Deploy a CockroachDB di Amazon EKS menggunakan repositori Terraform. Repositori kode berisi file dan folder berikut untuk Terraform:

- `modules`folder - Folder ini berisi modul Terraform untuk CockroachDB
- `main`folder - Folder ini berisi modul root yang memanggil modul anak CockroachDB untuk membuat cluster database CockroachDB.

Praktik terbaik

- Jangan mengurangi hingga kurang dari tiga node. Ini dianggap sebagai anti-pola pada CockroachDB dan dapat menyebabkan kesalahan. Untuk informasi selengkapnya, lihat Penskalaan Cluster dalam dokumentasi CockroachDB.
- Terapkan penskalaan otomatis Amazon EKS dengan menggunakan Karpernter atau Cluster Autoscaler. Hal ini memungkinkan cluster CockroachDB untuk skala horizontal dan node baru secara otomatis. Untuk informasi selengkapnya, lihat Komputasi klaster skala dengan Karpenter dan Cluster Autoscaler di dokumentasi Amazon EKS.

> **Note**
>
> Karena aturan penjadwalan `podAntiAffinity` Kubernetes, hanya satu pod CockroachDB yang dapat dijadwalkan dalam satu node Amazon EKS.

- Untuk praktik terbaik keamanan Amazon EKS, lihat Praktik Terbaik untuk Keamanan dalam dokumentasi Amazon EKS.
- Untuk praktik terbaik kinerja SQL untuk CockroachDB, lihat Praktik Terbaik Kinerja SQL dalam dokumentasi CockroachDB.
- Untuk informasi selengkapnya tentang menyiapkan backend jarak jauh Amazon Simple Storage Service (Amazon S3) untuk file status Terraform, lihat Amazon S3 di dokumentasi Terraform.

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode.	Masukkan perintah berikut untuk mengkloning repositori: `git clone https://github.com/aws-samples/crdb-cluster-eks-terraform.git`	DevOps insinyur, Git

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui variabel Terraform.	1. Masukkan perintah berikut untuk menavigasi ke folder utama di repositori kloning: `cd crdb-cluster-eks-terraform/main` 2. Buka file variabel.tf. 3. Konfigurasikan nilai default untuk variabel berikut: • `region`— Masukkan target Wilayah AWS • `eks_cluster_name` — Masukkan nama untuk kluster Amazon EKS target • `number_of_nodes` — Masukkan jumlah node yang akan digunakan 4. Simpan dan tutup file variabel.tf.	DevOps insinyur, Terraform

Menyebarkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan infrastruktur.	1. Masukkan perintah berikut untuk menginisialisasi penerapan Terraform: `terraform init`	DevOps insinyur, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Masukkan perintah berikut untuk menghasilkan rencana eksekusi:
	`terraform plan`
	3. Tinjau rencana, dan validasi sumber daya dan komponen infrastruktur yang akan dibuat.
	4. Masukkan perintah berikut untuk menyebarkan infrastruktur:
	`terraform apply`
	5. Saat diminta, masukkan ya untuk mengonfirmasi penerapan.
	6. Tunggu hingga penerapan selesai.	

Verifikasi penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi pembuatan sumber daya.	1. Masukkan perintah berikut untuk mengatur konteks Amazon EKS menggunakan AWS CLI: `aws eks update-kubeconfig –name <eks_cluster_name>`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Masukkan perintah berikut untuk memverifikasi jumlah pod yang menggunakan CockroachDB:
	`kubectl get pods -n <namespace>`
	Berikut ini adalah contoh output.
	`NAME READY STATUS RESTARTS AGE`
	`cockroach-operator-655fbf7847-zn9v8 1/1 Running 0 30m`
	`cockroachdb-0 1/1 Running 0 24m`
	`cockroachdb-1 1/1 Running 0 24m`
	`cockroachdb-2 1/1 Running 0 24m`
	3. Verifikasi bahwa jumlah pod sesuai dengan nilai yang Anda tentukan dalam file variable.tf.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Skala naik atau turun.	1. Dalam file variabel.tf, tambah atau kurangi jumlah node dan kemudian simpan file. 2. Ulangi langkah-langkah untuk menerapkan infrastru ktur melalui Terraform. Terraform menambahkan atau menghapus pod. 3. Ulangi langkah-langkah untuk memverifikasi jumlah pod yang menggunakan Cockroach DB. Misalnya, jika Anda meningkatkan jumlah node dari tiga menjadi empat, Anda sekarang akan melihat empat pod berjalan.	DevOps insinyur, Terraform

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus infrastruktur.	Menskalakan node ke `0` akan mengurangi biaya komputasi. Namun, Anda masih akan dikenakan biaya untuk volume Amazon EBS persisten yang dibuat oleh modul ini. Untuk menghilan gkan biaya penyimpanan, ikuti langkah-langkah berikut untuk menghapus semua volume:	Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Masukkan perintah berikut untuk menghapus infrastru ktur: `terraform destroy` 2. Saat diminta, masukkan ya untuk mengonfirmasi.	

Pemecahan masalah

Isu	Solusi
Kesalahan memvalidasi kredensyal penyedia	Saat Anda menjalankan Terraform `apply` atau `destroy` perintah, Anda mungkin mengalami kesalahan berikut: `Error: configuring Terraform AWS Provider: error validating provider credentials: error calling sts:GetCallerIdentity: operation error STS: GetCallerIdentity, https response error StatusCode: 403, RequestID: 123456a9-fbc1-40ed-b8d8-513d0133ba7f, api error InvalidClientTokenId: The security token included in the request is invalid.` Kesalahan ini disebabkan oleh kedaluwarsa token keamanan untuk kredensil yang digunakan dalam konfigurasi mesin lokal Anda. Untuk petunjuk tentang cara mengatasi kesalahan, lihat Mengatur dan melihat

Isu	Solusi
	pengaturan konfigurasi dalam AWS CLI dokumentasi.
Pod CockroachDB dalam status tertunda	1. Karena aturan penjadwalan `podAntiAffinity` Kubernetes, hanya satu pod CockroachDB yang dapat dijadwalkan dalam satu node Amazon EKS. Jika jumlah pod CockroachDB melebihi jumlah node Amazon EKS yang tersedia, maka pod Cockroach DB mungkin dalam status tertunda. Dalam hal ini, Anda perlu menerapkan Cluster Autoscaler atau Karpenter sehingga skala node Amazon EKS secara otomatis. Untuk informasi selengkapnya, lihat Komputasi klaster skala dengan Karpenter dan Cluster Autoscaler. 2. Periksa apakah node pekerja Kubernetes memiliki `node=cockroachdb` label dengan memasukkan perintah berikut: `kubectl get nodes --show-labels` Jika tidak, Anda perlu memastikan bahwa semua node pekerja diberi label dengan benar. Untuk informasi selengkapnya, lihat Mengedit konfigurasi grup node dalam dokumentasi Amazon EKS.

Sumber daya terkait

- Terapkan CockroachDB dalam Cluster Kubernetes Tunggal (dokumentasi CockroachDB)
- Mengatur CockroachDB di Beberapa Kluster Kubernetes (dokumentasi CockroachDB)
- AWS Penyedia (dokumentasi Terraform)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menerapkan contoh layanan mikro Java di Amazon EKS dan mengekspos layanan mikro menggunakan Application Load Balancer

Vijay Thompson dan Akkamahadevi Hiremath, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerapkan contoh layanan mikro Java sebagai aplikasi kontainer di Amazon Elastic Kubernetes Service (Amazon EKS) dengan menggunakan `eksctl` utilitas baris perintah dan Amazon Elastic Container Registry (Amazon ECR). Anda dapat menggunakan Application Load Balancer untuk memuat keseimbangan lalu lintas aplikasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI) versi 1.7 atau yang lebih baru, diinstal dan dikonfigurasi di macOS, Linux, atau Windows
- Daemon Docker yang sedang berjalan
- Utilitas baris `eksctl` perintah, diinstal dan dikonfigurasi di macOS, Linux, atau Windows (Untuk informasi selengkapnya, lihat Memulai Amazon EKS — eksctl dalam dokumentasi Amazon EKS.)
- Utilitas baris `kubectl` perintah, diinstal dan dikonfigurasi di macOS, Linux, atau Windows (Untuk informasi selengkapnya, lihat Menginstal atau memperbarui kubectl dalam dokumentasi Amazon EKS.)

Batasan

- Pola ini tidak mencakup pemasangan sertifikat SSL untuk Application Load Balancer.

Arsitektur

Tumpukan teknologi target

- Amazon ECR
- Amazon EKS
- Penyeimbang Beban Elastis

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk containerizing Java microservice di Amazon EKS.

Alat

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Elastic Load Balancing secara otomatis mendistribusikan lalu lintas masuk ke beberapa target, seperti instans Amazon Elastic Compute Cloud (EC2Amazon), container, dan alamat IP, dalam satu atau beberapa Availability Zone.
- eksctl membantu Anda membuat cluster di Amazon EKS.
- kubectl memungkinkan untuk menjalankan perintah terhadap cluster Kubernetes.
- Docker membantu Anda membangun, menguji, dan mengirimkan aplikasi dalam paket yang disebut kontainer.

Epik

Buat klaster Amazon EKS dengan menggunakan eksctl

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat klaster Amazon EKS.	Untuk membuat klaster Amazon EKS yang menggunakan dua EC2 instance Amazon t2.small sebagai node, jalankan perintah berikut: `eksctl create cluster --name <your-cluster-name> --version <version-`	Pengembang, Admin Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`number> --nodes=1 --node-type=t2.small` **Note** Prosesnya bisa memakan waktu antara 15 hingga 20 menit. Setelah cluster dibuat, konfigurasi Kubernetes yang sesuai ditambahkan ke file kubeconfig Anda. Anda dapat menggunakan `kubeconfig` file dengan `kubectl` untuk menyebarkan aplikasi di langkah selanjutnya.	
Verifikasi cluster Amazon EKS.	Untuk memverifikasi bahwa cluster dibuat dan Anda dapat terhubung dengannya, jalankan `kubectl get nodes` perintah.	Pengembang, Admin Sistem

Buat repositori Amazon ECR dan dorong gambar Docker.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR.	Ikuti petunjuk dari Membuat repositori pribadi di dokumentasi Amazon ECR.	Pengembang, Admin Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file XMLPOM.	Buat `pom.xml` file berdasark an kode file Contoh POM di bagian Informasi tambahan dari pola ini.	Pengembang, Admin Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file sumber.	Buat file sumber yang disebut `HelloWorld.java` di `src/main/java/eksExample` jalur berdasarkan contoh berikut:
	`package eksExample; import static spark.Spark.get; public class HelloWorld { public static void main(String[] args) { get("/", (req, res) -> { return "Hello World!"; }); } }`
	Pastikan untuk menggunakan struktur direktori berikut:
	`### Dockerfile ### deployment.yaml ### ingress.yaml ### pom.xml ### service.yaml ### src ### main ### java ### eksExample ### HelloWorld.java`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Dockerfile.	Buat `Dockerfile` berdasarkan kode Contoh Dockerfile di bagian Informasi tambahan dari pola ini.	Pengembang, Admin Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bangun dan dorong gambar Docker.	Di direktori tempat Anda `Dockerfile` ingin membangun, menandai, dan mendorong gambar ke Amazon ECR, jalankan perintah berikut:
	`aws ecr get-login -password --region <region>| docker login --username <username> --password-stdin <account_number>.dkr.ecr.<region>.amazonaws.com`
	`docker buildx build --platform linux/amd64 -t hello-world-java:v1 .`
	`docker tag hello-world-java:v1 <account_number>.dkr.ecr.<region>.amazonaws.com/<repository_name>:v1`
	`docker push <account_number>.dkr.ecr.<region>.amazonaws.com/<repository_name>:v1`
	Note
	Ubah Wilayah AWS, nomor akun, dan detail repositori di perintah sebelumnya. Pastikan untuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mencatat URL gambar untuk digunakan nanti. **Important** Sistem macOS dengan chip M1 memiliki masalah dalam membangun gambar yang kompatibel dengan Amazon EKS yang berjalan pada platform. AMD64 Untuk mengatasi masalah ini, gunakan docker buildx untuk membuat image Docker yang berfungsi di Amazon EKS.	

Menyebarkan layanan mikro Java

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file penyebaran.	Buat file YAMM yang disebut `deployment.yaml` berdasarkan kode file penyebaran Contoh di bagian Informasi tambahan dari pola ini.	Pengembang, Admin Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Gunakan URL gambar yang Anda salin sebelumnya sebagai jalur file gambar untuk repositori Amazon ECR.	
Menerapkan layanan mikro Java di kluster Amazon EKS.	Untuk membuat penerapan di cluster Amazon EKS Anda, jalankan `kubectl apply -f deployment.yaml` perintah.	Pengembang, Admin Sistem
Verifikasi status pod.	1. Untuk memverifikasi status pod, jalankan `kubectl get pods` perintah. 2. Tunggu status berubah menjadi Siap.	Pengembang, Admin Sistem
Buat sebuah layanan.	1. Buat file yang disebut `service.yaml` berdasarkan kode file layanan Contoh di bagian Informasi tambahan dari pola ini. 2. Jalankan perintah `kubectl apply -f service.yaml .`	Pengembang, Admin Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal add-on AWS Load Balancer Controller.	Ikuti petunjuk dari Menginstal add-on AWS Load Balancer Controller dalam dokumentasi Amazon EKS. **Note** Anda harus memiliki add-on yang diinstal untuk membuat Application Load Balancer atau Network Load Balancer untuk layanan Kubernetes.	Devloper, Sistem Admin
Buat sumber daya ingress.	Buat file YAMM yang disebut `ingress.yaml` berdasarkan kode file sumber daya masuk Contoh di bagian Informasi tambahan dari pola ini.	Pengembang, Admin Sistem
Buat Application Load Balancer.	Untuk menyebarkan sumber daya ingress dan membuat Application Load Balancer, jalankan perintah. `kubectl apply -f ingress.yaml`	Pengembang, Admin Sistem

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji dan verifikasi aplikasi.	1. Untuk mendapatkan nama DNS load balancer dari bidang ADDRESS, jalankan	Pengembang, Admin Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	perintah. `kubectl get ingress.networking.k8s.io/java-microservice-ingress` 2. Pada EC2 instance di VPC yang sama dengan node Amazon EKS Anda, jalankan perintah. `curl -v <DNS address from previous command>`	

Sumber daya terkait

- Membuat repositori pribadi (dokumentasi Amazon ECR)
- Mendorong gambar Docker (dokumentasi Amazon ECR)
- Pengontrol Ingress (Lokakarya Amazon EKS)
- Docker buildx (dokumen Docker)

Informasi tambahan

Contoh file POM

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>


  <groupId>helloWorld</groupId>
  <artifactId>helloWorld</artifactId>
  <version>1.0-SNAPSHOT</version>


  <dependencies>
```

```
    <dependency>
      <groupId>com.sparkjava</groupId><artifactId>spark-core</
artifactId><version>2.0.0</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId><artifactId>maven-jar-plugin</
artifactId><version>2.4</version>
        <configuration><finalName>eksExample</finalName><archive><manifest>
              <addClasspath>true</addClasspath><mainClass>eksExample.HelloWorld</
mainClass><classpathPrefix>dependency-jars/</classpathPrefix>
            </manifest></archive>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId><artifactId>maven-compiler-plugin</
artifactId><version>3.1</version>
        <configuration><source>1.8</source><target>1.8</target></configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId><artifactId>maven-assembly-plugin</
artifactId>
        <executions>
          <execution>
            <goals><goal>attached</goal></goals><phase>package</phase>
            <configuration>
              <finalName>eksExample</finalName>
              <descriptorRefs><descriptorRef>jar-with-dependencies</descriptorRef></
descriptorRefs>
              <archive><manifest><mainClass>eksExample.HelloWorld</mainClass></
manifest></archive>
            </configuration>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</project>
```

Contoh Dockerfile

```
FROM bellsoft/liberica-openjdk-alpine-musl:17

RUN apk add maven
WORKDIR /code

# Prepare by downloading dependencies
ADD pom.xml /code/pom.xml
RUN ["mvn", "dependency:resolve"]
RUN ["mvn", "verify"]

# Adding source, compile and package into a fat jar
ADD src /code/src
RUN ["mvn", "package"]

EXPOSE 4567
CMD ["java", "-jar", "target/eksExample-jar-with-dependencies.jar"]
```

Contoh berkas penyebaran

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: microservice-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app.kubernetes.io/name: java-microservice
  template:
    metadata:
      labels:
        app.kubernetes.io/name: java-microservice
    spec:
      containers:
      - name: java-microservice-container
        image: .dkr.ecr.amazonaws.com/:
        ports:
        - containerPort: 4567
```

Contoh file layanan

```
apiVersion: v1
```

```
kind: Service
metadata:
  name: "service-java-microservice"
spec:
  ports:
    - port: 80
      targetPort: 4567
      protocol: TCP
  type: NodePort
  selector:
    app.kubernetes.io/name: java-microservice
```

Contoh file sumber daya ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: "java-microservice-ingress"
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/load-balancer-name: apg2
    alb.ingress.kubernetes.io/target-type: ip
  labels:
    app: java-microservice
spec:
  rules:
    - http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: "service-java-microservice"
                port:
                  number: 80
```

Menerapkan aplikasi yang dikelompokkan ke Amazon ECS dengan menggunakan AWS Copilot

Jean-Baptiste Guillois, Mathew George, dan Thomas Scott, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menerapkan container di kluster Amazon Elastic Container Service (Amazon ECS) dengan dua cara—dengan menggunakan Konsol Manajemen Amazon Web Services (AWS), dan dengan menggunakan AWS Copilot—untuk mendemonstrasikan cara AWS Copilot menyederhanakan tugas penerapan.

Amazon ECS adalah layanan manajemen kontainer yang sangat skalabel dan cepat yang memudahkan untuk menjalankan, menghentikan, dan mengelola kontainer di cluster. kontainer Anda dijelaskan pada ketentuan tugas yang Anda gunakan untuk menjalankan tugas individu atau tugas yang ada dalam layanan. Anda dapat menjalankan tugas dan layanan Anda pada infrastruktur tanpa server yang dikelola oleh AWS Fargate. Atau, untuk kontrol lebih lanjut atas infrastruktur, Anda dapat menjalankan tugas dan layanan Anda di klaster instans Amazon Elastic Compute Cloud (Amazon EC2) yang Anda kelola.

Perintah AWS Copilot command line interface (CLI) menyederhanakan pembuatan, pelepasan, dan pengoperasian aplikasi kontainer siap produksi di Amazon ECS dari lingkungan pengembangan lokal. AWS Copilot CLI selaras dengan alur kerja pengembang yang mendukung praktik terbaik aplikasi modern: mulai dari menggunakan infrastruktur sebagai kode hingga membuat pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang disediakan atas nama pengguna. Anda dapat menggunakan AWS Copilot CLI sebagai bagian dari siklus pengembangan dan pengujian sehari-hari Anda sebagai alternatif untuk AWS Management Console.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi secara lokal untuk menggunakan akun AWS Anda (lihat petunjuk penginstalan dan instruksi konfigurasi dalam dokumentasi AWS CLI)
- AWS Copilot diinstal secara lokal (lihat petunjuk penginstalan di dokumentasi Amazon ECS)
- Docker diinstal pada mesin lokal Anda (lihat dokumentasi Docker)

Batasan

- Docker memberlakukan batas tarik 100 gambar kontainer per 6 jam per alamat IP pada paket gratis.

Arsitektur

Tumpukan teknologi target

- Lingkungan AWS diatur dengan virtual private cloud (VPC), subnet publik dan pribadi, dan grup keamanan
- Kluster Amazon ECS
- Layanan Amazon ECS dan definisi tugas
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon DynamoDB
- Penyeimbang Beban Aplikasi
- AWS Fargate
- Amazon Identity and Access Management (IAM)
- Amazon CloudWatch
- AWS CloudTrail

Arsitektur target

Saat Anda menerapkan aplikasi sampel untuk pola ini, beberapa tugas dibuat dan diterapkan di Availability Zone terpisah. Setiap tugas menyimpan data di Amazon DynamoDB. Saat Anda mengakses halaman web untuk suatu tugas, Anda dapat melihat data dari semua tugas lainnya.

Alat

Layanan AWS

- Amazon ECR — Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola AWS yang aman, terukur, dan andal. Amazon ECR mendukung repositori pribadi dengan izin berbasis sumber daya menggunakan IAM.

- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat untuk menjalankan, menghentikan, dan mengelola kontainer di klaster. Anda dapat menjalankan tugas dan layanan Anda pada infrastruktur tanpa server yang dikelola oleh AWS Fargate. Atau, untuk kontrol lebih lanjut atas infrastruktur, Anda dapat menjalankan tugas dan layanan Anda di klaster instans Amazon Elastic Compute Cloud (Amazon EC2) yang Anda kelola.
- AWS Copilot — AWS Copilot menyediakan antarmuka baris perintah yang membantu Anda meluncurkan dan mengelola aplikasi kontainer di AWS, termasuk mendorong ke registri, membuat definisi tugas, dan membuat klaster.
- AWS Fargate — AWS Fargate adalah mesin pay-as-you-go komputasi tanpa server yang memungkinkan Anda fokus membangun aplikasi tanpa mengelola server. AWS Fargate kompatibel dengan Amazon ECS dan Amazon Elastic Kubernetes Service (Amazon EKS). Saat menjalankan tugas dan layanan Amazon ECS dengan tipe peluncuran Fargate atau penyedia kapasitas Fargate, Anda mengemas aplikasi dalam wadah, menentukan persyaratan CPU dan memori, menentukan kebijakan jaringan dan IAM, dan meluncurkan aplikasi. Setiap tugas Fargate memiliki batas isolasi sendiri dan tidak berbagi kernel yang mendasarinya, sumber daya CPU, sumber daya memori, atau elastic network interface dengan tugas lain.
- Amazon DynamoDB - Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi dengan skalabilitas yang mulus.
- Elastic Load Balancing (ELB) — Elastic Load Balancing secara otomatis mendistribusikan lalu lintas masuk Anda ke beberapa target, EC2 seperti instance, kontainer, dan alamat IP, dalam satu atau beberapa Availability Zone. Ini memantau kesehatan target terdaftarnya, dan mengarahkan lalu lintas hanya ke target yang sehat. Elastic Load Balancing menskalakan load balancer Anda saat lalu lintas masuk Anda berubah seiring waktu. Hal ini dapat secara otomatis menskalakan sebagian besar beban kerja.

Alat

- Antarmuka Baris Perintah Docker
- AWS Command Line Interface (AWS CLI)
- Antarmuka baris perintah AWS Copilot

Kode

Kode untuk aplikasi sampel yang digunakan dalam pola ini tersedia di GitHub, di repositori Aplikasi Sampel Cluster. Ikuti petunjuk di bagian selanjutnya untuk menggunakan file sampel.

Epik

Terapkan tumpukan aplikasi - opsi 1 (AWS Management Console)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Kloning repositori kode sampel dengan menggunakan perintah: `git clone https://github.com/aws-samples/cluster-sample-app cluster-sample-app && cd cluster-sample-app`	Pengembang aplikasi, AWS DevOps
Buat repositori Amazon ECR Anda.	1. Masuk ke AWS Management Console dan buka konsol Amazon ECR di https://console.aws.amazon.com/ecr/repositori. 2. Pilih Buat repositori. 3. Untuk nama repositori, masukkan. cluster-sample-app 4. Untuk semua pengaturan lainnya, pertahankan nilai default. 5. Pilih Buat repositori. Untuk informasi selengkapnya, lihat Membuat repositor	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	i pribadi di dokumentasi Amazon ECR.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat, tag, dan dorong image Docker Anda ke repositori Amazon ECR Anda.	1. Pilih repositori yang baru saja Anda buat dan pilih Lihat perintah push.
	2. Salin perintah yang ditampilkan dan jalankan secara lokal untuk membangun, menandai, dan mendorong image docker Anda. Perintah ini akan mirip dengan yang berikut ini.
	
	Untuk mengautentikasi klien Docker Anda ke registri:
	
	`aws ecr get-login -password --region <YOUR_AWS_REGION> | docker login --username AWS --password-stdin <YOUR_AWS_ACCOUNT>.dkr.ecr.<YOUR_AWS_REGION>.amazonaws.com`
	
	Untuk membangun image Docker Anda:
	
	`docker build -t cluster-sample-app .`
	
	Untuk menandai gambar Docker Anda:
	
	`docker tag cluster-sample-app:latest`
		Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<YOUR_AWS_ACCOUNT> .dkr.ecr.<YOUR_AWS _REGION>.amazonaws .com/cluster-sample- app:latest Untuk mendorong image Docker ke repositori Anda: docker push <YOUR_AWS _ACCOUNT>.dkr.ecr. <YOUR_AWS_REGION>. amazonaws.com/clus ter-sample-app:latest	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan tumpukan aplikasi.	1. Buka CloudFormation konsol AWS di https://console.aws.amazon.com/cloudformation/. 2. Pilih Buat tumpukan. 3. Di bagian Siapkan templat, pilih Template sudah siap. 4. Di bagian Tentukan templat, pilih Unggah file templat. 5. Pilih file lokal **`cluster-sample-app-stack.yml`** yang Anda kloning dari GitHub repositori sebagai CloudFormation template, dan kemudian pilih Berikutnya. 6. Masukkan nama untuk tumpukan Anda, lalu pilih Berikutnya. 7. Simpan semua opsi default, lalu pilih Berikutnya. 8. Tinjau semua opsi, akui pembuatan sumber daya IAM, lalu pilih Buat tumpukan. 9. Ketika tumpukan aplikasi Anda telah digunakan, pilih tab Output, salin URL, dan buka di browser Anda untuk mengakses aplikasi.	AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang penerapan CloudFormation templat, lihat Membuat tumpukan di CloudFormation dokumentasi AWS.	

Terapkan tumpukan aplikasi - opsi 2 (AWS Copilot CLI)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Kloning repositori kode sampel dengan menggunakan perintah: `git clone https://github.com/aws-samples/cluster-sample-app cluster-sample-app && cd cluster-sample-app`	Pengembang aplikasi, AWS DevOps
Terapkan image container Anda ke AWS dengan menggunakan AWS Copilot CLI.	Terapkan aplikasi dalam satu langkah dengan menggunakan perintah berikut di direktori root proyek Anda: `copilot init --app cluster-sample-app --name demo --type "Load Balanced Web Service" --dockerfile ./Dockerfile --port 8080 --deploy` Anda kemudian harus dapat mengakses aplikasi dengan	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggunakan nama DNS yang disediakan sebagai output.	

Hapus sumber daya yang dibuat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya yang dibuat melalui AWS Management Console.	Jika Anda menggunakan opsi 1 (AWS Management Console) untuk menerapkan tumpukan aplikasi, ikuti langkah-langkah berikut saat Anda siap menghapus sumber daya yang Anda buat: 1. Buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation/. 2. Pilih tumpukan yang Anda buat, lalu pilih Hapus. 3. Buka konsol Amazon ECR di https://console.aws.amazon.com/ecr/repositori. 4. Pilih repositori yang Anda buat, lalu pilih Hapus.	Pengembang aplikasi, AWS DevOps
Hapus sumber daya yang dibuat oleh AWS Copilot.	Jika Anda menggunakan opsi 2 (AWS Copilot CLI) untuk menyebarkan tumpukan aplikasi, jalankan perintah berikut dari direktori root	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	proyek Anda saat Anda siap untuk menghapus sumber daya yang Anda buat: `copilot app delete`	

Sumber daya terkait

- Menginstal atau memperbarui versi terbaru AWS CLI (dokumentasi AWS CLI)
- Menggunakan antarmuka baris perintah AWS Copilot (dokumentasi Amazon ECS)
- Amazon ECS di AWS Fargate (dokumentasi Amazon ECR)
- Dokumentasi Amazon ECS
- Dokumentasi Amazon ECR
- CloudFormation Dokumentasi Amazon
- Docker Desktop (dokumentasi Docker)

Menerapkan aplikasi berbasis GRPC di cluster Amazon EKS dan mengaksesnya dengan Application Load Balancer

Kirankumar Chandrashekar dan Huy Nguyen, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara meng-host aplikasi berbasis GRPC di cluster Amazon Elastic Kubernetes Service (Amazon EKS) dan mengaksesnya dengan aman melalui Application Load Balancer.

gRPC adalah kerangka kerja panggilan prosedur jarak jauh sumber terbuka (RPC) yang dapat berjalan di lingkungan apa pun. Anda dapat menggunakannya untuk integrasi layanan mikro dan komunikasi client-server. Untuk informasi selengkapnya tentang gRPC, lihat postingan blog AWS dukungan Application Load Balancer untuk end-to-end HTTP/2 dan gRPC.

Pola ini menunjukkan cara meng-host aplikasi berbasis GRPC yang berjalan pada pod Kubernetes di Amazon EKS. Klien gRPC terhubung ke Application Load Balancer melalui protokol HTTP/2 dengan koneksi terenkripsi. SSL/TLS Application Load Balancer meneruskan lalu lintas ke aplikasi gRPC yang berjalan pada pod Amazon EKS. Jumlah pod gRPC dapat secara otomatis diskalakan berdasarkan traffic dengan menggunakan Kubernetes Horizontal Pod Autoscaler. Kelompok target Application Load Balancer melakukan pemeriksaan kesehatan pada node Amazon EKS, mengevaluasi apakah targetnya sehat, dan meneruskan lalu lintas hanya ke node yang sehat.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Docker, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- eksctl, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- `kubectl`, diinstal dan dikonfigurasi untuk mengakses sumber daya di kluster Amazon EKS Anda. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui kubectl di dokumentasi Amazon EKS.
- g RPCurl, diinstal dan dikonfigurasi.
- Cluster Amazon EKS baru atau yang sudah ada. Untuk informasi selengkapnya, lihat Memulai Amazon EKS.

- Terminal komputer Anda dikonfigurasi untuk mengakses kluster Amazon EKS. Untuk informasi selengkapnya, lihat Mengonfigurasi komputer untuk berkomunikasi dengan klaster Anda di dokumentasi Amazon EKS.
- AWS Load Balancer Controller, disediakan di klaster Amazon EKS.
- Nama host DNS yang ada dengan SSL atau SSL/TLS sertifikat yang valid. Anda dapat memperoleh sertifikat untuk domain Anda dengan menggunakan AWS Certificate Manager (ACM) atau mengunggah sertifikat yang sudah ada ke ACM. Untuk informasi selengkapnya tentang kedua opsi ini, lihat Meminta sertifikat publik dan Mengimpor sertifikat ke AWS Certificate Manager dalam dokumentasi ACM.

Arsitektur

Diagram berikut menunjukkan arsitektur yang diterapkan oleh pola ini.

Diagram berikut menunjukkan alur kerja di mana SSL/TLS lalu lintas diterima dari klien gRPC yang diturunkan ke Application Load Balancer. Lalu lintas diteruskan dalam plaintext ke server gRPC karena berasal dari virtual private cloud (VPC).

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Elastic Load Balancing mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.

Alat

- eksctl adalah alat CLI sederhana untuk membuat cluster di Amazon EKS.
- kubectl adalah utilitas baris perintah untuk menjalankan perintah terhadap klaster Kubernetes.
- AWS Load Balancer Controller membantu Anda mengelola AWS Elastic Load Balancer untuk klaster Kubernetes.
- g RPCurl adalah alat baris perintah yang membantu Anda berinteraksi dengan layanan gRPC.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub grpc-traffic-on-alb-to-eks.

Epik

Buat dan dorong image Docker server gRPC ke Amazon ECR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR.	Masuk ke AWS Management Console, buka konsol Amazon ECR, lalu buat repositori Amazon ECR. Untuk informasi selengkapnya, lihat Membuat repositori di dokumentasi Amazon ECR. Pastikan Anda merekam URL repositori Amazon ECR. Anda juga dapat membuat repositori Amazon ECR dengan AWS CLI dengan menjalankan perintah berikut: `aws ecr create-repository --repository-name helloworld-grpc`	Administrator awan
membuat gambar Docker.	1. Kloning repositori GitHub grpc-traffic-on-alb-to-eks.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`git clone https://github.com/aws-samples/grpc-traffic-on-alb-to-eks.git`
	2. Dari direktori root repositori, pastikan bahwa Dockerfile ada dan kemudian jalankan perintah berikut untuk membangun image Docker:
	`docker build -t <amazon_ecr_repository_url>:<Tag> .`
	Important
	Pastikan Anda mengganti `<amazon_ecr_repository_url>` dengan URL repositori Amazon ECR yang Anda buat sebelumnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Dorong gambar Docker ke Amazon ECR.	1. Jalankan perintah berikut untuk masuk ke repositori Amazon ECR:
	`aws ecr get-login -password --region us-east-1 --no-cli-auto-prompt | docker login --username AWS --password-stdin <your_aws_account_id>.dkr.ecr.us-east-1.amazonaws.com`
	2. Dorong image Docker ke repositori Amazon ECR dengan menjalankan perintah berikut:
	`docker push <your_aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/helloworld-grpc:1.0`
	⚠ Important
	Pastikan Anda mengganti `<your_aws_account_id>` dengan ID akun AWS Anda.
		DevOps insinyur

Menerapkan manifes Kubernetes ke cluster Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah nilai dalam file manifes Kubernetes.	1. Ubah file `grpc-samp le.yaml` manifes Kubernetes di folder Kubernetes dari repositori sesuai dengan kebutuhan Anda. Anda harus memodifikasi anotasi dan nama host di sumber daya ingress. Untuk sumber daya masuk sampel, lihat bagian Informasi tambahan. Untuk informasi selengkapnya tentang anotasi ingress, lihat anotasi Ingress di dokumentasi Kubernetes. 2. Di resource deployment Kubernetes, ubah resource deployment menjadi uniform resource identifie r (URI) `image` untuk repositori Amazon ECR tempat Anda mendorong image Docker. Untuk sumber daya penerapan contoh, lihat bagian Informasi tambahan.	DevOps insinyur
Menerapkan file manifes Kubernetes.	Menerapkan `grpc-samp le.yaml` file ke cluster Amazon EKS dengan menjalankan `kubectl` perintah berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`kubectl apply -f ./kubernetes/grpc-sample.yaml`	

Buat catatan DNS untuk FQDN Application Load Balancer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Catat FQDN untuk Application Load Balancer.	1. Jalankan `kubectl` perintah berikut untuk mendeskripsikan sumber daya ingress Kubernetes yang mengelola Application Load Balancer: `kubectl get ingress -n grpcserver` Output sampel disediakan di bagian Informasi tambahan. Dalam output, `HOSTS` bidang menampilkan nama host DNS tempat sertifikat SSL dibuat. 2. Catat nama domain (FQDN) Application Load Balancer yang sepenuhnya memenuhi syarat dari `Address` bidang output. 3. Buat catatan DNS yang menunjuk ke FQDN Application Load Balancer. Jika penyedia DNS Anda adalah Amazon Route 53, Anda dapat	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membuat catatan alias yang menunjuk ke FQDN Application Load Balancer. Untuk informasi selengkapnya tentang opsi ini, lihat Memilih antara catatan alias dan non-alias dalam dokumentasi Route 53.	

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji server gRPC.	Gunakan g RPCurl untuk menguji titik akhir dengan menjalankan perintah berikut: `grpcurl grpc.example.com:443 list grpc.reflection.v1alpha.ServerReflection helloworld.helloworld` **Note** Ganti `grpc.example.com` dengan nama DNS Anda.	DevOps insinyur
Uji server gRPC menggunakan klien gRPC.	Dalam `helloworld_client_ssl.py` contoh klien gRPC, ganti nama host dari `grpc.example.com` dengan nama host yang	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	digunakan untuk server gRPC.
	
	Contoh kode berikut menunjukkan respon dari server gRPC untuk permintaan klien:
	
	`python ./app/helloworld_client_ssl.py`
	`message: "Hello to gRPC server from Client"`
	
	`message: "Thanks for talking to gRPC server!! Welcome to hello world. Received message is \"Hello to gRPC server from Client\""`
	`received: true`
	
	Ini menunjukkan bahwa klien dapat berbicara dengan server dan koneksi berhasil.	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus catatan DNS.	Hapus catatan DNS yang menunjuk ke FQDN Application Load Balancer yang Anda buat sebelumnya.	Administrator awan
Lepaskan penyeimbang beban.	Di EC2 konsol Amazon, pilih Load Balancers, lalu hapus	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	load balancer yang dibuat oleh controller Kubernetes untuk resource ingress Anda.	
Hapus kluster Amazon EKS.	Hapus kluster Amazon EKS dengan menggunak an`eksctl`: `eksctl delete cluster -f ./eks.yaml`	AWS DevOps

Sumber daya terkait

- Penyeimbangan beban jaringan di Amazon EKS
- Grup sasaran untuk Application Load Balancers

Informasi tambahan

Sumber daya masuknya sampel:

```
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    alb.ingress.kubernetes.io/healthcheck-protocol: HTTP
    alb.ingress.kubernetes.io/ssl-redirect: "443"
    alb.ingress.kubernetes.io/backend-protocol-version: "GRPC"
    alb.ingress.kubernetes.io/listen-ports: '[{"HTTP": 80}, {"HTTPS":443}]'
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/target-type: ip
    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:<AWS-
Region>:<AccountId>:certificate/<certificate_ID>
  labels:
    app: grpcserver
    environment: dev
  name: grpcserver
```

```
  namespace: grpcserver
spec:
  ingressClassName: alb
  rules:
  - host: grpc.example.com # <----- replace this as per your host name for which the
 SSL certtficate is available in ACM
    http:
      paths:
      - backend:
          service:
            name: grpcserver
            port:
              number: 9000
        path: /
        pathType: Prefix
```

Sumber daya penyebaran sampel:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grpcserver
  namespace: grpcserver
spec:
  selector:
    matchLabels:
      app: grpcserver
  replicas: 1
  template:
    metadata:
      labels:
        app: grpcserver
    spec:
      containers:
      - name: grpc-demo
        image: <your_aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/helloworld-
grpc:1.0   #<------- Change to the URI that the Docker image is pushed to
        imagePullPolicy: Always
        ports:
        - name: grpc-api
          containerPort: 9000
        env:
        - name: POD_IP
```

```
          valueFrom:
            fieldRef:
              fieldPath: status.podIP
      restartPolicy: Always
```

Keluaran sampel:

```
NAME             CLASS           HOSTS                          Address
 PORTS          AGE
 grpcserver       <none>      <DNS-HostName>                 <ELB-address>
 80             27d
```

Menerapkan aplikasi kontainer saat AWS IoT Greengrass V2 berjalan sebagai wadah Docker

Salih Bakir, Giuseppe Di Bella, dan Gustav Svalander, Amazon Web Services

Ringkasan

AWS IoT Greengrass Version 2, saat diterapkan sebagai wadah Docker, secara native tidak mendukung menjalankan wadah aplikasi Docker. Pola ini menunjukkan kepada Anda cara membuat gambar kontainer khusus berdasarkan versi terbaru AWS IoT Greengrass V2 yang mengaktifkan fungsionalitas Docker-in-Docker (DiND). Dengan DiND, Anda dapat menjalankan aplikasi kontainer dalam lingkungan. AWS IoT Greengrass V2

Anda dapat menerapkan pola ini sebagai solusi yang berdiri sendiri atau mengintegrasikannya dengan platform orkestrasi kontainer seperti Amazon ECS Anywhere. Di salah satu model penerapan, Anda mempertahankan AWS IoT Greengrass V2 fungsionalitas penuh termasuk kemampuan pemrosesan AWS IoT SiteWise Edge, sekaligus mengaktifkan penerapan berbasis container yang dapat diskalakan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Untuk AWS IoT Greengrass Version 2 prasyarat umum, lihat Prasyarat dalam dokumentasi. AWS IoT Greengrass Version 2
- Docker Engine, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- Docker Compose (jika Anda menggunakan antarmuka baris perintah Docker Compose (CLI) untuk menjalankan image Docker).
- Sebuah sistem operasi Linux.
- Hypervisor dengan server host yang mendukung virtualisasi.
- Persyaratan sistem:
  - 2 GB RAM (minimum)
  - 5 GB ruang disk yang tersedia (minimum)
  - Untuk AWS IoT SiteWise Edge, CPU quad-core x86_64 dengan RAM 16 GB dan ruang disk yang tersedia 50 GB. Untuk informasi selengkapnya tentang pemrosesan AWS IoT SiteWise data, lihat Persyaratan paket pemrosesan data dalam AWS IoT SiteWise dokumentasi.

Versi produk

- AWS IoT Greengrass Version 2 versi 2.5.3 atau yang lebih baru
- Docker-in-Docker versi 1.0.0 atau yang lebih baru
- Docker Compose versi 1.22 atau yang lebih baru
- Docker Engine versi 20.10.12 atau yang lebih baru

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Tumpukan teknologi target

- Sumber data — Perangkat IoT, sensor, atau peralatan industri yang menghasilkan data untuk diproses
- AWS IoT Greengrass V2— Berjalan sebagai wadah Docker dengan D-in-D kemampuan, diterapkan pada infrastruktur tepi
- Aplikasi kontainer - Aplikasi khusus yang berjalan di dalam AWS IoT Greengrass V2 lingkungan sebagai wadah Docker bersarang
- (Opsional) Amazon ECS Anywhere — Orkestrasi kontainer yang mengelola penerapan kontainer AWS IoT Greengrass V2
- Lainnya Layanan AWS - AWS IoT Core, AWS IoT SiteWise, dan lainnya Layanan AWS untuk pemrosesan dan manajemen data

Arsitektur target

Diagram berikut menunjukkan contoh arsitektur penerapan target yang menggunakan Amazon ECS Anywhere, yang merupakan alat manajemen kontainer.

Diagram menunjukkan alur kerja berikut:

1: Penyimpanan gambar kontainer - Amazon ECR menyimpan gambar AWS IoT Greengrass kontainer dan wadah aplikasi khusus apa pun yang diperlukan untuk pemrosesan tepi.

2 dan 3: Penerapan kontainer — Amazon ECS Anywhere menyebarkan AWS IoT Greengrass image container dari Amazon ECR ke lokasi edge, mengelola siklus hidup kontainer dan proses penerapan.

4: Penerapan komponen — AWS IoT Greengrass Inti yang digunakan secara otomatis menyebarkan komponen yang relevan berdasarkan konfigurasinya. Komponen termasuk AWS IoT SiteWise Edge dan komponen pemrosesan tepi lain yang diperlukan dalam lingkungan kontainer.

5: Penyerapan data — Setelah sepenuhnya dikonfigurasi, AWS IoT Greengrass mulailah menelan telemetri dan data sensor dari berbagai sumber data IoT di lokasi tepi.

6: Pemrosesan data dan integrasi cloud — AWS IoT Greengrass Inti kontainer memproses data secara lokal menggunakan komponen yang digunakan (termasuk AWS IoT SiteWise Edge untuk data industri). Kemudian, ia mengirimkan data yang diproses ke AWS Cloud layanan untuk analisis dan penyimpanan lebih lanjut.

Alat

Layanan AWS

- Amazon ECS Anywhere membantu Anda menerapkan, menggunakan, dan mengelola tugas dan layanan Amazon ECS di infrastruktur Anda sendiri.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- AWS IoT Greengrass adalah waktu aktif edge Internet untuk Segala (IoT) sumber terbuka dan layanan cloud yang membantu Anda membangun, men-deploy, dan mengelola aplikasi IoT pada perangkat Anda.
- AWS IoT SiteWisemembantu Anda mengumpulkan, memodelkan, menganalisis, dan memvisualisasikan data dari peralatan industri dalam skala besar.

Alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.

- Docker Compose adalah alat untuk mendefinisikan dan menjalankan aplikasi multi-container.
- Docker Engine adalah teknologi kontainerisasi open source untuk membangun dan mengkontainerisasi aplikasi.

Repositori kode

Kode untuk pola ini tersedia di Docker-in-Docker repositori GitHub AWS IoT Greengrass v2.

Epik

Bangun AWS IoT Greengrass V2 Docker-in-Docker gambar

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning dan arahkan ke repositori.	Untuk mengkloning repositori, gunakan perintah berikut:
	`git clone https://github.com/aws-samples/aws-iot-greengrass-v2-docker-in-docker.git`
	Untuk menavigasi ke `docker` direktori, gunakan perintah berikut:
	`cd aws-iot-greengrass-v2-docker-in-docker/docker`
		DevOps insinyur, AWS DevOps

membuat gambar Docker.	Untuk membuat image Docker dengan versi default (terbaru), jalankan perintah berikut: `docker build -t x86_64/aws-iot-greengrass:latest .`	AWS DevOps, DevOps insinyur, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Atau, untuk membangun image Docker dengan versi tertentu, jalankan perintah berikut:
	`docker build --build-arg GREENGRASS_RELEASE_VERSION=2.12.0 -t x86_64/aws-iot-greengrass:2.12.0 .`
	Untuk memverifikasi build, jalankan perintah berikut:
	`docker images | grep aws-iot-greengrass`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

(Opsional) Dorong ke Amazon ECR.	1. Untuk membuat repositori, jalankan perintah berikut:
	`aws ecr create-repository --repository-name aws-iot-greengrass-dind --region us-east-1`
	2. Untuk mengautentikasi, jalankan perintah berikut:
	`aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin <ACCOUNT-ID>.dkr.ecr.us-east-1.amazonaws.com`
	3. Untuk menandai dan mendorong, jalankan perintah berikut:
	`docker tag x86_64/aws-iot-greengrass:latest <ACCOUNT-ID>.dkr.ecr.us-east-1.amazonaws.com/aws-iot-greengrass-dind:latest`
	`docker push <ACCOUNT_ID>.dkr.ecr.us-east-1.amazonaws.com/aws-iot-g`
		Pengembang aplikasi, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`reengrass-dind:latest`	

Konfigurasikan AWS kredensional

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih metode otentikasi.	Pilih salah satu opsi berikut: • Opsi 1 (Disarankan): Peran IAM - Gunakan jika berjalan di Amazon EC2, Amazon ECS, atau Amazon EKS dengan peran IAM yang tepat. Tidak diperlukan konfigurasi tambahan. • Opsi 2: Variabel lingkungan - Untuk pengujian dan pengembangan. • Opsi 3: File kredensial - Tidak direkomendasikan untuk produksi. • Opsi 4: `env.cfg` File lama.	Administrator AWS
Konfigurasikan metode otentikasi.	Untuk metode otentikasi yang Anda pilih, gunakan panduan konfigurasi berikut: • Opsi 2 (Variabel lingkungan) - Mempersiapkan`AWS_ACCESS_KEY_ID` ,`AWS_SECRET_ACCESS_KEY` , dan secara opsional`AWS_SESSION_TOKEN` .	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Opsi 3 (File kredensial) - Buat direktori dan file kredensil:
	  • (a) Jalankan perintah berikut:
	  `mkdir -p ~/greengrass-v2-credentials`
	  • (b) Buat file berikut di`~/greengrass-v2-credentials/credentials` :
	  `[default]`
	  `aws_access_key_id=YourAccessKey`
	  `aws_secret_access_key=YourSecretKey`
	  • (c) Amankan file sebagai berikut:
	  `chmod 600 ~/greengrass-v2-credentials/credentials`
	• Opsi 4 (**`env.cfg`**) - Buat `env.cfg` file di `docker` direktori sebagai berikut:
	  `GGC_ROOT_PATH=/greengrass/v2`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`AWS_REGION=us-east-1`
	`PROVISION=true`
	`THING_NAME=MyGreengrassCore`
	`THING_GROUP_NAME=MyGreengrassCoreGroup`
	`TES_ROLE_NAME=GreengrassV2TokenExchangeRole`
	`TES_ROLE_ALIAS_NAME=GreengrassCoreTokenExchangeRoleAlias`
	`COMPONENT_DEFAULT_USER=ggc_user:ggc_group`	

Jalankan dengan Docker Compose

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasi `docker-compose.yml` .	Perbarui `docker-compose.yml` file dengan variabel lingkungan sebagai berikut: • Variabel kunci: `PROVISION` , `AWS_REGIO`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`N ,THING_NAM E ,THING_GRO UP_NAME ,TES_ROLE_ NAME , TES_ROLE_ ALIAS_NAME`
	• Untuk kredensi melalui variabel lingkungan, hapus komentar berikut ini:
	`environment:`
	`- AWS_ACCESS_KEY_ID= YourKey`
	`- AWS_SECRET_ACCESS_ KEY=YourSecret`
	• Untuk file kredensial, hapus komentar berikut ini:
	`volumes:`
	`- ~/greengrass-v2-cr edentials:/root/.a ws/:ro`
	• Untuk persistensi log, tambahkan yang berikut ini:
	`volumes:`
	`- ./logs:/greengrass /v2/logs`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulai dan verifikasi wadah.	Untuk memulai di latar depan, jalankan perintah berikut:
	`docker-compose up --build`
	Atau, untuk memulai di latar belakang, jalankan perintah berikut:
	`docker-compose up --build -d`
	Untuk memverifikasi status, jalankan perintah berikut:
	`docker-compose ps`
	Untuk memantau log, jalankan perintah berikut:
	`docker-compose logs -f`
		DevOps insinyur

Jalankan dengan Docker CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kontainer dengan Docker CLI.	• Untuk memulai proses dasar tanpa penyediaan, jalankan perintah berikut: `docker run --init --privileged -it --name aws-iot-greengrass x86_64/aw`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`s-iot-greengrass:l atest`
	
	• Untuk memulai menjalankan dengan variabel lingkungan, jalankan perintah berikut:
	
	`docker run --init --privileged - it --name aws- iot-greengrass - e PROVISION=true - e AWS_REGION=us- east-1 -e AWS_ACCES S_KEY_ID=YourKey - e AWS_SECRET_ACCESS_ KEY=YourSecret -e THING_NAME=MyGreen grassCore -e THING_GROUP_NAME=M yGreengrassCoreGro up -p 8883:8883 x86_64/aws-iot-gre engrass:latest`
	
	• Untuk memulai proses dengan file kredensial, jalankan perintah berikut:
	
	`docker run --init --privileged -it --name aws-iot-g reengrass -v ~/ greengrass-v2-cr edentials:/root/.a`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`ws:ro -e PROVISION =true -e AWS_REGIO N=us-east-1 -e THING_NAME=MyGreen grassCore -p 8883:8883 x86_64/aw s-iot-greengrass:l atest`
	• Untuk memulai run dengan `env.cfg` file, jalankan perintah berikut:
	`docker run --init --privileged -it --name aws-iot-g reengrass -v ~/ greengrass-v2-cr edentials:/root/.a ws:ro --env-file env.cfg -p 8883:8883 x86_64/aws-iot-gre engrass:latest`
	• Untuk memulai proses dengan log persisten, jalankan perintah berikut:
	`docker run --init --privileged - it --name aws-iot- greengrass -v \ $(pwd)/logs:/gree ngrass/v2/logs`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`x86_64/aws-iot-greengrass:latest`	

Verifikasi wadah.	• Untuk memeriksa status kontainer, jalankan perintah berikut:
	`docker ps | grep aws-iot-greengrass`
	• Untuk memantau log, jalankan perintah berikut:
	`docker logs -f aws-iot-greengrass`
		DevOps insinyur

Kelola aplikasi kontainer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan aplikasi.	• Untuk menyebarkan wadah aplikasi, jalankan perintah berikut:
	`docker run -d --name my-app nginx:alpine`
	• Untuk memverifikasi penerapan, jalankan perintah berikut:
	`docker ps`
	• Untuk mengakses log, jalankan perintah berikut:
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`cd /greengrass/v2/logs && cat greengrass.log`	

Akses dan uji Docker-in-Docker.	1. Untuk membuka shell dalam wadah, jalankan perintah berikut:
	`docker exec -it aws-iot-greengrass sh`
	2. Untuk memverifikasi Docker, jalankan perintah berikut:
	`docker --version`
	`docker ps`
	3. Untuk menguji Docker-in-Docker, jalankan perintah berikut:
	`docker run --rm hello-world`
		DevOps insinyur

(Opsional) Integrasikan dengan Amazon ECS Anywhere

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan cluster Amazon ECS.	• Untuk membuat cluster Amazon ECS, jalankan perintah berikut: `aws ecs create-cluster --cluster-name greengrass-`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cluster --region us-east-1`
	
	• Untuk membuat aktivasi SSM, jalankan perintah berikut:
	
	`aws ssm create-activation --default-instance-name greengrass-edge --iam-role GreengrassECSAnywhereRole --registration-limit 10 --region us-east-1`
	
	• Untuk menginstal Agen SSM pada perangkat edge, jalankan perintah berikut:
	
	`curl -o install.sh https://s3.amazonaws.com/aws-ssm-downloads/latest/linux_amd64/install.sh`
	
	`chmod +x install.sh`
	
	`./install.sh activation-code activation-id us-east-1`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menerapkan tugas Amazon ECS.	• Buat `greengrass-task-definition.json` dengan konfigurasi tugas termasuk `containerDefinitions` dengan `privileged:true`, variabel lingkungan, dan volume mount untuk `/var/lib/docker` dan `/greengrass/v2`. Register tugas. Jalankan perintah berikut:
	
	`aws ecs register-task-definition --cli-input-json file://greengrass-task-definition.json`
	
	• Untuk membuat layanan, jalankan perintah berikut:
	
	`aws ecs create-service --cluster greengrass-cluster --service-name greengrass-service --task-definition greengrass-dind --desired-count 1 --launch-type EXTERNAL --region us-east-1`
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk memverifikasi penerapan, jalankan perintah berikut: `aws ecs describe-services --cluster greengrass-cluster --services greengrass-service --region us-east-1`	

Berhenti dan bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hentikan wadah.	• Untuk menghentikan container menggunakan Docker Compose, jalankan perintah berikut:
	`docker-compose stop`
	`docker-compose down`
	`docker-compose down -v`
	• Untuk menghentikan penampung (menghapus volume) dengan menggunakan Docker CLI, jalankan perintah berikut:
	`docker stop aws-iot-greengrass`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`docker rm aws-iot-greengrass`
	• Untuk menghentikan wadah dengan menggunakan force remove, jalankan perintah berikut:
	`docker rm -f aws-iot-greengrass`	

Pemecahan masalah

Isu	Solusi

Penampung gagal memulai dengan kesalahan izin.	1. Untuk memastikan bahwa container berjalan dengan `--privileged` flag yang diperlukan Docker-in-Docker, jalankan perintah berikut:
	`docker run --init --privileged -it --name aws-iot-greengrass x86_64/aws-iot-greengrass:latest`
	2. Dalam `docker-compose.yml` file, atur yang berikut ini:
	`privileged: true`

Isu	Solusi
	⚠ Warning `--privileged` memberikan hak istimewa yang diperluas ke wadah.

Penyediaan gagal dengan kesalahan kredensi.	Untuk memverifikasi kredenal dikonfigurasi dengan benar, gunakan langkah-langkah berikut:
	1. Periksa peran IAM:
	`aws sts get-caller-identity`
	2. Verifikasi variabel lingkungan:
	`docker exec -it aws-iot-greengrass sh -c 'echo \$AWS_ACCESS_KEY_ID'`
	3. Periksa file kredensional:
	`docker exec -it aws-iot-greengrass sh -c 'cat /root/.aws/credentials'`
	Pastikan bahwa izin IAM mencakup`iot:CreateThing` ,,, `iot:CreatePolicy` `iot:AttachPolicy` `iam:CreateRole` , dan. `iam:AttachRolePolicy`

Isu	Solusi

Tidak dapat terhubung ke daemon Docker di dalam wadah.	1. Periksa status daemon Docker:
	`docker exec -it aws-iot-greengrass sh -c 'ps aux | grep dockerd'`
	2. Verifikasi bahwa soket Docker ada:
	`docker exec -it aws-iot-greengrass sh -c 'ls -la /var/run/docker.sock'`
	3. Mulai secara manual jika diperlukan:
	`docker exec -it aws-iot-greengrass sh -c 'dockerd > /var/log/docker.log 2>&1 &'`

Kontainer kehabisan ruang disk.	1. Periksa ruang disk:
	`df -h`
	2. Bersihkan sumber daya Docker:
	`docker system prune -a --volumes`
	3. Hapus artefak Greengrass lama:
	`docker exec -it aws-iot-greengrass sh -c 'rm -rf /greengrass/v2/work/*  '`
	Pastikan ruang disk minimum: 5 GB untuk operasi dasar dan 50 GB untuk AWS IoT SiteWise Edge

Isu	Solusi

Membangun masalah.	1. Pastikan Anda berada di `docker` direktori:
	`cd docker && pwd`
	2. Verifikasi yang `Dockerfile` ada:
	`ls -la Dockerfile`
	3. Periksa apakah Docker diinstal dan dijalankan:
	`docker --version && docker ps`

Masalah konektivitas jaringan.	1. Uji resolusi DNS:
	`docker exec -it aws-iot-greengrass sh -c 'nslookup google.com'`
	2. Uji AWS konektivitas:
	`docker exec -it aws-iot-greengrass sh -c 'curl https://iot.us-east-1.amazonaws.com'`
	Verifikasi bahwa firewall memungkinkan lalu lintas HTTPS (443) dan MQTT (8883) keluar.

Isu	Solusi

Komponen Greengrass gagal diterapkan.	1. Periksa log Greengrass:
	`docker exec -it aws-iot-greengrass sh -c 'cat /greengrass/v2/logs/greengrass.log'`
	2. Verifikasi konfigurasi peran TES:
	`aws iot describe-role-alias --role-alias TES_ROLE_ALIAS_NAME`
	Periksa log khusus komponen di direktori. `/greengrass/v2/logs/`

Kontainer keluar segera setelah memulai.	1. Periksa log kontainer untuk kesalahan:
	`docker logs aws-iot-greengrass`
	2. Jalankan dengan shell interaktif untuk debugging:
	`docker run --init --privileged -it --entrypoint sh x86_64/aws-iot-greengrass:latest`
	Verifikasi semua variabel lingkungan yang diperlukan diatur dengan benar jika`PROVISION=true` . Pastikan `--init` bendera digunakan saat memulai wadah.

Sumber daya terkait

AWS sumber daya

- Layanan Kontainer Elastis Amazon
- Konfigurasikan pemrosesan data tepi untuk AWS IoT SiteWise model dan aset

- Apa itu AWS IoT Greengrass

Sumber daya lainnya

- Dokumentasi Docker

Informasi tambahan

- Untuk pemrosesan data AWS IoT SiteWise Edge, Docker harus tersedia di AWS IoT Greengrass lingkungan.
- Untuk menjalankan kontainer bersarang, Anda harus menjalankan AWS IoT Greengrass kontainer dengan kredensi tingkat administrator.

Menyebarkan wadah dengan menggunakan Elastic Beanstalk

Thomas Scott dan Jean-Baptiste Guillois, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, AWS Elastic Beanstalk mendukung Docker sebagai platform yang tersedia, sehingga container dapat berjalan dengan lingkungan yang dibuat. Pola ini menunjukkan cara menggunakan kontainer menggunakan layanan Elastic Beanstalk. Penyebaran pola ini akan menggunakan lingkungan server web berdasarkan platform Docker.

Untuk menggunakan Elastic Beanstalk untuk menyebarkan dan menskalakan aplikasi dan layanan web, Anda mengunggah kode dan penyebaran ditangani secara otomatis. Penyediaan kapasitas, penyeimbangan beban, penskalaan otomatis, dan pemantauan kesehatan aplikasi juga disertakan. Saat Anda menggunakan Elastic Beanstalk, Anda dapat mengambil kendali penuh atas sumber daya AWS yang dibuatnya atas nama Anda. Tidak ada biaya tambahan untuk Elastic Beanstalk. Anda hanya membayar untuk sumber daya AWS yang digunakan untuk menyimpan dan menjalankan aplikasi Anda.

Pola ini mencakup instruksi untuk penerapan menggunakan AWS Elastic Beanstalk Command Line Interface (EB CLI) dan AWS Management Console.

Kasus penggunaan

Kasus penggunaan untuk Elastic Beanstalk meliputi yang berikut:

- Menerapkan lingkungan prototipe untuk mendemonstrasikan aplikasi frontend. (Pola ini menggunakan Dockerfile sebagai contoh.)
- Menerapkan API untuk menangani permintaan API untuk domain tertentu.
- Terapkan solusi orkestrasi menggunakan Docker-Compose (tidak digunakan sebagai contoh praktis dalam pola `docker-compose.yml` ini).

Prasyarat dan batasan

Prasyarat

- Akun AWS
- AWS EB CLI diinstal secara lokal
- Docker diinstal pada mesin lokal

Batasan

- Ada batas tarik Docker 100 tarikan per 6 jam per alamat IP pada paket gratis.

Arsitektur

Tumpukan teknologi target

- Instans Amazon Elastic Compute Cloud (Amazon EC2)
- Grup keamanan
- Penyeimbang Beban Aplikasi
- Grup Auto Scaling

Arsitektur target

Otomatisasi dan skala

AWS Elastic Beanstalk dapat secara otomatis menskalakan berdasarkan jumlah permintaan yang dibuat. Sumber daya AWS yang dibuat untuk lingkungan mencakup satu Application Load Balancer, grup Auto Scaling, dan satu atau beberapa instans Amazon. EC2

Load balancer berada di depan EC2 instans Amazon, yang merupakan bagian dari grup Auto Scaling. EC2 Auto Scaling Amazon secara otomatis memulai EC2 instans Amazon tambahan untuk mengakomodasi peningkatan beban pada aplikasi Anda. Jika pemuatan pada aplikasi Anda berkurang, Amazon EC2 Auto Scaling menghentikan instance, tetapi setidaknya satu instance tetap berjalan.

Pemicu penskalaan otomatis

Grup Auto Scaling di lingkungan Elastic Beanstalk Anda menggunakan dua CloudWatch alarm Amazon untuk memulai operasi penskalaan. Pemicu default menskalakan ketika lalu lintas jaringan keluar rata-rata dari setiap instans lebih tinggi dari 6 MB atau lebih rendah dari 2 MB selama periode lima menit. Untuk menggunakan Amazon EC2 Auto Scaling secara efektif, konfigurasikan pemicu yang sesuai untuk aplikasi, jenis instans, dan persyaratan layanan Anda. Anda dapat menskalakan berdasar beberapa statistik termasuk latensi, I/O disk, utilisasi CPU, dan jumlah permintaan. Untuk informasi selengkapnya, lihat Pemicu Auto Scaling.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS EB Command Line Interface (EB CLI) adalah klien baris perintah yang dapat Anda gunakan untuk membuat, mengonfigurasi, dan mengelola lingkungan Elastic Beanstalk.
- Elastic Load Balancing mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone.

Layanan lainnya

- Docker mengemas perangkat lunak ke dalam unit standar yang disebut kontainer yang mencakup pustaka, alat sistem, kode, dan runtime.

Kode

Kode untuk pola ini tersedia di repositori Aplikasi Sampel GitHub Cluster.

Epik

Membangun dengan Dockerfile

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori jarak jauh.	• Untuk mengkloning repositori, jalankan perintah. `git clone https://github.com/aws-samples/cluster-sample-app.git < /p >`	Pengembang aplikasi, administrator AWS, AWS DevOps
Inisialisasi proyek Elastic Beanstalk Docker.	1. Buat file yang disebut `aws.json` di root. 2. Dalam `aws.json` file, tambahkan kode berikut.	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>{
	"AWSEBDoc
	kerrunVersion":"1",
	"Image":{
	"Name":"c
	luster-sample-app"
	},
	"Ports":[
	{
	"ContainerPort":80
	,
	"HostPort":8080
	}
	]
	}</pre> 3. Jalankan perintah `eb init -p docker` di root proyek.	

Uji proyek secara lokal.	1. Jalankan perintah `eb local run` di root proyek. 2. Uji aplikasi dengan menavigasi ke. `http://localhost`	Pengembang aplikasi, administrator AWS, AWS DevOps

Terapkan menggunakan EB CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan perintah penerapan	1. Jalankan perintah `eb create docker-sample-cluster-app` di root proyek.	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses versi yang digunakan.	Setelah perintah penerapan selesai, akses proyek menggunakan `eb open` perintah.	Pengembang aplikasi, administrator AWS, AWS DevOps

Terapkan menggunakan konsol

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan aplikasi dengan menggunakan browser.	1. Buka konsol . 2. Arahkan ke konsol Elastic Beanstalk. 3. Pilih Buat Aplikasi. 4. Untuk Nama Aplikasi, masukkan Cluster-Sample-App. 5. Pilih Docker sebagai platform. 6. Pilih Unggah kode Anda. 7. Pilih file.zip lokal Anda (di root proyek kloning) atau URL Simple Storage Service Amazon (Amazon S3) publik.	Pengembang aplikasi, administrator AWS, AWS DevOps
Akses versi yang digunakan.	Setelah penyebaran, akses aplikasi yang digunakan, dan pilih URL yang disediakan.	Pengembang aplikasi, administrator AWS, AWS DevOps

Sumber daya terkait

- Lingkungan server web

- Instal EB CLI di macOS
- Instal CLI EB secara manual

Informasi tambahan

Keuntungan menggunakan Elastic Beanstalk

- Penyediaan infrastruktur otomatis
- Manajemen otomatis platform yang mendasarinya
- Penambalan dan pembaruan otomatis untuk mendukung aplikasi
- Penskalaan otomatis aplikasi
- Kemampuan untuk menyesuaikan jumlah node
- Kemampuan untuk mengakses komponen infrastruktur jika diperlukan
- Kemudahan penyebaran di atas solusi penyebaran kontainer lainnya

Hasilkan alamat IP keluar statis menggunakan fungsi Lambda, Amazon VPC, dan arsitektur tanpa server

Thomas Scott, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara membuat alamat IP keluar statis di Amazon Web Services (AWS) Cloud dengan menggunakan arsitektur tanpa server. Organisasi Anda dapat memperoleh manfaat dari pendekatan ini jika ingin mengirim file ke entitas bisnis terpisah dengan menggunakan Secure File Transfer Protocol (SFTP). Ini berarti bahwa entitas bisnis harus memiliki akses ke alamat IP yang memungkinkan file melalui firewall-nya.

Pendekatan pola membantu Anda membuat fungsi AWS Lambda yang menggunakan alamat IP Elastis sebagai alamat IP keluar. Dengan mengikuti langkah-langkah dalam pola ini, Anda dapat membuat fungsi Lambda dan virtual private cloud (VPC) yang merutekan lalu lintas keluar melalui gateway internet dengan alamat IP statis. Untuk menggunakan alamat IP statis, Anda melampirkan fungsi Lambda ke VPC dan subnetnya.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Izin AWS Identity and Access Management (IAM) untuk membuat dan menerapkan fungsi Lambda, serta membuat VPC dan subnetnya. Untuk informasi selengkapnya tentang ini, lihat Peran eksekusi dan izin pengguna dalam dokumentasi AWS Lambda.
- Jika Anda berencana menggunakan infrastruktur sebagai kode (IAc) untuk menerapkan pendekatan pola ini, Anda memerlukan lingkungan pengembangan terintegrasi (IDE) seperti AWS Cloud9. Untuk informasi selengkapnya tentang ini, lihat Apa itu AWS Cloud9? dalam dokumentasi AWS Cloud9.

Arsitektur

Diagram berikut menunjukkan arsitektur tanpa server untuk pola ini.

Diagram menunjukkan alur kerja berikut:

1. Lalu lintas keluar `NAT gateway 1` masuk`Public subnet 1`.
2. Lalu lintas keluar `NAT gateway 2` masuk`Public subnet 2`.
3. Fungsi Lambda dapat berjalan di `Private subnet 1` atau. `Private subnet 2`
4. `Private subnet 1`dan `Private subnet 2` rute lalu lintas ke gateway NAT di subnet publik.
5. Gateway NAT mengirim lalu lintas keluar ke gateway internet dari subnet publik.
6. Data keluar ditransfer dari gateway internet ke server eksternal.

Tumpukan teknologi

- Lambda
- Amazon Virtual Private Cloud (Amazon VPC)

Otomatisasi dan skala

Anda dapat memastikan ketersediaan tinggi (HA) dengan menggunakan dua subnet publik dan dua subnet pribadi di Availability Zone yang berbeda. Bahkan jika satu Availability Zone menjadi tidak tersedia, solusi pola terus bekerja.

Alat

- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) menyediakan bagian AWS Cloud yang terisolasi secara logis tempat Anda dapat meluncurkan sumber daya AWS di jaringan virtual yang telah Anda tentukan. Jaringan virtual ini sangat mirip dengan jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaatnya yaitu menggunakan infrastruktur AWS yang dapat diskalakan.

Epik

Buat VPC baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC baru.	Masuk ke AWS Management Console, buka konsol Amazon VPC, lalu buat VPC bernama `Lambda VPC` yang memiliki `10.0.0.0/25` rentang CIDR. IPv4 Untuk informasi selengkap nya tentang membuat VPC, lihat Memulai Amazon VPC di dokumentasi Amazon VPC.	Administrator AWS

Buat dua subnet publik

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat subnet publik pertama.	1. Pada konsol Amazon VPC, pilih Subnet dan kemudian pilih Create Subnet. 2. Untuk tag Nama, masukkan`public-one` . 3. Untuk VPC, pilih. `Lambda VPC` 4. Pilih Availability Zone dan rekam. 5. Untuk blok IPv4 CIDR, masukkan `10.0.0.0/28` dan kemudian pilih Buat subnet.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat subnet publik kedua.	1. Pada konsol Amazon VPC, pilih Subnet dan kemudian pilih Create Subnet. 2. Untuk tag Nama, masukkan`public-two` . 3. Untuk VPC, pilih. `Lambda VPC` 4. **Important** Pilih Availabil ity Zone dan rekam. : Anda tidak dapat menggunak an Availability Zone yang berisi `public-one` subnet. 5. Untuk blok IPv4 CIDR, masukkan `10.0.0.16 /28` dan kemudian pilih Buat subnet.	Administrator AWS

Buat dua subnet pribadi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat subnet pribadi pertama.	1. Pada konsol Amazon VPC, pilih Subnet dan kemudian pilih Create Subnet. 2. Untuk tag Nama, masukkan`private-one` .	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Untuk VPC, pilih. `Lambda VPC` 4. Pilih Availability Zone yang berisi `public-one` subnet yang Anda buat sebelumnya. 5. Untuk blok IPv4 CIDR, masukkan `10.0.0.32/28` dan kemudian pilih Buat subnet.	
Buat subnet pribadi kedua.	1. Pada konsol Amazon VPC, pilih Subnet dan kemudian pilih Create Subnet. 2. Untuk tag Nama, masukkan`private-two` . 3. Untuk VPC, pilih. `Lambda VPC` 4. Pilih Availability Zone yang sama yang berisi `public-two` subnet yang Anda buat sebelumnya. 5. Untuk blok IPv4 CIDR, masukkan `10.0.0.64/28` dan kemudian pilih Buat subnet.	Administrator AWS

Buat dua alamat IP Elastis untuk gateway NAT Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat alamat IP elastis pertama.	1. Di konsol VPC Amazon, pilih Elastic IPs dan	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kemudian pilih Alokasikan alamat baru. 2. Pilih Alokasikan dan catat ID Alokasi untuk alamat IP Elastis yang baru dibuat. **Note** Alamat IP Elastis ini digunakan untuk gateway NAT pertama Anda.	
Buat alamat IP elastis kedua.	1. Di konsol VPC Amazon, pilih Elastic IPs dan kemudian pilih Alokasikan alamat baru. 2. Pilih Alokasikan dan catat ID Alokasi untuk alamat IP Elastis kedua ini. **Note** Alamat IP Elastis ini digunakan untuk gateway NAT kedua Anda.	Administrator AWS

Membuat gateway internet baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah gateway internet.	1. Di konsol VPC Amazon, pilih Internet Gateways dan kemudian pilih Buat gateway internet. 2. Masukkan `Lambda internet gateway` sebagai nama dan kemudian pilih Buat gateway internet. Pastikan Anda merekam ID gateway internet.	Administrator AWS
Lampirkan gateway internet ke VPC.	Pilih gateway internet yang baru Anda buat, dan lalu pilih Tindakan, Lampirkan ke VPC.	Administrator AWS

Buat dua gateway NAT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gateway NAT pertama.	1. Di konsol VPC Amazon, pilih NAT Gateways lalu pilih Create NAT Gateway. 2. Masukkan `nat-one` sebagai nama gateway NAT. 3. Pilih `public-one` sebagai subnet untuk membuat gateway NAT di. 4. Untuk jenis Konektivitas, pilih Publik.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Untuk ID alokasi IP Elastis, pilih alamat IP Elastis pertama yang Anda buat sebelumnya dan kaitkan dengan gateway NAT. 6. Pilih Buat gateway NAT.	
Buat gateway NAT kedua.	1. Di konsol VPC Amazon, pilih NAT Gateways lalu pilih Create NAT Gateway. 2. Masukkan `nat-two` sebagai nama gateway NAT. 3. Pilih `public-two` sebagai subnet untuk membuat gateway NAT di. 4. Untuk jenis Konektivitas, pilih Publik. 5. Untuk ID alokasi IP Elastis, pilih alamat IP Elastis kedua yang Anda buat sebelumny a dan kaitkan dengan gateway NAT. 6. Pilih Buat gateway NAT.	Administrator AWS

Buat tabel rute untuk subnet publik dan pribadi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel rute untuk subnet publik-satu.	1. Di konsol Amazon VPC, pilih Tabel Rute dan kemudian pilih Buat tabel rute.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Masukkan `public-one-subnet` sebagai nama tabel rute dan kemudian pilih Buat tabel rute. 3. Pilih tabel `public-one-subnet` rute, pilih Edit rute, lalu pilih Tambahkan rute. 4. Tentukan `0.0.0.0` di kotak Tujuan dan kemudian pilih ID gateway internet di Target daftar. 5. Pada tab Asosiasi subnet, pilih Edit asosiasi subnet, pilih **`public-one`** subnet dengan rentang **`10.0.0.0/28`** CIDR, lalu pilih Simpan asosiasi. 6. Pilih Simpan Perubahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel rute untuk subnet public-two.	1. Di konsol Amazon VPC, pilih Tabel Rute dan kemudian pilih Buat tabel rute. 2. Masukkan `public-two-subnet` sebagai nama tabel rute dan kemudian pilih Buat tabel rute. 3. Pilih tabel `public-two-subnet` rute, pilih Edit rute, lalu pilih Tambahkan rute. 4. Tentukan `0.0.0.0` di kotak Tujuan dan kemudian pilih ID gateway internet di Target daftar. 5. Pada tab Asosiasi subnet, pilih Edit asosiasi subnet, pilih **`public-two`** subnet dengan rentang **`10.0.0.16/28`** CIDR, lalu pilih Simpan asosiasi. 6. Pilih Simpan Perubahan.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel rute untuk subnet private-one.	1. Di konsol Amazon VPC, pilih Tabel Rute dan kemudian pilih Buat tabel rute. 2. Masukkan `private-one-subnet` sebagai nama tabel rute dan kemudian pilih Buat tabel rute. 3. Pilih tabel `private-one-subnet` rute, pilih Edit rute, lalu pilih Tambahkan rute. 4. Tentukan `0.0.0.0` di kotak Tujuan dan kemudian pilih gateway NAT di `public-one` subnet di daftar Target. 5. Pada tab Asosiasi subnet, pilih Edit asosiasi subnet, pilih **`private-one`** subnet dengan rentang **`10.0.0.32/28`** CIDR, lalu pilih Simpan asosiasi. 6. Pilih Simpan Perubahan.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel rute untuk subnet private-two.	1. Di konsol Amazon VPC, pilih Tabel Rute dan kemudian pilih Buat tabel rute. 2. Masukkan `private-two-subnet` sebagai nama tabel rute dan kemudian pilih Buat tabel rute. 3. Pilih tabel `private-two-subnet` rute, pilih Edit rute, lalu pilih Tambahkan rute. 4. Tentukan `0.0.0.0` di kotak Tujuan dan kemudian pilih gateway NAT di `public-two` subnet di daftar Target. 5. Pada tab Asosiasi subnet, pilih Edit asosiasi subnet, pilih **`private-two`** subnet dengan rentang **`10.0.0.64/28`** CIDR, lalu pilih Simpan asosiasi. 6. Pilih Simpan Perubahan.	Administrator AWS

Buat fungsi Lambda, tambahkan ke VPC, dan uji solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda baru.	1. Buka konsol AWS Lambda dan pilih Create function.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di bawah Informasi dasar, masukkan `Lambda test` di bawah Nama fungsi dan kemudian pilih bahasa pilihan Anda di bawah Runtime. 3. Pilih Buat fungsi.	
Tambahkan fungsi Lambda ke VPC Anda.	1. Di konsol AWS Lambda, pilih Fungsi lalu pilih fungsi yang Anda buat sebelumnya. 2. Pilih Konfigurasi, lalu pilih VPC. 3. Pilih Edit dan kemudian pilih `Lambda VPC` dan kedua subnet pribadi. 4. Pilih grup keamanan default untuk tujuan pengujian, lalu pilih Simpan.	Administrator AWS
Tulis kode untuk memanggil layanan eksternal.	1. Dalam bahasa pemrograman pilihan Anda, tulis kode untuk memanggil layanan eksternal yang mengembalikan alamat IP Anda. 2. Verifikasi bahwa alamat IP yang dikembalikan cocok dengan salah satu alamat IP Elastis Anda.	Administrator AWS

Sumber daya terkait

- Mengkonfigurasi fungsi Lambda untuk mengakses sumber daya di VPC

Identifikasi gambar kontainer duplikat secara otomatis saat bermigrasi ke repositori Amazon ECR

Rishabh Yadav dan Rishi Singla, Amazon Web Services

Ringkasan

Pola ini memberikan solusi otomatis untuk mengidentifikasi apakah gambar yang disimpan di repositori kontainer yang berbeda adalah duplikat. Pemeriksaan ini berguna saat Anda berencana untuk memigrasikan gambar dari repositori kontainer lain ke Amazon Elastic Container Registry (Amazon ECR).

Untuk informasi dasar, pola juga menjelaskan komponen gambar kontainer, seperti image digest, manifes, dan tag. Saat merencanakan migrasi ke Amazon ECR, Anda dapat memutuskan untuk menyinkronkan gambar kontainer di seluruh pendaftar kontainer dengan membandingkan intisari gambar. Sebelum memigrasikan gambar kontainer, Anda perlu memeriksa apakah gambar-gambar ini sudah ada di repositori Amazon ECR untuk mencegah duplikasi. Namun, sulit untuk mendeteksi duplikasi dengan membandingkan intisari gambar, dan ini dapat menyebabkan masalah pada fase migrasi awal. Pola ini membandingkan intisari dari dua gambar serupa yang disimpan dalam pendaftar kontainer yang berbeda dan menjelaskan mengapa intisari bervariasi, untuk membantu Anda membandingkan gambar secara akurat.

Prasyarat dan batasan

- Aktif Akun AWS
- Akses ke registri publik Amazon ECR
- Keakraban dengan yang berikut: Layanan AWS
  - AWS CodeCommit
  - AWS CodePipeline
  - AWS CodeBuild
  - AWS Identity and Access Management (IAM)
  - Amazon Simple Storage Service (Amazon S3)
- CodeCommit Kredensi yang dikonfigurasi (lihat instruksi)

Arsitektur

Komponen gambar kontainer

Diagram berikut menggambarkan beberapa komponen dari gambar kontainer. Komponen-komponen ini dijelaskan setelah diagram.

Istilah dan definisi

Istilah-istilah berikut didefinisikan dalam Spesifikasi Gambar Open Container Initiative (OCI).

- Registry: Layanan untuk penyimpanan dan manajemen gambar.
- Klien: Alat yang berkomunikasi dengan pendaftar dan bekerja dengan gambar lokal.
- Push: Proses untuk mengunggah gambar ke registri.
- Pull: Proses untuk mengunduh gambar dari registri.
- Blob: Bentuk biner konten yang disimpan oleh registri dan dapat diatasi dengan intisari.
- Indeks: Sebuah konstruksi yang mengidentifikasi beberapa manifes gambar untuk platform komputer yang berbeda (seperti x86-64 atau ARM 64-bit) atau jenis media. Untuk informasi selengkapnya, lihat Spesifikasi Indeks Gambar OCI.
- Manifest: Dokumen JSON yang mendefinisikan gambar atau artefak yang diunggah melalui titik akhir manifes. Manifes dapat mereferensikan gumpalan lain dalam repositori dengan menggunakan deskriptor. Untuk informasi selengkapnya, lihat Spesifikasi Manifes Gambar OCI.
- Filesystem layer: Pustaka sistem dan dependensi lain untuk sebuah gambar.
- Konfigurasi: Gumpalan yang berisi metadata artefak dan direferensikan dalam manifes. Untuk informasi selengkapnya, lihat Spesifikasi Konfigurasi Gambar OCI.
- Objek atau artefak: Item konten konseptual yang disimpan sebagai gumpalan dan dikaitkan dengan manifes yang menyertainya dengan konfigurasi.
- Digest: Pengidentifikasi unik yang dibuat dari hash kriptografi dari isi manifes. Image digest membantu mengidentifikasi gambar kontainer yang tidak dapat diubah secara unik. Saat Anda menarik gambar dengan menggunakan intisari, Anda akan mengunduh gambar yang sama setiap kali pada sistem operasi atau arsitektur apa pun. Untuk informasi lebih lanjut, lihat Spesifikasi Gambar OCI.
- Tag: Pengidentifikasi manifes yang dapat dibaca manusia. Dibandingkan dengan intisari gambar, yang tidak dapat diubah, tag bersifat dinamis. Tag yang menunjuk ke gambar dapat berubah dan berpindah dari satu gambar ke gambar lainnya, meskipun intisari gambar yang mendasarinya tetap sama.

Arsitektur target

Diagram berikut menampilkan arsitektur tingkat tinggi dari solusi yang disediakan oleh pola ini untuk mengidentifikasi gambar kontainer duplikat dengan membandingkan gambar yang disimpan di Amazon ECR dan repositori pribadi.

Alat

Layanan AWS

- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS CodeBuildadalah layanan build yang dikelola sepenuhnya yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodePipelinemembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.

Kode

Kode untuk pola ini tersedia di GitHub repositori Solusi otomatis untuk mengidentifikasi gambar kontainer duplikat antar repositori.

Praktik terbaik

- CloudFormation praktik terbaik
- AWS CodePipeline praktik terbaik

Epik

Tarik gambar kontainer dari repositori publik dan pribadi Amazon ECR

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tarik gambar dari repositori publik Amazon ECR.	Dari terminal, jalankan perintah berikut untuk menarik gambar `amazonlinux` dari repositori publik Amazon ECR.
	`$~ % docker pull public.ecr.aws/amazonlinux/amazonlinux:2018.03`
	Ketika gambar telah ditarik ke mesin lokal Anda, Anda akan melihat pull digest berikut, yang mewakili indeks gambar.
	`2018.03: Pulling from amazonlinux/amazonlinux`
	`4ddc0f8d367f: Pull complete`
	`Digest: sha256:f972d24199508c52de7ad37a298bda35d8a1bd7df158149b381c03f6c6e363b5`
	`Status: Downloaded newer image for public.ecr.aws/amazonlinux/amazonlinux:2018.03`
		Pengembang aplikasi, AWS DevOps, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`public.ecr.aws/amazonlinux/amazonlinux:2018.03`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Dorong gambar ke repositori pribadi Amazon ECR.	1. Buat repositori Amazon ECR pribadi yang dinamai `test_ecr_repository` di Wilayah AS Timur (Virginia N.) (). `us-east-1`
	
	`$~ % aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin <account-id>.dkr.ecr.us-east-1.amazonaws.com`
	`Login Succeeded`
	
	di mana `<account-id>` mengacu pada Anda Akun AWS.
	2. Tandai gambar lokal yang Anda tarik sebelumnya. Gunakan nilainya `public.ecr.aws/amazonlinux/amazonlinux:2018.03` dan dorong ke repositori pribadi Amazon ECR.
	
	`$~ % docker tag public.ecr.aws/amazonlinux/amazonlinux:2018.03 <account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest`
		Administrator AWS, AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`$~ % docker push <account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest`
	
	Saat Anda mendorong gambar ke repositori Amazon ECR, Docker akan mendorong gambar yang mendasarinya dan bukan indeks gambar.
	
	`The push refers to repository [<account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository] d5655967c2c4: Pushed latest: digest: sha256:52db9000073d93b9bdee6a7246a68c35a741aaade05a8f4febba0bf795cdac02 size: 529`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tarik gambar yang sama dari repositori pribadi Amazon ECR.	1. Dari terminal, jalankan perintah berikut untuk menarik gambar yang sebelumnya Anda dorong ke repositori pribadi Amazon ECR.
	
	`$~ % docker pull <account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest`
	`latest: Pulling from test_ecr_repository`
	`Digest: sha256:52db9000073d93b9bdee6a7246a68c35a741aaade05a8f4febba0bf795cdac02`
	`Status: Image is up to date for <account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest`
	`<account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest`
	
	Intisari untuk gambar ini cocok dengan intisari gambar yang Anda dorong ke repositori pribadi Amazon ECR, dan mewakili gambar yang mendasari nya. Nilai ini tidak cocok dengan indeks gambar
		Pengembang aplikasi, AWS DevOps, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	yang Anda tarik dari repositori publik.
	2. Untuk memverifikasi, ambil indeks gambar dengan digest.
	
	`curl -k -H "Authorization: Bearer $TOKEN" https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/sha256:f972d24199508c52de7ad37a298bda35d8a1bd7df158149b381c03f6c6e363b55`
	`{`
	`"schemaVersion": 2,`
	`"mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",`
	`"manifests": [`
	`{`
	`"mediaType": "application/vnd.docker.distribution.manifest.v2+json",`
	`"size": 529,`
	`"digest": "sha256:52db9000073d93b9bdee6a7246a68c35a741aaade05a8f4febba0bf795cdac02",`
	`"platform": {`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"architec
	ture": "amd64",
	"os":
	"linux"
	}
	}
	]
	}</pre>	

Bandingkan manifes gambar

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Temukan manifes gambar yang disimpan di repositori publik Amazon ECR.	Dari terminal, jalankan perintah berikut untuk menarik manifes gambar `public.ecr.aws/amazonlinux/amazonlinux:2018.03` dari repositori publik Amazon ECR.
	<pre>$~ % docker manifest
	inspect public.ec
	r.aws/amazonlinux/
	amazonlinux:2018.03
	{
	"schemaVersion": 2,
	"mediaType":
	"application/vnd.d
	ocker.distribution
	.manifest.list.v2+
	json",
	"manifests": [
	{
	"mediaType":
	"application/vnd.d
	ocker.distribution
	.manifest.v2+json",</pre>
		Administrator AWS, AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	"size": 529,
	"digest":
	"sha256:52db900007
	3d93b9bdee6a7246a6
	8c35a741aaade05a8f
	4febba0bf795cdac02",
	"platform": {
	"architec
	ture": "amd64",
	"os":
	"linux"
	}
	}
	]
	}	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Temukan manifes gambar yang disimpan di repositori pribadi Amazon ECR.	Dari terminal, jalankan perintah berikut untuk menarik manifes gambar `<account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest` dari repositori pribadi Amazon ECR.
	
	`$~ % docker manifest inspect <account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest`
	
	`{ "schemaVersion": 2, "mediaType": "application/vnd.docker.distribution.manifest.v2+json", "config": { "mediaType": "application/vnd.docker.container.image.v1+json", "size": 1477, "digest": "sha256:f7cee5e1af28ad4e147589c474d399b12d9b551ef4c3e11e02d982fce5eebc68" }, "layers": [{ "mediaType": "application/vnd.d`
		AWS DevOps, administrator sistem AWS, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ocker.image.rootfs .diff.tar.gzip", "size": 62267075, "digest": "sha256:4 ddc0f8d367f424871a 060e2067749f32bd36 a91085e714dcb15995 2f2d71453" }] }	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bandingkan intisari yang ditarik oleh Docker dengan intisari manifes untuk gambar di repositori pribadi Amazon ECR.	Pertanyaan lain adalah mengapa intisari yang disediakan oleh perintah docker pull berbeda dari intisari manifes untuk gambar. `<account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest` Intisari yang digunakan untuk docker pull mewakili intisari manifes gambar, yang disimpan dalam registri. Intisari ini dianggap sebagai root dari rantai hash, karena manifes berisi hash konten yang akan diunduh dan diimpor ke Docker. ID gambar yang digunakan dalam Docker dapat ditemukan dalam manifes ini sebagai`config.digest` . Ini mewakili konfigurasi gambar yang digunakan Docker. Jadi Anda bisa mengatakan bahwa manifes adalah amplop, dan gambar adalah isi amplop. Intisari manifes selalu berbeda dari ID gambar. Namun, manifes tertentu harus selalu menghasilkan ID gambar yang sama. Karena manifes digest adalah rantai hash, kami	AWS DevOps, administrator sistem AWS, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tidak dapat menjamin bahwa itu akan selalu sama untuk ID gambar yang diberikan. Dalam kebanyakan kasus, ini menghasilkan intisari yang sama, meskipun Docker tidak dapat menjaminnya. Perbedaan yang mungkin terjadi dalam manifes digest berasal dari Docker yang tidak menyimpan gumpalan yang dikompresi dengan gzip secara lokal. Oleh karena itu, mengekspor lapisan mungkin menghasilkan intisari yang berbeda, meskipun konten yang tidak dikompresi tetap sama. ID gambar memverifi kasi bahwa konten yang tidak dikompresi adalah sama; artinya, ID gambar sekarang menjadi pengenal yang dapat dialamatkan konten (). `chainID` Untuk mengonfirmasi informasi ini, Anda dapat membandingkan output dari perintah docker inspect di repositori publik dan pribadi Amazon ECR: 1. Jalankan perintah berikut dari terminal Anda untuk gambar yang disimpan di	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	repositori publik Amazon ECR.
	
	`$~ % docker inspect public.ecr.aws/amazonlinux/amazonlinux:2018.03`
	
	Untuk output dari perintah, lihat bagian Informasi tambahan.
	2. Jalankan perintah berikut dari terminal Anda untuk gambar yang disimpan di repositori pribadi Amazon ECR.
	
	`$~ % docker inspect <account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest`
	
	Untuk output dari perintah, lihat bagian Informasi tambahan.
	
	Hasilnya memverifikasi bahwa kedua gambar memiliki intisari ID gambar dan intisari lapisan yang sama.
	
	ID: `f7cee5e1af28ad4e147589c474d399b12d9b551ef4c3e11e02d982fce5eebc68`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Lapisan: `d5655967c 2c4e8d68f8ec7cf753 218938669e6c16ac13 24303c073c736a2e2a2` Selain itu, intisari didasarkan pada byte objek yang dikelola secara lokal (file lokal adalah tar dari lapisan gambar kontainer) atau gumpalan yang didorong ke server registri. Namun, ketika Anda mendorong gumpalan ke registri, tar dikompresi dan intisari dihitung dalam file tar terkompresi. Oleh karena itu, perbedaan nilai docker pull digest muncul dari kompresi yang diterapkan di tingkat registri (Amazon ECR private atau public). **Note** Penjelasan ini khusus untuk menggunakan klien Docker. Anda tidak akan melihat perilaku ini dengan klien lain seperti nerdctl atau Finch, karena mereka tidak secara otomatis mengompres gambar	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	selama operasi push dan pull.	

Secara otomatis mengidentifikasi gambar duplikat antara repositori publik dan pribadi Amazon ECR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Kloning repositori Github untuk pola ini ke dalam folder lokal: `$git clone https://github.com/aws-samples/automated-solution-to-identify-duplicate-container-images-between-repositories`	Administrator AWS, AWS DevOps
Siapkan CI/CD pipa.	GitHub Repositori menyertakan `.yaml` file yang membuat CloudFormation tumpukan untuk menyiapkan pipeline. AWS CodePipeline 1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol. 2. Buat tumpukan dengan menggunakan `pipeline.yaml` file template, yang ada di `code` folder di repositori kloning. 3. Menerima atau mengubah nilai default parameter.	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tentukan nilai untuk yang berikut: • Nama tumpukan • `ArtifactStoreBucketName` - Bucket S3 yang ada yang akan digunakan untuk menyimpan AWS CodePipeline artefak • `OutputBucket` - Bucket S3 yang ada yang akan digunakan untuk menyimpan gambar URIs duplikat • `SourceImageFile` - File teks yang ada bernama `input.txt` yang berisi gambar URIs dari repositori publik yang akan diperiksa terhadap repositori pribadi Amazon ECR untuk mendeteksi duplikasi 4. Tinjau dan sesuaikan opsi tumpukan, lalu pilih Kirim untuk menjalankan template. Pipeline akan diatur dengan dua tahap (CodeCommit dan CodeBuild, seperti yang ditunjukkan pada diagram arsitektur) untuk mengident ifikasi gambar di repositor	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	i pribadi yang juga ada di repositori publik. Pipeline dikonfigurasi dengan sumber daya berikut: • CodePipeline untuk orkestrasi pipa penyebaran. • Sebuah CodeCommit repositori untuk menyimpan skrip bash dan file input. Skrip bash digunakan untuk membandingkan gambar kontainer IDs di repositor i publik dan pribadi untuk menemukan duplikasi. Pemeriksaan ini dilakukan di semua repositori dalam yang ditentukan Akun AWS dalam satu. Wilayah AWS • Sebuah CodeBuild proyek untuk memanggil skrip bash untuk mengidentifikasi gambar yang sudah ada di repositori Amazon ECR. • Peran IAM yang diperlukan untuk memungkinkan akses. • Bucket S3 untuk menyimpan file output yang berisi gambar URIs. • Ember S3 lain untuk menyimpan CodePipeline artefak.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengisi CodeCommit repositori.	Untuk mengisi CodeCommit repositori, lakukan langkah-langkah ini: 1. Buka CodeCommit konsol dan arahkan ke Wilayah AWS tempat Anda membuat CloudFormation tumpukan. 2. Temukan repositori yang Anda sediakan dengan menggunakan CloudFormation skrip dari daftar, pilih URL Kloning, lalu salin protokol URL HTTPS untuk terhubung ke repositori. 3. Buka prompt perintah dan jalankan perintah git clone dengan URL HTTPS yang Anda salin pada langkah sebelumnya. 4. Arahkan ke direktori root. Buat file bernama `input.txt` dan isi file ini dengan registri gambar publik Amazon ECR URIs yang ingin Anda cari di repositori Amazon ECR pribadi. 5. Salin file`script.sh`,`buildspec.yml`, dan `input.txt` dari salinan lokal GitHub repositori Solusi otomatis untuk	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	mengidentifikasi gambar kontainer duplikat antara repositori ke repositori kloning. CodeCommit
	6. Unggah file ke CodeCommit dengan menggunakan perintah ini:
	`git add .`
	`git commit -m "added input files"`
	`git push`	

Bersihkan.	Untuk menghindari timbulnya tagihan future, hapus sumber daya dengan mengikuti langkah-langkah berikut: 1. Arahkan ke ember S3 yang menyimpan CodePipeline artefak, dan kosongkan ember. 2. Arahkan ke bucket S3 yang menyimpan gambar duplikat URIs, dan kosongkan bucket. 3. Arahkan ke CloudFormation konsol dan hapus tumpukan yang Anda buat untuk mengatur pipeline.	Administrator AWS

Pemecahan masalah

Isu	Solusi
Ketika Anda mencoba untuk mendorong, menarik, atau berinteraksi dengan CodeCommit repositori dari terminal atau baris perintah, Anda diminta untuk memberikan nama pengguna dan kata sandi, dan Anda harus menyediakan kredensi Git untuk pengguna IAM Anda.	Penyebab paling umum untuk kesalahan ini adalah sebagai berikut: • Komputer lokal Anda menjalankan sistem operasi yang tidak mendukung manajemen kredensi, atau tidak memiliki utilitas manajemen kredenal yang diinstal. • Kredensi Git untuk pengguna IAM Anda belum disimpan ke salah satu sistem manajemen kredensi ini. Tergantung pada sistem operasi dan lingkungan lokal Anda, Anda mungkin perlu menginstal pengelola kredensial, mengkonfigurasi pengelola kredensial yang disertakan dalam sistem operasi Anda, atau menyesuaikan lingkungan lokal Anda untuk menggunakan penyimpanan kredensial. Misalnya, jika komputer menjalankan macOS, maka Anda dapat menggunakan utilitas Keychain Access untuk menyimpan kredensial Anda. Jika komputer Anda menjalankan Windows, Anda dapat menggunakan Pengelola Kredensial Git yang diinstal dengan Git untuk Windows. Untuk informasi selengkapnya, lihat Penyetelan untuk pengguna HTTPS yang menggunakan kredensial Git dalam CodeCommit dokumentasi dan Penyimpanan Kredenal dalam dokumentasi Git.
Anda mengalami kesalahan HTTP 403 atau “tidak ada kredensial autentikasi dasar” saat	Anda mungkin menemukan pesan kesalahan ini dari perintah docker push atau docker pull, bahkan jika Anda telah berhasil mengauten

Isu	Solusi
Anda mendorong gambar ke repositori Amazon ECR.	tikasi ke Docker dengan menggunakan perintah aws ecr. get-login-password Penyebab yang diketahui adalah: • Anda telah mengautentikasi ke Wilayah yang berbeda. Untuk informasi selengkapnya, lihat Autentikasi registri pribadi di dokumentasi Amazon ECR. • Anda telah mengautentikasi untuk mendorong ke repositori yang tidak memiliki izin untuk Anda. Untuk informasi selengkap nya, lihat Kebijakan repositori pribadi di dokumentasi Amazon ECR. • Token Anda telah kedaluwarsa. Periode kedaluwarsa default untuk token yang diperoleh dengan menggunakan `GetAuthor izationToken` operasi adalah 12 jam.

Sumber daya terkait

- Solusi otomatis untuk mengidentifikasi gambar kontainer duplikat antara repositori (repositori) GitHub
- Galeri publik Amazon ECR
- Gambar pribadi di Amazon ECR (dokumentasi Amazon ECR)
- AWS::CodePipeline::Pipeline sumber daya (CloudFormation dokumentasi)
- Spesifikasi Format Gambar OCI

Informasi tambahan

Output inspeksi Docker untuk gambar di repositori publik Amazon ECR

```
[
    {
```

```
        "Id":
 "sha256:f7cee5e1af28ad4e147589c474d399b12d9b551ef4c3e11e02d982fce5eebc68",
        "RepoTags": [
            "<account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest",
            "public.ecr.aws/amazonlinux/amazonlinux:2018.03"
        ],
        "RepoDigests": [
            "<account-id>.dkr.ecr.us-east-1.amazonaws.com/
test_ecr_repository@sha256:52db9000073d93b9bdee6a7246a68c35a741aaade05a8f4febba0bf795cdac02",
            "public.ecr.aws/amazonlinux/
amazonlinux@sha256:f972d24199508c52de7ad37a298bda35d8a1bd7df158149b381c03f6c6e363b5"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2023-02-23T06:20:11.575053226Z",
        "Container":
 "ec7f2fc7d2b6a382384061247ef603e7d647d65f5cd4fa397a3ccbba9278367c",
        "ContainerConfig": {
            "Hostname": "ec7f2fc7d2b6",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/sh",
                "-c",
                "#(nop) ",
                "CMD [\"/bin/bash\"]"
            ],
            "Image":
 "sha256:c1bced1b5a65681e1e0e52d0a6ad17aaf76606149492ca0bf519a466ecb21e51",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": {}
        },
```

```
        "DockerVersion": "20.10.17",
        "Author": "",
        "Config": {
            "Hostname": "",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/bash"
            ],
            "Image":
 "sha256:c1bced1b5a65681e1e0e52d0a6ad17aaf76606149492ca0bf519a466ecb21e51",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": null
        },
        "Architecture": "amd64",
        "Os": "linux",
        "Size": 167436755,
        "VirtualSize": 167436755,
        "GraphDriver": {
            "Data": {
                "MergedDir": "/var/lib/docker/overlay2/
c2c2351a82b26cbdf7782507500e5adb5c2b3a2875bdbba79788a4b27cd6a913/merged",
                "UpperDir": "/var/lib/docker/overlay2/
c2c2351a82b26cbdf7782507500e5adb5c2b3a2875bdbba79788a4b27cd6a913/diff",
                "WorkDir": "/var/lib/docker/overlay2/
c2c2351a82b26cbdf7782507500e5adb5c2b3a2875bdbba79788a4b27cd6a913/work"
            },
            "Name": "overlay2"
        },
        "RootFS": {
            "Type": "layers",
            "Layers": [
```

```
 "sha256:d5655967c2c4e8d68f8ec7cf753218938669e6c16ac1324303c073c736a2e2a2"
            ]
        },
        "Metadata": {
            "LastTagTime": "2023-03-02T10:28:47.142155987Z"
        }
    }
]
```

Output inspeksi Docker untuk gambar di repositori pribadi Amazon ECR

```
[
    {
        "Id":
 "sha256:f7cee5e1af28ad4e147589c474d399b12d9b551ef4c3e11e02d982fce5eebc68",
        "RepoTags": [
            "<account-id>.dkr.ecr.us-east-1.amazonaws.com/test_ecr_repository:latest",
            "public.ecr.aws/amazonlinux/amazonlinux:2018.03"
        ],
        "RepoDigests": [
            "<account-id>.dkr.ecr.us-east-1.amazonaws.com/
test_ecr_repository@sha256:52db9000073d93b9bdee6a7246a68c35a741aaade05a8f4febba0bf795cdac02",
            "public.ecr.aws/amazonlinux/
amazonlinux@sha256:f972d24199508c52de7ad37a298bda35d8a1bd7df158149b381c03f6c6e363b5"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2023-02-23T06:20:11.575053226Z",
        "Container":
 "ec7f2fc7d2b6a382384061247ef603e7d647d65f5cd4fa397a3ccbba9278367c",
        "ContainerConfig": {
            "Hostname": "ec7f2fc7d2b6",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
```

```
            ],
            "Cmd": [
                "/bin/sh",
                "-c",
                "#(nop) ",
                "CMD [\"/bin/bash\"]"
            ],
            "Image":
"sha256:c1bced1b5a65681e1e0e52d0a6ad17aaf76606149492ca0bf519a466ecb21e51",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": {}
        },
        "DockerVersion": "20.10.17",
        "Author": "",
        "Config": {
            "Hostname": "",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/bash"
            ],
            "Image":
"sha256:c1bced1b5a65681e1e0e52d0a6ad17aaf76606149492ca0bf519a466ecb21e51",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": null
        },
        "Architecture": "amd64",
        "Os": "linux",
        "Size": 167436755,
```

```
        "VirtualSize": 167436755,
        "GraphDriver": {
            "Data": {
                "MergedDir": "/var/lib/docker/overlay2/
c2c2351a82b26cbdf7782507500e5adb5c2b3a2875bdbba79788a4b27cd6a913/merged",
                "UpperDir": "/var/lib/docker/overlay2/
c2c2351a82b26cbdf7782507500e5adb5c2b3a2875bdbba79788a4b27cd6a913/diff",
                "WorkDir": "/var/lib/docker/overlay2/
c2c2351a82b26cbdf7782507500e5adb5c2b3a2875bdbba79788a4b27cd6a913/work"
            },
            "Name": "overlay2"
        },
        "RootFS": {
            "Type": "layers",
            "Layers": [

 "sha256:d5655967c2c4e8d68f8ec7cf753218938669e6c16ac1324303c073c736a2e2a2"
            ]
        },
        "Metadata": {
            "LastTagTime": "2023-03-02T10:28:47.142155987Z"
        }
    }
]
```

Instal Agen SSM di node pekerja Amazon EKS dengan menggunakan Kubernetes DaemonSet

Mahendra Revanasiddappa, Amazon Web Services

Ringkasan

Catatan, September 2021: Amazon EKS terbaru yang dioptimalkan AMIs menginstal Agen SSM secara otomatis. Untuk informasi lebih lanjut, lihat catatan rilis untuk Juni 2021 AMIs.

Di Amazon Elastic Kubernetes Service (Amazon EKS), karena pedoman keamanan, node pekerja tidak memiliki pasangan kunci Secure Shell (SSH) yang melekat padanya. Pola ini menunjukkan bagaimana Anda dapat menggunakan tipe DaemonSet sumber daya Kubernetes untuk menginstal AWS Systems Manager Agent (SSM Agent) pada semua node pekerja, alih-alih menginstalnya secara manual atau mengganti Amazon Machine Image (AMI) untuk node. DaemonSet menggunakan pekerjaan cron pada node pekerja untuk menjadwalkan instalasi Agen SSM. Anda juga dapat menggunakan pola ini untuk menginstal paket lain pada node pekerja.

Saat Anda memecahkan masalah di klaster, menginstal Agen SSM sesuai permintaan memungkinkan Anda membuat sesi SSH dengan node pekerja, mengumpulkan log, atau melihat konfigurasi instans, tanpa pasangan kunci SSH.

Prasyarat dan batasan

Prasyarat

- Cluster Amazon EKS yang ada dengan node pekerja Amazon Elastic Compute Cloud (Amazon EC2).
- Instance kontainer harus memiliki izin yang diperlukan untuk berkomunikasi dengan layanan SSM. Peran terkelola AWS Identity and Access Management (IAM) Amazon SSMManaged InstanceCore menyediakan izin yang diperlukan agar Agen SSM berjalan pada instans. EC2 Untuk informasi selengkapnya, lihat dokumentasi AWS Systems Manager.

Batasan

- Pola ini tidak berlaku untuk AWS Fargate, karena DaemonSets tidak didukung pada platform Fargate.
- Pola ini hanya berlaku untuk node pekerja berbasis Linux.

- DaemonSet Pod berjalan dalam mode istimewa. Jika klaster Amazon EKS memiliki webhook yang memblokir pod dalam mode istimewa, Agen SSM tidak akan diinstal.

Arsitektur

Diagram berikut menggambarkan arsitektur untuk pola ini.

Alat

Alat

- kubectl adalah utilitas baris perintah yang digunakan untuk berinteraksi dengan kluster Amazon EKS. Pola ini digunakan `kubectl` untuk menyebarkan a DaemonSet di kluster Amazon EKS, yang akan menginstal Agen SSM di semua node pekerja.
- Amazon EKS memudahkan Anda menjalankan Kubernetes di AWS tanpa harus menginstal, mengoperasikan, dan memelihara bidang kontrol atau node Kubernetes Anda sendiri. Kubernetes adalah sebuah sistem sumber terbuka untuk melakukan otomatisasi terhadap deployment, penskalaan, dan pengelolaan aplikasi terkontainer.
- AWS Systems Manager Session Manager memungkinkan Anda mengelola EC2 instans, instans lokal, dan mesin virtual (VMs) melalui shell interaktif berbasis browser satu klik, atau melalui AWS Command Line Interface (AWS CLI).

Kode

Gunakan kode berikut untuk membuat file DaemonSet konfigurasi yang akan menginstal Agen SSM di cluster Amazon EKS. Ikuti instruksi di bagian Epik.

```
cat << EOF > ssm_daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  labels:
    k8s-app: ssm-installer
  name: ssm-installer
  namespace: kube-system
spec:
  selector:
    matchLabels:
```

```
      k8s-app: ssm-installer
  template:
    metadata:
      labels:
        k8s-app: ssm-installer
    spec:
      containers:
      - name: sleeper
        image: busybox
        command: ['sh', '-c', 'echo I keep things running! && sleep 3600']
      initContainers:
      - image: amazonlinux
        imagePullPolicy: Always
        name: ssm
        command: ["/bin/bash"]
        args: ["-c","echo '* * * * * root yum install -y https://s3.amazonaws.com/
ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm & rm -rf /etc/
cron.d/ssmstart' > /etc/cron.d/ssmstart"]
        securityContext:
          allowPrivilegeEscalation: true
        volumeMounts:
        - mountPath: /etc/cron.d
          name: cronfile
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      volumes:
      - name: cronfile
        hostPath:
          path: /etc/cron.d
          type: Directory
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      terminationGracePeriodSeconds: 30
EOF
```

Epik

Mengatur kubectl

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan konfigurasikan kubectl untuk mengakses kluster EKS.	Jika `kubectl` belum diinstal dan dikonfigurasi untuk mengakses kluster Amazon EKS, lihat Menginstal kubectl di dokumentasi Amazon EKS.	DevOps

Menyebarkan DaemonSet

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file DaemonSet konfigurasi.	Gunakan kode di bagian Kode sebelumnya dalam pola ini untuk membuat file DaemonSet konfigurasi yang disebut`ssm_daemonset.yaml`, yang akan digunakan ke cluster Amazon EKS. Pod yang diluncurkan oleh DaemonSet memiliki wadah utama dan `init` wadah. Wadah utama memiliki `sleep` perintah. `init`Wadah menyertakan `command` bagian yang membuat file pekerjaan cron untuk menginstal Agen SSM di jalur. `/etc/cron.d/` Pekerjaan cron hanya berjalan sekali, dan file yang dibuatnya dihapus secara	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	otomatis setelah pekerjaan selesai. Ketika wadah init telah selesai, wadah utama menunggu selama 60 menit sebelum keluar. Setelah 60 menit, pod baru diluncurkan. Pod ini menginstal Agen SSM, jika tidak ada, atau memperbarui Agen SSM ke versi terbaru. Jika diperlukan, Anda dapat memodifikasi `sleep` perintah untuk me-restart pod sekali sehari atau untuk menjalankan lebih sering.	
Terapkan DaemonSet di cluster Amazon EKS.	Untuk menyebarkan file DaemonSet konfigurasi yang Anda buat pada langkah sebelumnya di klaster Amazon EKS, gunakan perintah berikut: `kubectl apply -f ssm_daemonset.yaml` Perintah ini membuat sebuah DaemonSet untuk menjalankan pod pada node pekerja untuk menginstal Agen SSM.	DevOps

Sumber daya terkait

- Menginstal kubectl (dokumentasi Amazon EKS)
- Menyiapkan Session Manager (dokumentasi AWS Systems Manager)

Instal Agen dan CloudWatch agen SSM di node pekerja Amazon EKS menggunakan preBootstrapCommands

Kamahadevi Hiremath, Amazon Web Services

Ringkasan

Pola ini menyediakan contoh kode dan langkah-langkah untuk menginstal AWS Systems Manager Agent (Agen SSM) dan CloudWatch agen Amazon di node pekerja Amazon Elastic Kubernetes Service (Amazon EKS) di Amazon Web Services (AWS) Cloud selama pembuatan klaster Amazon EKS. Anda dapat menginstal Agen dan CloudWatch agen SSM dengan menggunakan `preBootstrapCommands` properti dari skema file `eksctl` konfigurasi (dokumentasi Weaveworks). Kemudian, Anda dapat menggunakan Agen SSM untuk terhubung ke node worker Anda tanpa menggunakan key pair Amazon Elastic Compute Cloud EC2 (Amazon). Selain itu, Anda dapat menggunakan CloudWatch agen untuk memantau pemanfaatan memori dan disk pada node pekerja Amazon EKS Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Utilitas baris perintah eksctl, diinstal dan dikonfigurasi di macOS, Linux, atau Windows
- Utilitas baris perintah kubectl, diinstal dan dikonfigurasi di macOS, Linux, atau Windows

Batasan

- Sebaiknya hindari menambahkan skrip yang sudah berjalan lama ke `preBootstrapCommands` properti, karena ini menunda node bergabung dengan klaster Amazon EKS selama aktivitas penskalaan. Kami menyarankan Anda membuat Amazon Machine Image (AMI) kustom sebagai gantinya.
- Pola ini hanya berlaku untuk instans Amazon EC2 Linux.

Arsitektur

Tumpukan teknologi

- Amazon CloudWatch
- Amazon Elastic Kubernetes Service (Amazon EKS)
- AWS Systems Manager Parameter Store

Arsitektur target

Diagram berikut menunjukkan contoh pengguna yang menghubungkan ke node pekerja Amazon EKS menggunakan Agen SSM yang diinstal menggunakan. `preBootstrapCommands`

Diagram menunjukkan alur kerja berikut:

1. Pengguna membuat kluster Amazon EKS dengan menggunakan file `eksctl` konfigurasi dengan `preBootstrapCommands` properti, yang menginstal Agen dan CloudWatch agen SSM.
2. Setiap instance baru yang bergabung dengan cluster nanti karena aktivitas penskalaan dibuat dengan Agen dan agen SSM yang sudah diinstal sebelumnya. CloudWatch
3. Pengguna terhubung ke Amazon EC2 dengan menggunakan Agen SSM dan kemudian memonitor memori dan pemanfaatan disk dengan menggunakan agen. CloudWatch

Alat

- Amazon CloudWatch membantu Anda memantau metrik sumber daya AWS Anda dan aplikasi yang Anda jalankan di AWS secara real time.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Systems Manager Parameter Store menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.
- AWS Systems Manager Session Manager membantu Anda mengelola EC2 instans, instans lokal, dan mesin virtual melalui shell berbasis browser interaktif, sekali klik, atau melalui AWS Command Line Interface (AWS CLI).
- eksctl adalah utilitas baris perintah untuk membuat dan mengelola cluster Kubernetes di Amazon EKS.
- kubectl adalah utilitas baris perintah untuk berkomunikasi dengan server API cluster.

Epik

Buat kluster Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Simpan file konfigurasi CloudWatch agen.	Simpan file konfigurasi CloudWatch agen di AWS Systems Manager Parameter Store di Wilayah AWS tempat Anda ingin membuat klaster Amazon EKS. Untuk melakukan ini, buat parameter di AWS Systems Manager Parameter Store dan catat nama parameter (misalnya ,`AmazonCloudwatch-linux`). Untuk informasi selengkapnya, lihat kode file konfigurasi CloudWatch agen Contoh di bagian Informasi tambahan dari pola ini.	DevOps insinyur
Buat file konfigurasi eksctl dan cluster.	1. Buat file `eksctl` konfigurasi yang menyertakan langkah-langkah instalasi CloudWatch agen dan Agen SSM. Untuk informasi selengkapnya, lihat Contoh kode file konfigurasi eksctl di bagian Informasi tambahan dari pola ini. 2. Buat cluster dengan menjalankan `eksctl`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`create cluster -f cluster.yaml` perintah.	

Verifikasi bahwa Agen dan CloudWatch agen SSM bekerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji Agen SSM.	Gunakan SSH untuk menyambung ke node cluster Amazon EKS Anda dengan menggunakan salah satu metode yang tercakup dalam Memulai sesi dari dokumenta si AWS Systems Manager.	AWS DevOps
Uji CloudWatch agennya.	Gunakan CloudWatch konsol untuk memvalidasi CloudWatc h agen: 1. Masuk ke AWS Management Console dan buka CloudWatch konsol. 2. Pada panel navigasi, perluas Metrik, lalu pilih Semua metrik. 3. Di kotak pencarian di tab Browse, masukkan lalu pilih CWAgent metrik untuk melihat metrik memori dan disk.	AWS DevOps

Sumber daya terkait

- Menginstal dan menjalankan CloudWatch agen di server Anda (CloudWatch dokumentasi Amazon)
- Membuat parameter Systems Manager (konsol) (dokumentasi AWS Systems Manager)
- Buat file konfigurasi CloudWatch agen (CloudWatch dokumentasi Amazon)
- Memulai sesi (AWS CLI) (dokumentasi AWS Systems Manager)
- Memulai sesi (EC2 konsol Amazon) (dokumentasi AWS Systems Manager)

Informasi tambahan

Contoh file konfigurasi CloudWatch agen

Dalam contoh berikut, CloudWatch agen dikonfigurasi untuk memantau pemanfaatan disk dan memori pada instance Amazon Linux:

```
{
    "agent": {
        "metrics_collection_interval": 60,
        "run_as_user": "cwagent"
    },
    "metrics": {
        "append_dimensions": {
            "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
            "ImageId": "${aws:ImageId}",
            "InstanceId": "${aws:InstanceId}",
            "InstanceType": "${aws:InstanceType}"
        },
        "metrics_collected": {
            "disk": {
                "measurement": [
                    "used_percent"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
            "mem": {
                "measurement": [
```

```
                    "mem_used_percent"
                ],
                "metrics_collection_interval": 60
            }
        }
    }
}
```

Contoh file konfigurasi eksctl

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: test
  region: us-east-2
  version: "1.24"
managedNodeGroups:
  - name: test
    minSize: 2
    maxSize: 4
    desiredCapacity: 2
    volumeSize: 20
    instanceType: t3.medium
    preBootstrapCommands:
    - sudo yum install amazon-ssm-agent -y
    - sudo systemctl enable amazon-ssm-agent
    - sudo systemctl start amazon-ssm-agent
    - sudo yum install amazon-cloudwatch-agent -y
    - sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c ssm:AmazonCloudwatch-linux
    iam:
      attachPolicyARNs:
        - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
        - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
        - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
        - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Detail kode tambahan

- Di baris terakhir `preBootstrapCommands` properti, `AmazonCloudwatch-linux` adalah nama parameter yang dibuat di AWS System Manager Parameter Store. Anda harus menyertakan `AmazonCloudwatch-linux` di Parameter Store di Wilayah AWS yang sama tempat Anda

membuat klaster Amazon EKS. Anda juga dapat menentukan jalur file, tetapi sebaiknya gunakan Systems Manager untuk otomatisasi dan penggunaan kembali yang lebih mudah.

- Jika Anda menggunakan `preBootstrapCommands` dalam file `eksctl` konfigurasi, Anda akan melihat dua templat peluncuran di AWS Management Console. Template peluncuran pertama mencakup perintah yang ditentukan dalam`preBootstrapCommands`. Template kedua mencakup perintah yang ditentukan dalam `preBootstrapCommands` dan data pengguna Amazon EKS default. Data ini diperlukan untuk mendapatkan node untuk bergabung dengan cluster. Grup Auto Scaling grup node menggunakan data pengguna ini untuk memutar instance baru.
- Jika Anda menggunakan `iam` atribut dalam file `eksctl` konfigurasi, Anda harus mencantumkan kebijakan Amazon EKS default dengan kebijakan tambahan apa pun yang diperlukan dalam kebijakan AWS Identity and Access Management (IAM) terlampir. Dalam cuplikan kode dari Buat file konfigurasi eksctl dan langkah cluster, *`CloudWatchAgentServerPolicy`* dan kebijakan tambahan `AmazonSSMMangedInstanceCore` ditambahkan untuk memastikan bahwa CloudWatch agen dan Agen SSM berfungsi seperti yang diharapkan. Kebijakan`AmazonEKSWorkerNodePolicy`,`AmazonEKS_CNI_Policy`, adalah `AmazonEC2ContainerRegistryReadOnly` kebijakan wajib yang diperlukan agar klaster Amazon EKS berfungsi dengan benar.

Migrasikan Pengontrol Ingress NGINX saat mengaktifkan Mode Otomatis Amazon EKS

Olawale Olaleye dan Shamanth Devagari, Amazon Web Services

Ringkasan

Mode Otomatis EKS untuk Amazon Elastic Kubernetes Service (Amazon EKS) dapat mengurangi overhead operasional menjalankan beban kerja Anda di klaster Kubernetes. Mode ini memungkinkan AWS untuk juga mengatur dan mengelola infrastruktur atas nama Anda. Ketika Anda mengaktifkan Mode Otomatis EKS pada cluster yang ada, Anda harus hati-hati merencanakan migrasi konfigurasi NGINX Ingress Controller. Ini karena transfer langsung Network Load Balancers tidak dimungkinkan.

Anda dapat menggunakan strategi blue/green penerapan untuk memigrasikan instans NGINX Ingress Controller saat Anda mengaktifkan Mode Otomatis EKS EKS di klaster Amazon EKS yang ada.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Cluster Amazon EKS yang menjalankan Kubernetes versi 1.29 atau yang lebih baru
- Pengaya Amazon EKS yang menjalankan versi minimum
- Versi terbaru dari kubectl
- Instance NGINX Ingress Controller yang ada
- (Opsional) Zona yang dihosting di Amazon Route 53 untuk perpindahan lalu lintas berbasis DNS

Arsitektur

Penerapan biru/hijau adalah strategi penyebaran di mana Anda membuat dua lingkungan yang terpisah namun identik. Blue/green deployment menyediakan kemampuan rilis downtime dan rollback mendekati nol. Ide dasarnya adalah untuk menggeser lalu lintas antara dua lingkungan identik yang menjalankan versi berbeda dari aplikasi Anda.

Gambar berikut menunjukkan migrasi Network Load Balancers dari dua instance NGINX Ingress Controller yang berbeda saat mengaktifkan Mode Otomatis EKS. Anda menggunakan blue/green penyebaran untuk mengalihkan lalu lintas antara dua Network Load Balancers.

Namespace asli adalah namespace biru. Di sinilah layanan dan instance NGINX Ingress Controller asli berjalan, sebelum Anda mengaktifkan Mode Otomatis EKS. Layanan dan instance asli terhubung ke Network Load Balancer yang memiliki nama DNS yang dikonfigurasi di Route 53. AWS Load Balancer Controller menggunakan Network Load Balancer ini di target virtual private cloud (VPC).

Diagram menunjukkan alur kerja berikut untuk menyiapkan lingkungan untuk blue/green penerapan:

1. Instal dan konfigurasikan instance NGINX Ingress Controller lain di namespace yang berbeda, namespace hijau.
2. Di Route 53, konfigurasikan nama DNS untuk Network Load Balancer baru.

Alat

Layanan AWS

- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- Elastic Load Balancing mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat lainnya

- Helm adalah pengelola paket open source untuk Kubernetes yang membantu Anda menginstal dan mengelola aplikasi di klaster Kubernetes Anda.
- kubectl adalah antarmuka baris perintah yang membantu Anda menjalankan perintah terhadap klaster Kubernetes.
- NGINX Ingress Controller menghubungkan aplikasi dan layanan Kubernetes dengan penanganan permintaan, auth, sumber daya kustom swalayan, dan debugging.

Epik

Tinjau lingkungan yang ada

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan bahwa instans NGINX Ingress Controller asli beroperasi.	Masukkan perintah berikut untuk memverifikasi bahwa sumber daya di `ingress-nginx` namespace beroperasi. Jika Anda telah menerapkan NGINX Ingress Controller di namespace lain, perbarui nama namespace dalam perintah ini.
	`kubectl get all -n ingress-nginx`
	Pada output, konfirmasikan bahwa pod NGINX Ingress Controller dalam status berjalan. Berikut ini adalah contoh output:
	`NAME READY STATUS RESTARTS AGE pod/ingress-nginx-admission-create-xqn9d 0/1 Completed 0 88m pod/ingress-nginx-admission-patch-lhk4j 0/1 Completed 1 88m pod/ingress-nginx-controller-68f68f859-`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`xrz74 1/1 Running 2 (10m ago) 72m`
	
	`NAME`
	
	`TYPE CLUSTER-IP EXTERNAL-IP`
	
	`PORT(S) AGE`
	`service/ingress-nginx-controller LoadBalancer 10.100.67.255 k8s-ingressn-ingressn-abcdefg-12345.elb.eu-west-1.amazonaws.com 80:30330/TCP,443:31462/TCP 88m`
	`service/ingress-nginx-controller-admission ClusterIP 10.100.201.176 <none> 443/TCP 88m`
	
	`NAME`
	
	`READY UP-TO-DATE AVAILABLE AGE`
	`deployment.apps/ingress-nginx-controller 1/1 1 1 88m`
	
	`NAME`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`DESIRED CURRENT READY AGE replicaset.apps/ingress-nginx-controller-68f68f859 1 1 1 72m replicaset.apps/ingress-nginx-controller-d8c96cf68 0 0 0 88m NAME STATUS COMPLETIONS DURATION AGE job.batch/ingress-nginx-admission-create Complete 1/1 4s 88m job.batch/ingress-nginx-admission-patch Complete 1/1 5s 88m`	

Menerapkan HTTPd beban kerja sampel untuk menggunakan NGINX Ingress Controller

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sumber daya Kubernetes.	Masukkan perintah berikut untuk membuat contoh deployment, service, dan ingress Kubernetes: `kubectl create deployment demo --image=httpd --port=80`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`kubectl expose deployment demo`
	
	`kubectl create ingress demo --class=nginx \ --rule nginxautomode.local.dev/=demo:80`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tinjau sumber daya yang digunakan.	Masukkan perintah berikut untuk melihat daftar sumber daya yang digunakan:
	`kubectl get all,ingress`
	Pada output, konfirmasikan bahwa HTTPd pod sampel berada dalam status berjalan. Berikut ini adalah contoh output:
	`NAME READY STATUS RESTARTS AGE pod/demo-7d94f8cb4f-q68wc 1/1 Running 0 59m`
	`NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE service/demo ClusterIP 10.100.78.155 <none> 80/TCP 59m service/kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 117m`
	`NAME READY UP-TO-DATE AVAILABLE AGE deployment.apps/demo 1/1 1 1 59m`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`NAME DESIRED CURRENT READY AGE replicaset.apps/ demo-7d94f8cb4f 1 1 1 59m`
	
	`NAME CLASS HOSTS ADDRESS PORTS AGE ingress.networ king.k8s.io/demo nginx nginxauto mode.local.dev k8s-ingre ssn-ingressn-abcde fg-12345.elb.eu-we st-1.amazonaws.com 80 56m`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan bahwa layanan dapat dijangkau.	Masukkan perintah berikut untuk mengonfirmasi bahwa layanan dapat dijangkau melalui nama DNS Network Load Balancer:
	`curl -H "Host: nginxautomode.local.dev" http://k8s-ingressn-ingressn-abcdefg-12345.elb.eu-west-1.amazonaws.com`
	Berikut ini adalah output yang diharapkan:
	`<html><body><h1>It works!</h1></body></html>`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

(Opsional) Buat catatan DNS.	1. Ikuti petunjuk dalam Membuat catatan menggunakan konsol Amazon Route 53 (dokumentasi Route 53) untuk membuat catatan DNS untuk domain yang dikonfigurasi.
	2. Masukkan perintah berikut untuk mengonfirmasi bahwa layanan dapat dijangkau melalui nama domain yang dikonfigurasi:
	`curl "http://nginxautomode.local.dev/?[1-5]"`
	Berikut ini adalah output yang diharapkan:
	`<html><body><h1>It works!</h1></body></html>`
	`<html><body><h1>It works!</h1></body></html>`
	`<html><body><h1>It works!</h1></body></html>`
	`<html><body><h1>It works!</h1></body></html>`
	`<html><body><h1>It works!</h1></body></html>`
		DevOps insinyur, AWS DevOps

Aktifkan Mode Otomatis EKS pada cluster yang ada

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan Mode Otomatis EKS.	Ikuti petunjuk di Aktifkan Mode Otomatis EKS pada klaster yang ada (dokumentasi Amazon EKS).	AWS DevOps

Instal Pengontrol Ingress NGINX baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan instance NGINX Ingress Controller baru.	1. Unduh template deploy.yaml.
	2. Buka template deploy.yaml di editor pilihan Anda.
	3. Di `kind: Namespace` bagian ini, masukkan nama unik untuk namespace, seperti: `ingress-nginx-v2`
	`apiVersion: v1`
	`kind: Namespace`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ingress-nginx`
	`  app.kubernetes.io/name: ingress-nginx`
	`  name: ingress-nginx-v2`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Untuk setiap bagian, perbarui `namespace` nilai ke nama baru.
	5. Di `kind: Deployment` bagian ini, lakukan hal berikut:
	a. Masukkan nilai unik untuk `--controller-class`, seperti `k8s.io/ingress-nginx-v2`.
	b. Masukkan nilai unik untuk `--ingress-class`, seperti `nginx-v2`.
	`apiVersion: apps/v1`
	`kind: Deployment`
	`  name: ingress-nginx-controller`
	`  namespace: ingress-nginx-v2`
	`...`
	`    spec:`
	`      containers:`
	`      - args:`
	`        - /nginx-ingress-controller`
	`        - --publish-service=$(POD_NAMESPACE)/ingress-nginx-controller`
	`        - --election-id=ingress-nginx-leader`
	`        - --controller-class=k8s.io/ingress-nginx-v2`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>- --ingress-class=nginx-v2</pre>
	6. Di `kind: IngressClass` bagian tersebut, masukkan nilai yang sama untuk `--controller-class` dan `--ingress-class` yang Anda gunakan di bagian sebelumnya:
	<pre>apiVersion: networking.k8s.io/v1
	kind: IngressClass
	metadata:
	  labels:
	    app.kubernetes.io/component: controller
	    app.kubernetes.io/instance: ingress-nginx
	    app.kubernetes.io/name: ingress-nginx
	    app.kubernetes.io/part-of: ingress-nginx
	    app.kubernetes.io/version: 1.12.0
	  name: nginx-v2
	spec:
	  controller: k8s.io/ingress-nginx-v2</pre>
	7. Di bagian berikut, tambahkan `loadBalancerClass: eks.amazonaws.com/nlb` untuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	menyediakan Network Load Balancer untuk instance NGINX Ingress Controller:
	
	`apiVersion: v1`
	`kind: Service`
	`metadata:`
	`  name: ingress-nginx-controller`
	`  namespace: ingress-nginx-v2`
	`spec:`
	`...`
	`  selector:`
	`  app.kubernetes.io/component: controller`
	`  app.kubernetes.io/instance: ingress-nginx`
	`  app.kubernetes.io/name: ingress-nginx`
	`  type: LoadBalancer`
	`  loadBalancerClass: eks.amazonaws.com/nlb`
	
	8. Simpan dan tutup template deploy.yaml.	

Menerapkan instance NGINX Instance Controller baru.	Masukkan perintah berikut untuk menerapkan file manifes yang dimodifikasi: `kubectl apply -f deploy.yaml`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan penerapan yang berhasil.	Masukkan perintah berikut untuk memverifikasi bahwa sumber daya di `ingress-nginx-v2` namespace beroperasi:
	`kubectl get all -n ingress-nginx-v2`
	Pada output, konfirmasikan bahwa pod NGINX Ingress Controller berada dalam keadaan berjalan. Berikut ini adalah contoh output:
	`NAME READY STATUS RESTARTS AGE pod/ingress-nginx-admission-create-7shrj 0/1 Completed 0 24s pod/ingress-nginx-admission-patch-vkxr5 0/1 Completed 1 24s pod/ingress-nginx-controller-757bfcbc6d-4fw52 1/1 Running 0 24s NAME TYPE CLUSTER-IP EXTERNAL-IP`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>PORT(S)
	AGE
	service/ingress-
	nginx-controller
	LoadBalancer
	10.100.208.114
	k8s-ingressn-ingre
	ssn-2e5e37fab6-848
	337cd9c9d520f.elb.eu-
	west-1.amazonaws.com
	80:31469/TCP,443:3
	0658/TCP 24s
	service/ingress-
	nginx-controller-
	admission ClusterIP
	10.100.150.114
	<none>
	
	443/TCP
	24s
	
	NAME
	
	READY UP-TO-DATE
	AVAILABLE AGE
	deployment.apps/
	ingress-nginx-co
	ntroller 1/1 1
	1 24s
	
	NAME
	
	DESIRED CURRENT
	READY AGE
	replicaset.apps/
	ingress-nginx-co
	ntroller-757bfcbc6d</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`1 1 1 24s NAME STATUS COMPLETIONS DURATION AGE job.batch/ingress-nginx-admission-create Complete 1/1 4s 24s job.batch/ingress-nginx-admission-patch Complete 1/1 5s 24s`	
Buat ingress baru untuk beban HTTPd kerja sampel.	Masukkan perintah berikut untuk membuat ingress baru untuk beban HTTPd kerja sampel yang ada: `kubectl create ingress demo-new --class=nginx-v2 \ --rule nginxautomode.local.dev/=demo:80`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan bahwa ingress baru berfungsi.	Masukkan perintah berikut untuk mengonfirmasi bahwa ingress baru berfungsi:
	`curl -H "Host: nginxautomode.local.dev" k8s-ingressn-ingressn-2e5e37fab6-848337cd9c9d520f.elb.eu-west-1.amazonaws.com`
	Berikut ini adalah output yang diharapkan:
	`<html><body><h1>It works!</h1></body></html>`
		DevOps insinyur

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Potong ke namespace baru.	1. (Opsional) Ikuti petunjuk dalam Mengedit catatan (Dokumentasi Route 53) untuk memperbarui catatan DNS. 2. Ketika Anda telah mengonfirmasi bahwa instans NGINX Ingress Controller baru beroperasi seperti yang diharapkan, hapus yang asli.	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Hapus Pengontrol AWS Load Balancer yang dikelola sendiri. Untuk petunjuknya, lihat Memigrasi aplikasi dari ALB Ingress Controller (dokumentasi Amazon EKS) yang tidak digunakan lagi. 4. Kuras grup node terkelola . Untuk petunjuk, lihat Menghapus dan menguras grup simpul (dokumentasi eksctl).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tinjau dua masuknya.	Masukkan perintah berikut untuk meninjau dua ingress yang dibuat untuk beban kerja sampel HTTPd :
	`kubectl get ingress`
	Berikut ini adalah contoh output:
	`NAME CLASS HOSTS ADDRESS PORTS AGE`
	`demo nginx nginxautomode.local.dev k8s-ingressn-ingressn-abcdefg-12345.elb.eu-west-1.amazonaws.com 80 95m`
	`demo-new nginx-v2 nginxautomode.local.dev k8s-ingressn-ingressn-2e5e37fab6-848337cd9c9d520f.elb.eu-west-1.amazonaws.com 80 33s`
		DevOps insinyur

Sumber daya terkait

- Aktifkan Mode Otomatis EKS pada klaster yang ada (dokumentasi Amazon EKS)

- Memecahkan masalah load balancer yang dibuat oleh service controller Kubernetes di Amazon EKS (Re:Post Knowledge Center)AWS
- NGINX Ingress Controller (dokumentasi NGINX)

Migrasikan beban kerja kontainer Anda dari Azure Red Hat OpenShift (ARO) ke (ROSA Layanan OpenShift Red Hat di AWS)

Naveen Ramasamy, Srikanth Rangavajhala, dan Gireesh Sreekantan, Amazon Web Services

Ringkasan

Pola ini memberikan step-by-step instruksi untuk memigrasikan beban kerja kontainer dari Azure Red Hat OpenShift (ARO) ke Layanan OpenShift Red Hat di AWS (ROSA). ROSA adalah layanan Kubernetes terkelola yang disediakan oleh Red Hat bekerja sama dengan. AWS Ini membantu Anda menerapkan, mengelola, dan menskalakan aplikasi kontainer Anda dengan menggunakan platform Kubernetes, dan manfaat dari keahlian Red Hat di Kubernetes dan infrastrukturnya. AWS Cloud

Migrasi beban kerja kontainer dari ARO, dari cloud lain, atau dari tempat ke ROSA melibatkan transfer aplikasi, konfigurasi, dan data dari satu platform ke platform lainnya. Pola ini membantu memastikan transisi yang mulus sambil mengoptimalkan AWS Cloud layanan, keamanan, dan efisiensi biaya. Ini mencakup dua metode untuk memigrasikan beban kerja Anda ke cluster ROSA: CI/CD dan Migration Toolkit for Containers (MTC).

Pola ini mencakup kedua metode. Metode yang Anda pilih tergantung pada kompleksitas dan kepastian proses migrasi Anda. Jika Anda memiliki kontrol penuh atas status aplikasi Anda dan dapat menjamin penyiapan yang konsisten melalui pipeline, kami sarankan Anda menggunakan CI/CD metode ini. Namun, jika status aplikasi Anda melibatkan ketidakpastian, perubahan tak terduga, atau ekosistem yang kompleks, sebaiknya gunakan MTC sebagai jalur yang andal dan terkontrol untuk memigrasikan aplikasi dan datanya ke klaster baru. Untuk perbandingan rinci dari dua metode, lihat bagian Informasi tambahan.

Manfaat bermigrasi ke ROSA:

- ROSA terintegrasi dengan mulus AWS sebagai layanan asli. Ini mudah diakses melalui Konsol Manajemen AWS dan ditagih melalui satu Akun AWS. Ini menawarkan kompatibilitas penuh dengan yang lain Layanan AWS dan memberikan dukungan kolaboratif dari keduanya AWS dan Red Hat.
- ROSA mendukung penyebaran hybrid dan multi-cloud. Ini memungkinkan aplikasi berjalan secara konsisten di pusat data lokal dan beberapa lingkungan cloud.
- ROSA mendapat manfaat dari fokus keamanan Red Hat, dan menyediakan fitur seperti kontrol akses berbasis peran (RBAC), pemindaian gambar, dan penilaian kerentanan untuk memastikan lingkungan wadah yang aman.

- ROSA dirancang untuk menskalakan aplikasi dengan mudah dan menyediakan opsi ketersediaan tinggi. Ini memungkinkan aplikasi tumbuh sesuai kebutuhan sambil mempertahankan keandalan.
- ROSA mengotomatiskan dan menyederhanakan penerapan klaster Kubernetes dibandingkan dengan pengaturan manual dan metode manajemen. Ini mempercepat proses pengembangan dan penyebaran.
- ROSA mendapat manfaat dari AWS Cloud layanan, dan menyediakan integrasi tanpa batas dengan AWS penawaran seperti layanan basis data, solusi penyimpanan, dan layanan keamanan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Izin yang dikonfigurasi untuk ROSA Layanan AWS yang bergantung pada untuk memberikan fungsionalitas. Untuk informasi lebih lanjut, lihat Prasyarat dalam dokumentasi ROSA.
- ROSA diaktifkan pada konsol ROSA. Untuk instruksi, lihat dokumentasi ROSA.
- Cluster ROSA diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Memulai ROSA di dokumentasi ROSA. Untuk memahami berbagai metode untuk menyiapkan cluster ROSA, lihat Panduan AWS Preskriptif panduan strategi implementasi ROSA.
- Konektivitas jaringan dibuat dari jaringan lokal ke AWS melalui AWS Direct Connect(lebih disukai) atau AWS Virtual Private Network (Site-to-Site VPN).
- Instans Amazon Elastic Compute Cloud (Amazon EC2) atau server virtual lain untuk menginstal alat seperti`aws client`, klien OpenShift CLI `oc` (), klien ROSA, dan biner Git.

Prasyarat tambahan untuk metode ini: CI/CD

- Akses ke server Jenkins lokal dengan izin untuk membuat pipeline baru, menambahkan tahapan, menambahkan OpenShift kluster, dan melakukan build.
- Akses ke repositori Git tempat kode sumber aplikasi dipertahankan, dengan izin untuk membuat cabang Git baru dan melakukan commit ke cabang baru.

Prasyarat tambahan untuk metode MTC:

- Bucket Amazon Simple Storage Service (Amazon S3), yang akan digunakan sebagai repositori replikasi.

- Akses administratif ke cluster ARO sumber. Ini diperlukan untuk mengatur koneksi MTC.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

ROSA menyediakan tiga pola penyebaran jaringan: publik, pribadi, dan. AWS PrivateLink PrivateLinkmemungkinkan tim rekayasa keandalan situs Red Hat (SRE) untuk mengelola cluster dengan menggunakan subnet pribadi yang terhubung ke PrivateLink titik akhir cluster dalam VPC yang ada.

Memilih PrivateLink opsi menyediakan konfigurasi paling aman. Untuk alasan itu, kami merekomendasikannya untuk beban kerja yang sensitif atau persyaratan kepatuhan yang ketat. Untuk informasi tentang opsi penyebaran jaringan publik dan pribadi, lihat OpenShift dokumentasi Red Hat.

> **Important**
>
> Anda dapat membuat PrivateLink cluster hanya pada waktu instalasi. Anda tidak dapat mengubah cluster untuk digunakan PrivateLink setelah instalasi.

Diagram berikut mengilustrasikan PrivateLink arsitektur untuk klaster ROSA yang digunakan Direct Connect untuk terhubung ke lingkungan lokal dan ARO.

AWS izin untuk ROSA

Untuk AWS izin ROSA, kami sarankan Anda menggunakan AWS Security Token Service (AWS STS) dengan token dinamis yang berumur pendek. Metode ini menggunakan peran dan kebijakan yang telah ditentukan dengan hak istimewa paling sedikit untuk memberikan izin minimal ROSA untuk beroperasi di, dan mendukung instalasi ROSA, bidang kontrol Akun AWS, dan fungsionalitas komputasi.

Pemindahan pipa CI/CD

CI/CD pipeline redeployment is the recommended method for users who have a mature CI/CDpipa. Ketika Anda memilih opsi ini, Anda dapat menggunakan strategi DevOps penyebaran apa pun untuk secara bertahap mengalihkan beban aplikasi Anda ke penerapan di ROSA.

> **Note**
>
> Pola ini mengasumsikan kasus penggunaan umum di mana Anda memiliki pipeline Git, JFrog Artifactory, dan Jenkins lokal. Pendekatan ini mengharuskan Anda membuat konektivitas jaringan dari jaringan lokal ke AWS jaringan lokal Direct Connect, dan menyiapkan klaster ROSA sebelum mengikuti petunjuk di bagian Epics. Lihat bagian Prasyarat untuk detailnya.

Diagram berikut menunjukkan alur kerja untuk metode ini.

Metode MTC

Anda dapat menggunakan Migration Toolkit for Containers (MTC) untuk memigrasikan beban kerja kontainer Anda di antara lingkungan Kubernetes yang berbeda, seperti dari ARO ke ROSA. MTC menyederhanakan proses migrasi dengan mengotomatiskan beberapa tugas utama dan menyediakan kerangka kerja yang komprehensif untuk mengelola siklus hidup migrasi.

Diagram berikut menunjukkan alur kerja untuk metode ini.

Alat

Layanan AWS

- AWS DataSyncadalah layanan transfer dan penemuan data online yang membantu Anda memindahkan file atau data objek ke, dari, dan di antara layanan AWS penyimpanan.
- AWS Direct Connectmenghubungkan jaringan internal Anda ke Direct Connect lokasi melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke publik Layanan AWS sambil melewati penyedia layanan internet di jalur jaringan Anda.
- AWS PrivateLinkmembantu Anda membuat koneksi pribadi searah dari cloud pribadi virtual Anda (VPCs) ke layanan di luar VPC.
- Layanan OpenShift Red Hat di AWS (ROSA) adalah layanan terkelola yang membantu OpenShift pengguna Red Hat untuk membangun, menskalakan, dan mengelola aplikasi kontainer. AWS

- AWS Security Token Service (AWS STS) membantu Anda meminta kredensyal hak istimewa terbatas sementara untuk pengguna.

Alat-alat lainnya

- Migration Toolkit for Containers (MTC) menyediakan konsol dan API untuk memigrasi aplikasi kontainer dari ARO ke ROSA.

Praktik terbaik

- Untuk ketahanan dan jika Anda memiliki beban kerja kepatuhan keamanan, siapkan klaster ROSA multi-AZ yang menggunakan. PrivateLink Untuk informasi lebih lanjut, lihat dokumentasi ROSA.

> **Note**
>
> PrivateLink tidak dapat dikonfigurasi setelah instalasi.

- Bucket S3 yang Anda gunakan untuk repositori replikasi tidak boleh bersifat publik. Gunakan kebijakan bucket S3 yang sesuai untuk membatasi akses.
- Jika Anda memilih metode MTC, gunakan opsi migrasi Tahap untuk mengurangi jendela downtime selama cutover.
- Tinjau kuota layanan Anda sebelum dan sesudah Anda menyediakan klaster ROSA. Jika perlu, mintalah peningkatan kuota sesuai dengan kebutuhan Anda. Untuk informasi lebih lanjut, lihat dokumentasi Service Quotas.
- Tinjau pedoman keamanan ROSA dan terapkan praktik terbaik keamanan.
- Kami menyarankan Anda menghapus administrator cluster default setelah instalasi. Untuk informasi selengkapnya, lihat OpenShift dokumentasi Red Hat.
- Gunakan penskalaan otomatis kumpulan mesin untuk mengurangi node pekerja yang tidak digunakan di klaster ROSA untuk mengoptimalkan biaya. Untuk informasi lebih lanjut, lihat Lokakarya ROSA.
- Gunakan layanan Red Hat Cost Management untuk OpenShift Container Platform untuk lebih memahami dan melacak biaya cloud dan kontainer. Untuk informasi lebih lanjut, lihat Lokakarya ROSA.
- Memantau dan mengaudit layanan dan aplikasi infrastruktur klaster ROSA dengan menggunakan Layanan AWS. Untuk informasi lebih lanjut, lihat Lokakarya ROSA.

Epik

Opsi 1: Gunakan CI/CD pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan cluster ROSA baru ke Jenkins.	1. Di konsol Jenkins, pilih Kelola Jenkins, Konfigurasi Sistem. 2. Pada halaman Kelola Jenkins, di bagian OpenShift Plug-in, pilih Tambahkan cluster baru. 3. Berikan informasi yang diperlukan seperti nama cluster, URL server API, dan informasi token untuk mengautentikasi dengan ROSA.	Administrator AWS, administrator sistem AWS, AWS DevOps
Tambahkan `oc` klien ke node Jenkins Anda.	1. Pada konsol Jenkins, pilih Manage Jenkins, Global Tool Configuration. 2. Di bagian OpenShift Client Tools, instal versi klien OpenShift CLI (`oc`) yang sama dengan versi cluster ROSA Anda.	Administrator AWS, administrator sistem AWS, AWS DevOps
Buat cabang Git baru.	Buat cabang baru di repositori Git Anda untuk. `rosa-dev` Cabang ini memisahkan perubahan parameter kode atau konfigurasi untuk ROSA dari pipeline yang ada.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tag gambar untuk ROSA.	Di tahap build Anda, gunakan tag yang berbeda untuk mengidentifikasi gambar yang dibuat dari pipeline ROSA.	Administrator AWS, administrator sistem AWS, AWS DevOps
Buat pipa.	Buat pipeline Jenkins baru yang mirip dengan pipeline Anda yang ada. Untuk pipeline ini, gunakan cabang `rosa-dev` Git yang Anda buat sebelumnya, dan pastikan untuk menyertakan tahap checkout, pemindaian kode, dan build Git yang identik dengan pipeline Anda yang ada.	Administrator AWS, administrator sistem AWS, AWS DevOps
Tambahkan tahap penerapan ROSA.	Di pipeline baru, tambahkan tahapan untuk diterapkan ke cluster ROSA dan referensi cluster ROSA yang Anda tambahkan ke konfigurasi global Jenkins.	Administrator AWS, AWS DevOps, administrator sistem AWS
Mulai membangun baru.	Di Jenkins, pilih pipeline Anda dan pilih Build now, atau mulai build baru dengan melakukan perubahan ke `rosa-dev` cabang di Git.	Administrator AWS, AWS DevOps, administrator sistem AWS
Verifikasi penyebaran.	Gunakan perintah oc atau konsol ROSA untuk memverifikasi bahwa aplikasi telah digunakan pada cluster ROSA target Anda.	Administrator AWS, AWS DevOps, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin data ke cluster target.	Untuk beban kerja stateful, salin data dari cluster sumber ke cluster target dengan menggunakan AWS DataSync atau utilitas open source seperti rsync, atau Anda dapat menggunakan metode MTC. Lihat informasi yang lebih lengkap dalam dokumentasi AWS DataSync.	Administrator AWS, AWS DevOps, administrator sistem AWS
Uji aplikasi Anda.	1. Ambil URL rute untuk aplikasi Anda dengan menggunakan perintah rute oc atau konsol ROSA. 2. Uji aplikasi Anda dengan menggunakan URL rute.	Administrator AWS, AWS DevOps, administrator sistem AWS
Potong.	Jika pengujian Anda berhasil, gunakan kebijakan Amazon Route 53 yang sesuai untuk memindahkan lalu lintas dari aplikasi yang dihosting ARO ke aplikasi yang dihosting Rosa-host. Saat Anda menyelesaikan langkah ini, beban kerja aplikasi Anda akan sepenuhnya beralih ke cluster ROSA.	Administrator AWS, administrator sistem AWS

Opsi 2: Gunakan MTC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal operator MTC.	Instal operator MTC pada cluster ARO dan ROSA: 1. Di konsol ARO atau ROSA, arahkan ke OperatorH ubhalaman Operator. 2. Di kotak Filter menurut kata kunci, temukan atau ketik MTC. 3. Pilih operator MTC untuk menampilkan informasi tambahan. 4. Setelah Anda membaca informasi tentang operator, pilih Instal.	Administrator AWS, AWS DevOps, administrator sistem AWS
Konfigurasikan lalu lintas jaringan ke repositori replikasi.	Jika Anda menggunakan server proxy, konfigura sikan untuk memungkinkan lalu lintas jaringan antara repositori replikasi dan cluster. Repositori replikasi adalah objek penyimpanan perantara yang digunakan MTC untuk memigrasikan data. Cluster sumber dan target harus memiliki akses jaringan ke repositori replikasi selama migrasi.	Administrator AWS, AWS DevOps, administrator sistem AWS
Tambahkan cluster sumber ke MTC.	Di konsol web MTC, tambahkan cluster sumber ARO.	Administrator AWS, AWS DevOps, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan Amazon S3 sebagai repositori replikasi Anda.	Di konsol web MTC, tambahkan bucket Amazon S3 (lihat Prasyarat) sebagai repositori replikasi.	Administrator AWS, AWS DevOps, administrator sistem AWS
Buat rencana migrasi.	Di konsol web MTC, buat rencana migrasi dan tentukan jenis transfer data sebagai Salin. Ini akan menginstr uksikan MTC untuk menyalin data dari cluster sumber (ARO) ke bucket S3, dan dari bucket ke cluster target (ROSA).	Administrator AWS, AWS DevOps, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan rencana migrasi.	Jalankan paket migrasi dengan menggunakan opsi Stage atau Cutover: • Pilih Stage untuk menyalin data ke cluster target tanpa menghentikan aplikasi Anda. Anda dapat menjalankan beberapa migrasi tahap untuk menyalin sebagian besar data Anda sebelum migrasi cutover. Ini mempersingkat durasi migrasi cutover. • Pilih Cutover untuk menghentikan aplikasi Anda di cluster sumber sambil memindahkan resource ke cluster target. Untuk mencegah penghenti an aplikasi, Anda dapat menghapus kotak centang Hentikan transaksi di cluster sumber selama migrasi.	Administrator AWS, AWS DevOps, administrator sistem AWS

Pemecahan masalah

Isu	Solusi
Masalah konektivitas	Saat memigrasikan beban kerja kontainer dari ARO ke ROSA, Anda mungkin mengalami masalah konektivitas yang harus diselesai kan untuk memastikan migrasi berhasil. Untuk

Isu	Solusi
	mengatasi masalah konektivitas ini (tercantum dalam tabel ini) selama migrasi, perencanaan yang cermat, koordinasi dengan jaringan dan tim keamanan Anda, dan pengujian menyeluruh sangat penting. Menerapkan strategi migrasi bertahap dan memverifikasi konektivitas di setiap langkah akan membantu meminimalkan potensi gangguan dan memastikan transisi yang mulus dari ARO ke ROSA.
Perbedaan konfigurasi jaringan	ARO dan ROSA mungkin memiliki variasi dalam konfigurasi jaringan mereka, seperti pengaturan jaringan virtual (VNet), subnet, dan kebijakan jaringan. Untuk komunikasi yang tepat antar layanan, pastikan pengaturan jaringan sejajar antara kedua platform.
Grup keamanan dan aturan firewall	ROSA dan ARO mungkin memiliki grup keamanan default dan pengaturan firewall yang berbeda. Pastikan untuk menyesuaikan dan memperbarui aturan ini untuk mengizinkan lalu lintas yang diperlukan untuk menjaga konektivitas antar kontainer dan layanan selama migrasi.
Alamat IP dan perubahan DNS	Saat Anda memigrasikan beban kerja, alamat IP dan nama DNS mungkin berubah. Konfigurasikan ulang aplikasi yang mengandalkan nama DNS statis IPs atau spesifik.
Akses layanan eksternal	Jika aplikasi Anda bergantung pada layanan eksternal seperti database atau APIs, Anda mungkin harus memperbarui pengaturan koneksi mereka untuk memastikan mereka dapat berkomunikasi dengan layanan baru dari ROSA.

Isu	Solusi
Konfigurasi Azure Private Link	Jika Anda menggunakan Azure Private Link atau layanan endpoint pribadi di ARO, Anda perlu menyiapkan fungsionalitas yang setara di ROSA untuk memastikan konektivitas pribadi antar sumber daya.
Site-to-Site VPN atau Direct Connect pengaturan	Jika ada Site-to-Site VPN atau Direct Connect koneksi antara jaringan lokal dan ARO, Anda perlu membuat koneksi serupa dengan ROSA untuk komunikasi tanpa gangguan dengan sumber daya lokal Anda.
Pengaturan masuk dan penyeimbang beban	Konfigurasi untuk pengontrol masuk dan penyeimbang beban mungkin berbeda antara ARO dan ROSA. Konfigurasikan ulang pengaturan ini untuk mempertahankan akses eksternal ke layanan Anda.
Sertifikat dan penanganan TLS	Jika aplikasi Anda menggunakan sertifikat SSL atau TLS, pastikan sertifikat tersebut valid dan dikonfigurasi dengan benar di ROSA.
Akses registri kontainer	Jika kontainer Anda di-host di registri kontainer eksternal, siapkan autentikasi yang tepat dan izin akses untuk ROSA.
Pencatatan dan pemantauan	Perbarui konfigurasi pemantauan dan pencatatan untuk mencerminkan infrastruktur baru di ROSA sehingga Anda dapat terus memantau kesehatan dan kinerja kontainer Anda secara efektif.

Sumber daya terkait

AWSreferensi

- Apa itu Layanan OpenShift Red Hat di AWS? (Dokumentasi ROSA)
- Memulai dengan ROSA (dokumentasi ROSA)
- Layanan OpenShift Red Hat di AWS strategi implementasi (Panduan AWS Preskriptif)
- Layanan OpenShift Red Hat di AWS Sekarang GA (posting AWS blog)
- Lokakarya ROSA
- ROSA FAQ
- FAQ Lokakarya ROSA
- Harga ROSA

OpenShift Dokumentasi Red Hat

- Memasang cluster dengan cepat AWS
- Memasang cluster AWS di jaringan terbatas
- Menginstal cluster AWS ke dalam VPC yang ada
- Menginstal klaster pada infrastruktur yang disediakan pengguna dengan menggunakan templat AWS CloudFormation
- Menginstal cluster AWS di jaringan terbatas dengan infrastruktur yang disediakan pengguna
- Menginstal cluster AWS dengan kustomisasi
- Memulai dengan OpenShift CLI

Informasi tambahan

Memilih antara opsi pemindahan MTC dan CI/CD pipa

Migrasi aplikasi dari satu OpenShift cluster ke cluster lainnya membutuhkan pertimbangan yang cermat. Idealnya, Anda menginginkan transisi yang mulus dengan menggunakan CI/CD pipeline untuk menerapkan ulang aplikasi dan menangani migrasi data volume persisten. Namun, dalam praktiknya, aplikasi yang berjalan di cluster rentan terhadap perubahan tak terduga dari waktu ke waktu. Perubahan ini dapat menyebabkan aplikasi secara bertahap menyimpang dari status penerapan aslinya. MTC menawarkan solusi untuk skenario di mana konten yang tepat dari namespace tidak pasti dan migrasi mulus dari semua komponen aplikasi ke cluster baru adalah penting.

Membuat pilihan yang tepat membutuhkan evaluasi skenario spesifik Anda dan menimbang manfaat dari setiap pendekatan. Dengan demikian, Anda dapat memastikan migrasi yang sukses dan mulus

yang selaras dengan kebutuhan dan prioritas Anda. Berikut adalah panduan tambahan untuk memilih di antara dua opsi.

Pemindahan pipa CI/CD

Metode CI/CD pipeline adalah pendekatan yang disarankan jika aplikasi Anda dapat digunakan kembali dengan percaya diri menggunakan pipeline. Ini memastikan bahwa migrasi dikontrol, dapat diprediksi, dan selaras dengan praktik penerapan yang ada. Saat Anda memilih metode ini, Anda dapat menggunakan penerapan biru/hijau atau strategi penerapan kenari untuk secara bertahap mengalihkan beban ke penerapan di ROSA. Untuk skenario ini, pola ini mengasumsikan bahwa Jenkins mengatur penerapan aplikasi dari lingkungan lokal.

Keuntungan:

- Anda tidak memerlukan akses administratif ke cluster ARO sumber atau perlu menyebarkan operator apa pun di cluster sumber atau tujuan.
- Pendekatan ini membantu Anda beralih lalu lintas secara bertahap dengan menggunakan DevOps strategi.

Kekurangan:

- Dibutuhkan lebih banyak upaya untuk menguji fungsionalitas aplikasi Anda.
- Jika aplikasi Anda berisi data persisten, diperlukan langkah-langkah tambahan untuk menyalin data dengan menggunakan AWS DataSync atau alat lainnya.

Migrasi MTC

Di dunia nyata, menjalankan aplikasi dapat mengalami perubahan tak terduga yang menyebabkan mereka menjauh dari penerapan awal. Pilih opsi MTC ketika Anda tidak yakin tentang status aplikasi Anda saat ini di cluster sumber. Misalnya, jika ekosistem aplikasi Anda mencakup berbagai komponen, konfigurasi, dan volume penyimpanan data, sebaiknya pilih MTC untuk memastikan migrasi lengkap yang mencakup aplikasi dan seluruh lingkungannya.

Keuntungan:

- MTC menyediakan pencadangan lengkap dan pemulihan beban kerja.
- Ini akan menyalin data persisten dari sumber ke target saat memigrasikan beban kerja.
- Itu tidak memerlukan akses ke repositori kode sumber.

Kekurangan:

- Anda memerlukan hak administratif untuk menginstal operator MTC pada cluster sumber dan tujuan.
- DevOps Tim membutuhkan pelatihan untuk menggunakan alat MTC dan melakukan migrasi.

Jalankan tugas Amazon ECS di Amazon WorkSpaces dengan Amazon ECS Anywhere

Akash Kumar, Amazon Web Services

Ringkasan

Amazon Elastic Container Service (Amazon ECS) Anywhere mendukung penerapan tugas Amazon ECS di lingkungan apa pun, termasuk infrastruktur terkelola Amazon Web Services (AWS) dan infrastruktur yang dikelola pelanggan. Anda dapat melakukan ini saat menggunakan pesawat kontrol terkelola AWS sepenuhnya yang berjalan di cloud dan selalu up to date.

Perusahaan sering menggunakan Amazon WorkSpaces untuk mengembangkan aplikasi berbasis kontainer. Ini memerlukan Amazon Elastic Compute Cloud (Amazon EC2) atau AWS Fargate dengan cluster Amazon ECS untuk menguji dan menjalankan tugas ECS. Sekarang, dengan menggunakan Amazon ECS Anywhere, Anda dapat menambahkan WorkSpaces Amazon sebagai instans eksternal langsung ke cluster ECS, dan Anda dapat menjalankan tugas secara langsung. Ini mengurangi waktu pengembangan Anda, karena Anda dapat menguji container Anda dengan cluster ECS secara lokal di Amazon. WorkSpaces Anda juga dapat menghemat biaya penggunaan EC2 atau instance Fargate untuk menguji aplikasi kontainer Anda.

Pola ini menampilkan cara menerapkan tugas ECS di Amazon WorkSpaces dengan Amazon ECS Anywhere. Ini mengatur cluster ECS dan menggunakan AWS Directory Service Simple AD untuk meluncurkan file. WorkSpaces Kemudian contoh tugas ECS meluncurkan NGINX di. WorkSpaces

Prasyarat dan batasan

- Akun AWS yang aktif
- Antarmuka Baris Perintah AWS (AWS CLI)
- Kredensi AWS dikonfigurasi pada mesin Anda

Arsitektur

Tumpukan teknologi target

- Awan pribadi virtual (VPC)
- Kluster Amazon ECS
- Amazon WorkSpaces

- AWS Directory Service dengan Simple AD

Arsitektur target

Arsitektur mencakup layanan dan sumber daya berikut:

- Cluster ECS dengan subnet publik dan pribadi dalam VPC khusus
- Simple AD di VPC untuk menyediakan akses pengguna ke Amazon WorkSpaces
- Amazon WorkSpaces disediakan di VPC menggunakan Simple AD
- AWS Systems Manager diaktifkan untuk menambahkan Amazon WorkSpaces sebagai instans terkelola
- Menggunakan Amazon ECS dan AWS Systems Manager Agent (SSM Agent), Amazon WorkSpaces ditambahkan ke Systems Manager dan cluster ECS
- Contoh tugas ECS untuk dijalankan di WorkSpaces dalam cluster ECS

Alat

- AWS Directory Service Simple Active Directory (Simple AD) adalah direktori terkelola mandiri yang didukung oleh Server Kompatibel Direktori Aktif Samba 4. Simple AD menyediakan subset fitur yang ditawarkan oleh AWS Managed Microsoft AD, termasuk kemampuan untuk mengelola pengguna dan terhubung dengan aman ke instans Amazon EC2 .
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan terukur yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Systems Manager membantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan manajemen aplikasi dan sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola sumber daya AWS Anda dengan aman dalam skala besar.
- Amazon WorkSpaces membantu Anda menyediakan desktop Microsoft Windows atau Amazon Linux virtual berbasis cloud untuk pengguna Anda, yang dikenal sebagai. WorkSpaces

WorkSpaces menghilangkan kebutuhan untuk mendapatkan dan menyebarkan perangkat keras atau menginstal perangkat lunak yang kompleks.

Epik

Siapkan cluster ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan cluster ECS.	Untuk membuat cluster ECS, ikuti petunjuk dalam dokumentasi AWS, termasuk langkah-langkah berikut: • Untuk kompatibilitas klaster Pilih, pilih Jaringan saja, yang akan mendukung Amazon WorkSpace sebagai instans eksternal ke cluster ECS. • Pilih untuk membuat VPC baru.	Arsitek awan

Luncurkan Amazon WorkSpaces

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Simple AD dan luncurkan Amazon WorkSpaces.	Untuk menyediakan direktori Simple AD untuk VPC yang baru dibuat dan meluncurkan Amazon WorkSpaces, ikuti petunjuk dalam dokumentasi AWS.	Arsitek awan

Menyiapkan AWS Systems Manager untuk lingkungan hybrid

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh skrip terlampir.	Di mesin lokal Anda, unduh `ssm-activation.json` file `ssm-trust-policy.json` dan file yang ada di bagian Lampiran.	Arsitek awan

Tambahkan peran IAM.	Tambahkan variabel lingkungan berdasarkan kebutuhan bisnis Anda.
	`export AWS_DEFAULT_REGION=${AWS_REGION_ID}`
	`export ROLE_NAME=${ECS_TASK_ROLE}`
	`export CLUSTER_NAME=${ECS_CLUSTER_NAME}`
	`export SERVICE_NAME=${ECS_CLUSTER_SERVICE_NAME}`
	Jalankan perintah berikut.
	`aws iam create-role --role-name $ROLE_NAME --assume-role-policy-document file://ssm-trust-policy.json`
		Arsitek awan

Tambahkan SSMManaged InstanceCore kebijakan Amazon ke peran IAM.	Jalankan perintah berikut. `aws iam attach-role-policy --role-name $ROLE_NAME --policy-arn arn:aws:iam::aws:p`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`olicy/AmazonSSMMan agedInstanceCore`	
Tambahkan kebijakan EC2 ContainerServicefor EC2 Peran Amazon ke peran IAM.	Jalankan perintah berikut. `aws iam attach-role-policy --role-name $ROLE_NAME --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role`	Arsitek awan
Verifikasi peran IAM.	Untuk memverifikasi peran IAM, jalankan perintah berikut. `aws iam list-attached-role-policies --role-name $ROLE_NAME`	Arsitek awan
Aktifkan Systems Manager.	Jalankan perintah berikut. `aws ssm create-activation --iam-role $ROLE_NAME | tee ssm-activation.json`	Arsitek awan

Tambahkan WorkSpaces ke cluster ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke Anda WorkSpaces.	Untuk menyambung dan mengatur Ruang Kerja Anda, ikuti petunjuk dalam dokumentasi AWS.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh skrip penginstalan ecs-anywhere.	Pada prompt perintah, jalankan perintah berikut: `curl -o "ecs-anywhere-install.sh" "https://amazon-ecs-agent-packages-preview.s3.us-east-1.amazonaws.com/ecs-anywhere-install.sh" && sudo chmod +x ecs-anywhere-install.sh`	Pengembang aplikasi
Periksa integritas skrip shell.	(Opsional) Jalankan perintah berikut. `curl -o "ecs-anywhere-install.sh.sha256" "https://amazon-ecs-agent-packages-preview.s3.us-east-1.amazonaws.com/ecs-anywhere-install.sh.sha256" && sha256sum -c ecs-anywhere-install.sh.sha256`	Pengembang aplikasi
Tambahkan repositori EPEL di Amazon Linux.	Untuk menambahkan repositori Paket Ekstra untuk Enterprise Linux (EPEL), jalankan perintah. `sudo amazon-linux-extras install epel -y`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Amazon ECS Anywhere.	Untuk menjalankan skrip instal, gunakan perintah berikut. `sudo ./ecs-anywhere-install.sh --cluster $CLUSTER_NAME --activation-id $ACTIVATION_ID --activation-code $ACTIVATION_CODE --region $AWS_REGION`	
Periksa informasi instance dari cluster ECS.	Untuk memeriksa informasi instans Systems Manager dan ECS cluster dan memvalidasi yang WorkSpaces ditambahkan pada cluster, jalankan perintah berikut dari mesin lokal Anda. `aws ssm describe-instance-information" && "aws ecs list-container-instances --cluster $CLUSTER_NAME`	Pengembang aplikasi

Tambahkan tugas ECS untuk WorkSpaces

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM eksekusi tugas.	Unduh `task-execution-assume-role.json` dan `external-task-defi`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`nition.json` dari bagian Lampiran. Pada mesin lokal Anda, jalankan perintah berikut. `aws iam --region $AWS_DEFAULT_REGION create-role --role-name $ECS_TASK_EXECUTION_ROLE --assume-role-policy-document file://task-execution-assume-role.json`	
Tambahkan kebijakan ke peran eksekusi.	Jalankan perintah berikut. `aws iam --region $AWS_DEFAULT_REGION attach-role-policy --role-name $ECS_TASK_EXECUTION_ROLE --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran tugas.	Jalankan perintah berikut. `aws iam --region $AWS_DEFAULT_REGION create-role --role-name $ECS_TASK_EXECUTION_ROLE --assume-role-policy-document file://task-execution-assume-role.json`	Arsitek awan
Daftarkan definisi tugas ke cluster.	Pada mesin lokal Anda, jalankan perintah berikut. `aws ecs register-task-definition --cli-input-json file://external-task-definition.json`	Arsitek awan
Jalankan tugas.	Pada mesin lokal Anda, jalankan perintah berikut. `aws ecs run-task --cluster $CLUSTER_NAME --launch-type EXTERNAL --task-definition nginx`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi status tugas yang sedang berjalan.	Untuk mengambil ID tugas, jalankan perintah berikut.
	`export TEST_TASKID= $(aws ecs list-tasks -- cluster $CLUSTER_NAME | jq -r '.taskArns[0]')`
	Dengan ID tugas, jalankan perintah berikut.
	`aws ecs describe-tasks --cluster $CLUSTER_ NAME --tasks ${TEST_TA SKID}`
		Arsitek awan

Verifikasi tugas di WorkSpace.	Untuk memeriksa bahwa NGINX berjalan pada WorkSpace, jalankan perintah. `curl http://lo calhost:8080`	Pengembang aplikasi

Sumber daya terkait

- Cluster ECS
- Menyiapkan lingkungan hibrida
- Amazon WorkSpaces
- Sederhana AD

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Jalankan wadah API Docker web ASP.NET Core pada instance Amazon Linux EC2

Vijai Anand Ramalingam dan Sreelaxmi Pai, Amazon Web Services

Ringkasan

Pola ini diperuntukkan bagi orang-orang yang mulai mengkontainerisasi aplikasi mereka di Amazon Web Services (AWS) Cloud. Saat Anda mulai membuat container aplikasi di cloud, biasanya tidak ada platform orkestrasi kontainer yang disiapkan. Pola ini membantu Anda menyiapkan infrastruktur AWS dengan cepat untuk menguji aplikasi kontainer Anda tanpa memerlukan infrastruktur orkestrasi kontainer yang rumit.

Langkah pertama dalam perjalanan modernisasi adalah mengubah aplikasi. Jika ini adalah aplikasi .NET Framework lama, Anda harus terlebih dahulu mengubah runtime menjadi ASP.NET Core. Kemudian, lakukan hal berikut:

- Buat gambar kontainer Docker
- Jalankan wadah Docker menggunakan gambar yang dibangun
- Validasi aplikasi sebelum menerapkannya pada platform orkestrasi kontainer apa pun, seperti Amazon Elastic Container Service (Amazon ECS) Service Elastic Container (Amazon ECS) atau Amazon Elastic Kubernetes Service (Amazon EKS).

Pola ini mencakup aspek build, run, dan validasi pengembangan aplikasi modern pada instance Amazon Elastic Compute Cloud EC2 (Amazon) Linux.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif
- Peran AWS Identity and Access Management (IAM) dengan akses yang memadai untuk membuat sumber daya AWS untuk pola ini
- Visual Studio Community 2022 atau yang lebih baru diunduh dan diinstal
- Proyek .NET Framework dimodernisasi menjadi ASP.NET Core
- Sebuah GitHub repositori

Versi produk

- Komunitas Visual Studio 2022 atau yang lebih baru

Arsitektur

Arsitektur target

Pola ini menggunakan CloudFormation template AWS untuk membuat arsitektur yang sangat tersedia yang ditunjukkan pada diagram berikut. Instans Amazon EC2 Linux diluncurkan di subnet pribadi. AWS Systems Manager Session Manager digunakan untuk mengakses instans Amazon EC2 Linux pribadi dan untuk menguji API yang berjalan di container Docker.

1. Akses ke instance Linux melalui Session Manager

Alat

Layanan AWS

- AWS Command Line Interface — AWS Command Line Interface (AWS CLI) adalah alat open source untuk berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda. Dengan konfigurasi minimal, Anda dapat menjalankan perintah AWS CLI yang menerapkan fungsionalitas yang setara dengan yang disediakan oleh AWS Management Console berbasis browser.
- AWS Management Console — AWS Management Console adalah aplikasi web yang terdiri dan mengacu pada kumpulan konsol layanan yang luas untuk mengelola sumber daya AWS. Saat pertama kali masuk, Anda akan melihat halaman beranda konsol. Halaman beranda menyediakan akses ke setiap konsol layanan dan menawarkan satu tempat untuk mengakses informasi yang Anda perlukan untuk melakukan tugas terkait AWS Anda.
- AWS Systems Manager Session Manager — Session Manager adalah kemampuan AWS Systems Manager yang dikelola sepenuhnya. Dengan Session Manager, Anda dapat mengelola instans Amazon Elastic Compute Cloud (Amazon EC2). Session Manager menyediakan manajemen node yang aman dan dapat diaudit tanpa perlu membuka port masuk, memelihara host bastion, atau mengelola kunci SSH.

Alat lainnya

- Visual Studio 2022 — Visual Studio 2022 adalah lingkungan pengembangan terintegrasi (IDE).
- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.

Kode

```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
 WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["DemoNetCoreWebAPI/DemoNetCoreWebAPI.csproj", "DemoNetCoreWebAPI/"]
RUN dotnet restore "DemoNetCoreWebAPI/DemoNetCoreWebAPI.csproj"
COPY . .
WORKDIR "/src/DemoNetCoreWebAPI"
RUN dotnet build "DemoNetCoreWebAPI.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "DemoNetCoreWebAPI.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "DemoNetCoreWebAPI.dll"]
```

Epik

Kembangkan API web ASP.NET Core

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh ASP.NET Core web API menggunakan Visual Studio.	Untuk membuat contoh ASP.NET Core web API, lakukan hal berikut: 1. Buka Visual Studio 2022. 2. Pilih Buat proyek baru.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Pilih template proyek ASP.NET Core Web API, dan pilih Berikutnya.
	4. Untuk nama proyek, masukkan DemoNetCo reWebAPI, dan pilih Berikutnya.
	5. Pilih Buat.
	6. Untuk menjalankan proyek secara lokal, tekan F5.
	7. Verifikasi bahwa titik akhir WeatherForecastAPI default mengembalikan hasil menggunakan Swagger.
	8. Buka prompt perintah, arahkan ke folder proyek.cs proj, dan jalankan perintah berikut untuk mendorong API web baru ke repositori Anda. GitHub
	`git add --all`
	`git commit -m "Initial Version"`
	`git push`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat Dockerfile.	Untuk membuat Dockerfile, lakukan salah satu hal berikut:
	• Buat Dockerfile secara manual menggunak an contoh Dockerfile di bagian Kode. Berdasarkan persyaratan, pilih gambar dasar.NET yang sesuai. Untuk informasi tentang gambar.NET dan ASP.NET Core terkait, lihat Docker hub.
	• Buat Dockerfile menggunak an Visual Studio dan Docker Desktop. Di explorer solusi, klik kanan pada proyek, pilih Add -> Docker Support. Untuk OS Target, pilih Linux. Pastikan bahwa Dockerfile baru berada di jalur yang sama dengan file solusi (.sln).
	
	Untuk mendorong perubahan ke GitHub repositori Anda, jalankan perintah berikut.
	`git add --all`
	`git commit -m "Dockerfi le added"`
	`git push`
		Pengembang aplikasi

Siapkan instans Amazon EC2 Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyiapkan infrastruktur.	Luncurkan CloudFormation template AWS untuk membuat infrastruktur, yang mencakup hal-hal berikut: • Virtual private cloud (VPC), menggunakan AWS VPC Quick Start, dengan dua subnet publik dan dua subnet pribadi yang mencakup dua Availability Zone. • Peran IAM yang diperluka n untuk mengaktifkan AWS Systems Manager. • Di salah satu subnet pribadi, contoh demo Amazon Linux 2 dengan Agen SSM terbaru. Meskipun instans ini tidak memiliki konektivi tas langsung dari internet, instans ini dapat diakses dengan aman menggunak an AWS Systems Manager Session Manager tanpa memerlukan host bastion. **Note** Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkap	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nya, lihat Amazon Linux 2 FAQs. Untuk mempelajari selengkap nya tentang mengakses EC2 instans Amazon pribadi menggunakan Session Manager tanpa memerlukan host bastion, lihat postingan blog Menuju dunia tanpa benteng.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke instans Amazon EC2 Linux.	Untuk terhubung ke instance Amazon EC2 Linux di subnet pribadi, lakukan hal berikut: 1. Buka EC2 konsol Amazon. 2. Di panel navigasi, pilih Instans. 3. Pilih instans demo Amazon Linux 2, dan pilih Connect. **Note** Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs. 4. Pilih Session Manager. 5. Pilih Connect untuk membuka jendela terminal baru. 6. Jalankan perintah berikut. `sudo su`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal dan mulai Docker.	Untuk menginstal dan memulai Docker di instance Amazon EC2 Linux, lakukan hal berikut:
	1. Untuk menginstal Docker, jalankan perintah berikut.
	`yum install -y docker`
	2. Untuk memulai layanan Docker, jalankan perintah berikut.
	`service docker start`
	3. Untuk memverifikasi instalasi Docker, jalankan perintah berikut.
	`docker info`
		Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal Git dan kloning repositori.	Untuk menginstal Git pada instance Amazon EC2 Linux dan mengkloning repositori dari GitHub, lakukan hal berikut.
	1. Untuk menginstal Git, jalankan perintah berikut.
	`yum install git -y`
	2. Untuk mengkloning repositori, jalankan perintah berikut.
	`git clone https://github.com/<username>/<repo-name>.git`
	3. Untuk menavigasi ke Dockerfile, jalankan perintah berikut.
	`cd <repo-name>/DemoNetCoreWebAPI/`
		Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bangun dan jalankan wadah Docker.	Untuk membuat image Docker dan menjalankan container di dalam instance Amazon EC2 Linux, lakukan hal berikut:
	1. Untuk membuat image Docker, jalankan perintah berikut.
	`docker build -t aspnetcorewebapiimage -f Dockerfile .`
	2. Untuk melihat semua gambar Docker, jalankan perintah berikut.
	`docker images`
	3. Untuk membuat dan menjalankan wadah, jalankan perintah berikut.
	`docker run -d -p 80:80 --name aspnetcorewebapicontainer aspnetcorewebapiimage`
		Pengembang aplikasi, administrator AWS, AWS DevOps

Uji API web

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji API web menggunakan perintah curl.	Untuk menguji API web, jalankan perintah berikut.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`curl -X GET "http://localhost/WeatherForecast" -H "accept: text/plain"` Verifikasi respons API. **Note** Anda bisa mendapatkan perintah curl untuk setiap titik akhir dari Swagger saat Anda menjalankannya secara lokal.	

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus semua sumber daya.	Hapus tumpukan untuk menghapus semua sumber daya. Ini memastikan bahwa Anda tidak dikenakan biaya untuk layanan apa pun yang tidak Anda gunakan.	Administrator AWS, AWS DevOps

Sumber daya terkait

- Connect ke instans Linux Anda dari Windows menggunakan PuTTY
- Buat API web dengan ASP.NET Core
- Menuju dunia tanpa benteng

Jalankan beban kerja stateful dengan penyimpanan data persisten dengan menggunakan Amazon EFS di Amazon EKS dengan AWS Fargate

Ricardo Morais, Rodrigo Bersa, dan Lucio Pereira, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk mengaktifkan Amazon Elastic File System (Amazon EFS) sebagai perangkat penyimpanan untuk container yang berjalan di Amazon Elastic Kubernetes Service (Amazon EKS) dengan menggunakan AWS Fargate untuk menyediakan sumber daya komputasi Anda.

Pengaturan yang dijelaskan dalam pola ini mengikuti praktik terbaik keamanan dan memberikan keamanan saat istirahat dan keamanan dalam perjalanan secara default. Untuk mengenkripsi sistem file Amazon EFS Anda, ia menggunakan kunci AWS Key Management Service (AWS KMS), tetapi Anda juga dapat menentukan alias kunci yang mengirimkan proses pembuatan kunci KMS.

Anda dapat mengikuti langkah-langkah dalam pola ini untuk membuat namespace dan profil Fargate untuk aplikasi proof-of-concept (PoC), menginstal driver Amazon EFS Container Storage Interface (CSI) yang digunakan untuk mengintegrasikan cluster Kubernetes dengan Amazon EFS, mengkonfigurasi kelas penyimpanan, dan menyebarkan aplikasi PoC. Langkah-langkah ini menghasilkan sistem file Amazon EFS yang dibagi di antara beberapa beban kerja Kubernetes, berjalan di atas Fargate. Pola ini disertai dengan skrip yang mengotomatiskan langkah-langkah ini.

Anda dapat menggunakan pola ini jika Anda menginginkan persistensi data dalam aplikasi kontainer Anda dan ingin menghindari kehilangan data selama operasi penskalaan. Contoh:

- DevOps alat — Skenario umum adalah mengembangkan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) strategy. In this case, you can use Amazon EFS as a shared file system to store configurations among different instances of the CI/CD tool or to store a cache (for example, an Apache Maven repository) for pipeline stages among different instances of the CI/CDalat.
- Server web — Skenario umum adalah menggunakan Apache sebagai server web HTTP. Anda dapat menggunakan Amazon EFS sebagai sistem file bersama untuk menyimpan file statis yang dibagikan di antara berbagai instance server web. Dalam skenario contoh ini, modifikasi diterapkan langsung ke sistem file alih-alih file statis yang dimasukkan ke dalam gambar Docker.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Cluster Amazon EKS yang sudah ada dengan Kubernetes versi 1.17 atau yang lebih baru (diuji hingga versi 1.27)
- Sistem file Amazon EFS yang sudah ada untuk mengikat Kubernetes StorageClass dan menyediakan sistem file secara dinamis
- Izin administrasi klaster
- Konteks dikonfigurasi untuk menunjuk ke cluster Amazon EKS yang diinginkan

Batasan

- Ada beberapa batasan yang perlu dipertimbangkan saat Anda menggunakan Amazon EKS dengan Fargate. Misalnya, penggunaan beberapa konstruksi Kubernetes, seperti DaemonSets dan kontainer istimewa, tidak didukung. Untuk informasi selengkapnya, tentang batasan Fargate, lihat pertimbangan AWS Fargate dalam dokumentasi Amazon EKS.
- Kode yang disediakan dengan pola ini mendukung workstation yang menjalankan Linux atau macOS.

Versi produk

- AWS Command Line Interface (AWS CLI) versi 2 atau yang lebih baru
- Driver Amazon EFS CSI versi 1.0 atau yang lebih baru (diuji hingga versi 2.4.8)
- eksctl versi 0.24.0 atau yang lebih baru (diuji hingga versi 0.158.0)
- jq versi 1.6 atau yang lebih baru
- kubectl versi 1.17 atau yang lebih baru (diuji hingga versi 1.27)
- Kubernetes versi 1.17 atau yang lebih baru (diuji hingga versi 1.27)

Arsitektur

Arsitektur target terdiri dari infrastruktur berikut:

- Awan pribadi virtual (VPC)
- Dua Zona Ketersediaan
- Subnet publik dengan gateway NAT yang menyediakan akses internet

- Subnet pribadi dengan cluster Amazon EKS dan target pemasangan Amazon EFS (juga dikenal sebagai titik pemasangan)
- Amazon EFS di tingkat VPC

Berikut ini adalah infrastruktur lingkungan untuk cluster Amazon EKS:

- Profil AWS Fargate yang mengakomodasi konstruksi Kubernetes di tingkat namespace
- Namespace Kubernetes dengan:
  - Dua pod aplikasi didistribusikan di seluruh Availability Zone
  - Satu klaim volume persisten (PVC) terikat pada volume persisten (PV) di tingkat cluster
- PV di seluruh cluster yang terikat pada PVC di namespace dan yang menunjuk ke target pemasangan Amazon EFS di subnet pribadi, di luar cluster

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang dapat Anda gunakan untuk berinteraksi dengan layanan AWS dari baris perintah.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di AWS Cloud. Dalam pola ini, ia menyediakan sistem file yang sederhana, terukur, dikelola sepenuhnya, dan dibagikan untuk digunakan dengan Amazon EKS.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa perlu menginstal atau mengoperasikan cluster Anda sendiri.
- AWS Fargate adalah mesin komputasi tanpa server untuk Amazon EKS. Ini membuat dan mengelola sumber daya komputasi untuk aplikasi Kubernetes Anda.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- eksctl adalah utilitas baris perintah untuk membuat dan mengelola cluster Kubernetes di Amazon EKS.

- kubectl adalah antarmuka baris perintah yang membantu Anda menjalankan perintah terhadap klaster Kubernetes.
- jq adalah alat baris perintah untuk mengurai JSON.

Kode

Kode untuk pola ini disediakan dalam Konfigurasi GitHub Persistence dengan Amazon EFS di Amazon EKS menggunakan repo AWS Fargate. Skrip diatur oleh epik, di folder `epic01` melaluiepic06, sesuai dengan urutan di bagian Epik dalam pola ini.

Praktik terbaik

Arsitektur target mencakup layanan dan komponen berikut, dan mengikuti praktik terbaik AWS Well-Architected Framework:

- Amazon EFS, yang menyediakan sistem file NFS elastis yang sederhana, dapat diskalakan, dan dikelola sepenuhnya. Ini digunakan sebagai sistem file bersama di antara semua replikasi aplikasi PoC yang berjalan di pod, yang didistribusikan dalam subnet pribadi dari kluster Amazon EKS yang dipilih.
- Target pemasangan Amazon EFS untuk setiap subnet pribadi. Ini memberikan redundansi per Availability Zone dalam virtual private cloud (VPC) cluster.
- Amazon EKS, yang menjalankan beban kerja Kubernetes. Anda harus menyediakan kluster Amazon EKS sebelum menggunakan pola ini, seperti yang dijelaskan di bagian Prasyarat.
- AWS KMS, yang menyediakan enkripsi saat istirahat untuk konten yang disimpan dalam sistem file Amazon EFS.
- Fargate, yang mengelola sumber daya komputasi untuk kontainer sehingga Anda dapat fokus pada persyaratan bisnis alih-alih beban infrastruktur. Profil Fargate dibuat untuk semua subnet pribadi. Ini menyediakan redundansi per Availability Zone dalam virtual private cloud (VPC) cluster.
- Kubernetes Pods, untuk memvalidasi bahwa konten dapat dibagikan, dikonsumsi, dan ditulis oleh berbagai instance aplikasi.

Epik

Menyediakan kluster Amazon EKS (opsional)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat klaster Amazon EKS.	**Note** Jika Anda sudah memiliki cluster yang digunakan, lewati ke epik berikutnya. Buat klaster Amazon EKS di akun AWS Anda yang ada. Di GitHub direktori, gunakan salah satu pola untuk menyebarkan cluster Amazon EKS dengan menggunakan Terraform atau eksctl. Untuk informasi selengkapnya, lihat Membuat klaster Amazon EKS di dokumentasi Amazon EKS. Dalam pola Terraform, ada juga contoh yang menunjukkan cara: menautkan profil Fargate ke cluster Amazon EKS Anda, membuat sistem file Amazon EFS, dan menerapkan driver Amazon EFS CSI di	Administrator AWS, administrator Terraform atau eksctl, administrator Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	cluster Amazon EKS Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Variabel lingkungan ekspor.	Jalankan skrip env.sh. Ini memberikan informasi yang diperlukan pada langkah selanjutnya.
	`source ./scripts/env.sh`
	`Inform the AWS Account ID: <13-digit-account-id>`
	`Inform your AWS Region: <aws-Region-code>`
	`Inform your Amazon EKS Cluster Name: <amazon-eks-cluster-name>`
	`Inform the Amazon EFS Creation Token: <self-genereated-uuid>`
	Jika belum dicatat, Anda bisa mendapatkan semua informasi yang diminta di atas dengan perintah CLI berikut.
	`# ACCOUNT ID`
	`aws sts get-caller-identity --query "Account" --output text`
	`# REGION CODE`
	`aws configure get region`
	`# CLUSTER EKS NAME`
	`aws eks list-clusters --query "clusters" --output text`
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`# GENERATE EFS TOKEN` `uuidgen`	

Buat namespace Kubernetes dan profil Fargate yang ditautkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat namespace Kubernetes dan profil Fargate untuk beban kerja aplikasi.	Buat namespace untuk menerima beban kerja aplikasi yang berinteraksi dengan Amazon EFS. Jalankan `create-k8s-ns-and-linked-fargate-profile.sh` skrip. Anda dapat memilih untuk menggunakan nama namespace kustom atau namespace default yang disediakan. `poc-efs-eks-fargate`
	
	Dengan nama namespace aplikasi khusus:
	
	`export $APP_NAMESPACE=<CUSTOM_NAME>`
	`./scripts/epic01/create-k8s-ns-and-linked-fargate-profile.sh \`
	`-c "$CLUSTER_NAME" -n "$APP_NAMESPACE"`
	
	Tanpa nama namespace aplikasi khusus:
		Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`./scripts/epic01/create-k8s-ns-and-linked-fargate-profile.sh \ -c "$CLUSTER_NAME"` di `$CLUSTER_NAME` mana nama cluster Amazon EKS Anda. `-n <NAMESPACE>` Parameternya opsional; jika tidak diinformasikan, nama namespace yang dihasilkan default akan diberikan.	

Buat sistem file Amazon EFS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan token unik.	Amazon EFS memerlukan token pembuatan untuk memastikan operasi idempoten (memanggil operasi dengan token pembuatan yang sama tidak berpengaruh). Untuk memenuhi persyaratan ini, Anda harus menghasilkan token unik melalui teknik yang tersedia. Misalnya, Anda dapat membuat pengenal unik universal (UUID) untuk digunakan sebagai token pembuatan.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file Amazon EFS.	Buat sistem file untuk menerima file data yang dibaca dan ditulis oleh beban kerja aplikasi. Anda dapat membuat sistem file terenkripsi atau tidak terenkripsi. (Sebagai praktik terbaik, kode untuk pola ini menciptakan sistem terenkripsi untuk mengaktifkan enkripsi saat istirahat secara default.) Anda dapat menggunakan kunci AWS KMS simetris yang unik untuk mengenkripsi sistem file Anda. Jika kunci kustom tidak ditentukan, kunci yang dikelola AWS akan digunakan. Gunakan skrip create-efs.sh untuk membuat sistem file Amazon EFS terenkripsi atau tidak terenkripsi, setelah Anda membuat token unik untuk Amazon EFS. Dengan enkripsi saat istirahat, tanpa kunci KMS: `./scripts/epic02/create-efs.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN"`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	di `$CLUSTER_NAME` mana nama cluster Amazon EKS Anda dan `$EFS_CREATION_TOKEN` merupakan token pembuatan unik untuk sistem file.
	
	Dengan enkripsi saat istirahat, dengan kunci KMS:
	
	`./scripts/epic02/create-efs.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN" \ -k "$KMS_KEY_ALIAS"`
	
	di `$CLUSTER_NAME` mana nama kluster Amazon EKS Anda, `$EFS_CREATION_TOKEN` adalah token pembuatan unik untuk sistem file, dan `$KMS_KEY_ALIAS` merupakan alias untuk kunci KMS.
	
	Tanpa enkripsi:
	
	`./scripts/epic02/create-efs.sh -d \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di `$CLUSTER_NAME` mana nama cluster Amazon EKS Anda, `$EFS_CREATION_TOKEN` adalah token pembuatan unik untuk sistem file, dan `–d` menonaktifkan enkripsi saat istirahat.	
Buat grup keamanan.	Buat grup keamanan untuk memungkinkan kluster Amazon EKS mengakses sistem file Amazon EFS.	Administrator sistem AWS
Perbarui aturan masuk untuk grup keamanan.	Perbarui aturan masuk grup keamanan untuk mengizinkan lalu lintas masuk untuk pengaturan berikut: • Protokol TCP - port 2049 • Sumber — Rentang blok CIDR untuk subnet pribadi di VPC yang berisi klaster Kubernetes	Administrator sistem AWS
Tambahkan target pemasangan untuk setiap subnet pribadi.	Untuk setiap subnet pribadi klaster Kubernetes, buat target mount untuk sistem file dan grup keamanan.	Administrator sistem AWS

Instal komponen Amazon EFS ke dalam klaster Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan driver Amazon EFS CSI.	Terapkan driver Amazon EFS CSI ke dalam cluster.	Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pengemudi menyediakan penyimpanan sesuai dengan klaim volume persisten yang dibuat oleh aplikasi. Jalankan `create-k8s-efs-csi-sc.sh` skrip untuk menyebarkan driver Amazon EFS CSI dan kelas penyimpanan ke dalam cluster. `./scripts/epic03/create-k8s-efs-csi-sc.sh` Skrip ini menggunakan `kubectl` utilitas, jadi pastikan konteksnya telah dikonfigurasi dan arahkan ke cluster Amazon EKS yang diinginkan.	
Deploy kelas penyimpanan.	Menerapkan kelas penyimpanan ke dalam cluster untuk penyedia Amazon EFS (efs.csi.aws.com).	Pengguna Kubernetes dengan izin yang diberikan

Instal aplikasi PoC ke dalam cluster Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan volume persisten.	Terapkan volume persisten, dan tautkan ke kelas penyimpanan yang dibuat dan ke ID sistem file Amazon EFS. Aplikasi ini menggunakan volume persisten untuk	Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membaca dan menulis konten. Anda dapat menentuka n ukuran apa pun untuk volume persisten di bidang penyimpanan. Kubernete s membutuhkan bidang ini, tetapi karena Amazon EFS adalah sistem file elastis, ia tidak menerapkan kapasitas sistem file apa pun. Anda dapat menerapkan volume persisten dengan atau tanpa enkripsi. (Driver Amazon EFS CSI mengaktifkan enkripsi secara default, sebagai praktik terbaik.) Jalankan `deploy-poc-app.sh` skrip untuk menyebarkan volume persisten, klaim volume persisten, dan dua beban kerja. Dengan enkripsi dalam perjalanan: `./scripts/epic04/deploy-poc-app.sh \ -t "$EFS_CREATION_TOKEN"` di mana `$EFS_CREATION_TOKEN` adalah token penciptaan unik untuk sistem file.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tanpa enkripsi dalam perjalanan: `./scripts/epic04/deploy-poc-app.sh -d \ -t "$EFS_CREATION_TOKEN"` di `$EFS_CREATION_TOKEN` mana token pembuatan unik untuk sistem file, dan `-d` menonaktifkan enkripsi saat transit.	
Terapkan klaim volume persisten yang diminta oleh aplikasi.	Terapkan klaim volume persisten yang diminta oleh aplikasi, dan tautkan ke kelas penyimpanan. Gunakan mode akses yang sama dengan volume persisten yang Anda buat sebelumnya. Anda dapat menentukan ukuran apa pun untuk klaim volume persisten di bidang penyimpanan. Kubernetes membutuhkan bidang ini, tetapi karena Amazon EFS adalah sistem file elastis, ia tidak menerapkan kapasitas sistem file apa pun.	Pengguna Kubernetes dengan izin yang diberikan
Menyebarkan beban kerja 1.	Menerapkan pod yang mewakili beban kerja 1 dari aplikasi. Beban kerja ini menulis konten ke file `/data/out1.txt` .	Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan beban kerja 2.	Menerapkan pod yang mewakili beban kerja 2 dari aplikasi. Beban kerja ini menulis konten ke file`/data/out2.txt` .	Pengguna Kubernetes dengan izin yang diberikan

Validasi persistensi, daya tahan, dan kemampuan berbagi sistem file

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa status`PersistentVolume` .	Masukkan perintah berikut untuk memeriksa status file`PersistentVolume` . `kubectl get pv` Untuk keluaran contoh, lihat bagian Informasi tambahan.	Pengguna Kubernetes dengan izin yang diberikan
Periksa status`PersistentVolumeClaim` .	Masukkan perintah berikut untuk memeriksa status file`PersistentVolumeClaim` . `kubectl -n poc-efs-eks-fargate get pvc` Untuk keluaran contoh, lihat bagian Informasi tambahan.	Pengguna Kubernetes dengan izin yang diberikan
Validasi bahwa beban kerja 1 dapat menulis ke sistem file.	Masukkan perintah berikut untuk memvalidasi bahwa beban kerja 1 menulis ke. `/data/out1.txt`	Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`kubectl exec -ti poc-app1 -n poc-efs-eks-fargate -- tail -f /data/out1.txt`
	
	Hasilnya mirip dengan yang berikut:
	
	`...`
	`Thu Sep  3 15:25:07 UTC 2023 - PoC APP 1`
	`Thu Sep  3 15:25:12 UTC 2023 - PoC APP 1`
	`Thu Sep  3 15:25:17 UTC 2023 - PoC APP 1`
	`...`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi bahwa beban kerja 2 dapat menulis ke sistem file.	Masukkan perintah berikut untuk memvalidasi bahwa beban kerja 2 menulis ke. `/data/out2.txt`
	`kubectl -n $APP_NAMESPACE exec -ti poc-app2 -- tail -f /data/out2.txt`
	Hasilnya mirip dengan yang berikut:
	`... Thu Sep 3 15:26:48 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:53 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:58 UTC 2023 - PoC APP 2 ...`
		Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi bahwa beban kerja 1 dapat membaca file yang ditulis oleh beban kerja 2.	Masukkan perintah berikut untuk memvalidasi bahwa beban kerja 1 dapat membaca `/data/out2.txt` file yang ditulis oleh beban kerja 2.
	`kubectl exec -ti poc-app1 -n poc-efs-eks-fargate -- tail -n 3 /data/out2.txt`
	Hasilnya mirip dengan yang berikut:
	`... Thu Sep 3 15:26:48 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:53 UTC 2023 - PoC APP 2 Thu Sep 3 15:26:58 UTC 2023 - PoC APP 2 ...`
		Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi bahwa beban kerja 2 dapat membaca file yang ditulis oleh beban kerja 1.	Masukkan perintah berikut untuk memvalidasi bahwa beban kerja 2 dapat membaca `/data/out1.txt` file yang ditulis oleh beban kerja 1.
	`kubectl -n $APP_NAMESPACE exec -ti poc-app2 -- tail -n 3 /data/out1.txt`
	Hasilnya mirip dengan yang berikut:
	`... Thu Sep 3 15:29:22 UTC 2023 - PoC APP 1 Thu Sep 3 15:29:27 UTC 2023 - PoC APP 1 Thu Sep 3 15:29:32 UTC 2023 - PoC APP 1 ...`
		Pengguna Kubernetes dengan izin yang diberikan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi bahwa file disimpan setelah Anda menghapus komponen aplikasi.	Selanjutnya, Anda menggunakan skrip untuk menghapus komponen aplikasi (volume persisten, klaim volume persisten, dan pod), dan memvalidasi bahwa file `/data/out1.txt` dan `/data/out2.txt` disimpan dalam sistem file. Jalankan skrip `validate-efs-content.sh` menggunakan perintah berikut.
	`./scripts/epic05/validate-efs-content.sh \ -t "$EFS_CREATION_TOKEN"`
	di mana `$EFS_CREATION_TOKEN` adalah token penciptaan unik untuk sistem file.
	Hasilnya mirip dengan yang berikut:
	`pod/poc-app-validation created Waiting for pod get Running state... Waiting for pod get Running state... Waiting for pod get Running state... Results from execution of 'find /data' on`
		Pengguna Kubernetes dengan izin yang diberikan, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	validation process pod: /data /data/out2.txt /data/out1.txt	

Memantau operasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pantau log aplikasi.	Sebagai bagian dari operasi dua hari, kirimkan log aplikasi ke Amazon CloudWatch untuk pemantauan.	Administrator sistem AWS, pengguna Kubernetes dengan izin yang diberikan
Pantau container Amazon EKS dan Kubernetes dengan Container Insights.	Sebagai bagian dari operasi dua hari, pantau sistem Amazon EKS dan Kubernetes dengan menggunakan Amazon Container Insights. CloudWatch Alat ini mengumpulkan, mengumpulkan, dan merangkum metrik dari aplikasi kontainer pada tingkat dan dimensi yang berbeda. Untuk informasi selengkapnya, lihat bagian Sumber daya terkait.	Administrator sistem AWS, pengguna Kubernetes dengan izin yang diberikan
Pantau Amazon EFS dengan CloudWatch.	Sebagai bagian dari operasi dua hari, pantau sistem file menggunakan Amazon CloudWatch, yang mengumpulkan dan memproses data mentah dari Amazon EFS	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menjadi metrik yang dapat dibaca, mendekati waktu nyata. Untuk informasi selengkapnya, lihat bagian Sumber daya terkait.	

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan semua sumber daya yang dibuat untuk pola tersebut.	Setelah Anda menyelesaikan pola ini, bersihkan semua sumber daya, untuk menghindari biaya AWS. Jalankan `clean-up-resources.sh` skrip untuk menghapus semua sumber daya setelah Anda selesai menggunakan aplikasi PoC. Lengkapi salah satu opsi berikut. Dengan enkripsi saat istirahat, dengan kunci KMS: `./scripts/epic06/clean-up-resources.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN" \ -k "$KMS_KEY_ALIAS"` di `$CLUSTER_NAME` mana nama kluster Amazon	Pengguna Kubernetes dengan izin yang diberikan, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	EKS Anda, `$EFS_CREATION_TOKEN` adalah token pembuatan untuk sistem file, dan `$KMS_KEY_ALIAS` merupakan alias untuk kunci KMS. Tanpa enkripsi saat istirahat: `./scripts/epic06/clean-up-resources.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN"` di `$CLUSTER_NAME` mana nama cluster Amazon EKS Anda dan `$EFS_CREATION_TOKEN` merupakan token pembuatan untuk sistem file.	

Sumber daya terkait

Referensi

- AWS Fargate untuk Amazon EKS sekarang mendukung Amazon EFS (pengumuman)
- Cara menangkap log aplikasi saat menggunakan Amazon EKS di AWS Fargate (posting blog)
- Menggunakan Wawasan Kontainer (CloudWatch dokumentasi Amazon)
- Menyiapkan Wawasan Kontainer di Amazon EKS dan Kubernetes (dokumentasi Amazon) CloudWatch
- Metrik Amazon EKS dan Kubernetes Container Insights (dokumentasi Amazon) CloudWatch
- Memantau Amazon EFS dengan Amazon CloudWatch (dokumentasi Amazon EFS)

GitHub tutorial dan contoh

- Penyediaan statis
- Enkripsi dalam perjalanan
- Mengakses sistem file dari beberapa pod
- Mengkonsumsi Amazon EFS di StatefulSets
- Pemasangan subpath
- Menggunakan titik akses Amazon EFS
- Cetak Biru Amazon EKS untuk Terraform

Alat yang dibutuhkan

- Menginstal AWS CLI versi 2
- Instalasi eksctl
- Instalasi kubectl
- Instalasi jq

Informasi tambahan

Berikut ini adalah contoh output dari `kubectl get pv` perintah.

```
NAME         CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
            STORAGECLASS   REASON   AGE
poc-app-pv   1Mi        RWX            Retain           Bound    poc-efs-eks-fargate/
poc-app-pvc   efs-sc                  3m56s
```

Berikut ini adalah contoh output dari `kubectl -n poc-efs-eks-fargate get pvc` perintah.

```
NAME          STATUS   VOLUME       CAPACITY   ACCESS MODES   STORAGECLASS   AGE
poc-app-pvc   Bound    poc-app-pv   1Mi        RWX            efs-sc         4m34s
```

Siapkan penskalaan otomatis berbasis peristiwa di Amazon EKS dengan menggunakan Amazon EKS Pod Identity dan KEDA

Dipen Desai, Abhay Diwan, Kamal Joshi, dan Mahendra Revanasiddappa, Amazon Web Services

Ringkasan

Platform orkestrasi, seperti Amazon Elastic Kubernetes Service (Amazon EKS), telah menyederhanakan pengelolaan siklus hidup aplikasi berbasis container. Ini membantu organisasi fokus pada membangun, mengamankan, mengoperasikan, dan memelihara aplikasi berbasis kontainer. Karena penerapan berbasis peristiwa menjadi lebih umum, organisasi lebih sering menskalakan penerapan Kubernetes berdasarkan berbagai sumber peristiwa. Metode ini, dikombinasikan dengan auto scaling, dapat menghasilkan penghematan biaya yang signifikan dengan menyediakan sumber daya komputasi sesuai permintaan dan penskalaan efisien yang disesuaikan dengan logika aplikasi.

KEDA adalah autoscaler berbasis peristiwa berbasis Kubernetes. KEDA membantu Anda menskalakan kontainer apa pun di Kubernetes berdasarkan jumlah peristiwa yang perlu diproses. Ini ringan dan terintegrasi dengan cluster Kubernetes apa pun. Ia juga bekerja dengan komponen Kubernetes standar, seperti Horizontal Pod Autoscaling (HPA). KEDA juga menawarkan TriggerAuthentication, yang merupakan fitur yang membantu Anda mendelegasikan otentikasi. Hal ini memungkinkan Anda untuk menjelaskan parameter otentikasi yang terpisah dari ScaledObject dan wadah penyebaran.

AWS menyediakan peran AWS Identity and Access Management (IAM) yang mendukung beragam opsi penerapan Kubernetes, termasuk Amazon EKS, Amazon EKS Anywhere, Layanan OpenShift Red Hat di AWS (ROSA), dan cluster Kubernetes yang dikelola sendiri di Amazon Elastic Compute Cloud (Amazon). EC2 Peran ini menggunakan konstruksi IAM, seperti penyedia identitas OpenID Connect (OIDC) dan kebijakan kepercayaan IAM, untuk beroperasi di berbagai lingkungan tanpa bergantung langsung pada layanan Amazon EKS atau. APIs Untuk informasi selengkapnya, lihat peran IAM untuk akun layanan di dokumentasi Amazon EKS.

Amazon EKS Pod Identity menyederhanakan proses akun layanan Kubernetes untuk mengambil peran IAM tanpa memerlukan penyedia OIDC. Ini memberikan kemampuan untuk mengelola kredensyal untuk aplikasi Anda. Alih-alih membuat dan mendistribusikan AWS kredensialmu ke container atau menggunakan peran EC2 instans Amazon, kamu mengaitkan peran IAM dengan akun layanan Kubernetes dan mengonfigurasi Pod kamu untuk menggunakan akun layanan. Ini membantu

Anda menggunakan peran IAM di beberapa klaster dan menyederhanakan manajemen kebijakan dengan mengaktifkan penggunaan kembali kebijakan izin di seluruh peran IAM.

Dengan menerapkan KEDA dengan Amazon EKS Pod Identity, bisnis dapat mencapai penskalaan otomatis berbasis peristiwa yang efisien dan pengelolaan kredensi yang disederhanakan. Skala aplikasi berdasarkan permintaan, yang mengoptimalkan pemanfaatan sumber daya dan mengurangi biaya.

Pola ini membantu Anda mengintegrasikan Amazon EKS Pod Identity dengan KEDA. Ini menampilkan bagaimana Anda dapat menggunakan akun `keda-operator` layanan dan mendelegasikan otentikasi dengan. `TriggerAuthentication` Ini juga menjelaskan cara mengatur hubungan kepercayaan antara peran IAM untuk operator KEDA dan peran IAM untuk aplikasi. Hubungan kepercayaan ini memungkinkan KEDA untuk memantau pesan dalam antrian acara dan menyesuaikan penskalaan untuk objek Kubernetes tujuan.

Prasyarat dan batasan

Prasyarat

- AWS Command Line Interface (AWS CLI) versi 2.13.17 atau yang lebih baru, diinstal
- Python versi 3.11.5 atau yang lebih baru, diinstal
- AWS SDK for Python (Boto3) versi 1.34.135 atau yang lebih baru, diinstal
- Helm versi 3.12.3 atau yang lebih baru, diinstal
- kubectl versi 1.25.1 atau yang lebih baru, diinstal
- Docker Engine versi 26.1.1 atau yang lebih baru, diinstal
- Cluster Amazon EKS versi 1.24 atau yang lebih baru, dibuat
- Prasyarat untuk membuat agen Amazon EKS Pod Identity, dipenuhi

Batasan

- Diperlukan bahwa Anda membangun hubungan kepercayaan antara `keda-operator` peran dan `keda-identity` peran. Instruksi disediakan di bagian Epik dari pola ini.

Arsitektur

Dalam pola ini, Anda membuat AWS sumber daya berikut:

- Repositori Amazon Elastic Container Registry (Amazon ECR) - Dalam pola ini, repo ini diberi nama. `keda-pod-identity-registry` Repo pribadi ini digunakan untuk menyimpan gambar Docker dari aplikasi sampel.
- Antrian Amazon Simple Queue Service (Amazon SQS) - Dalam pola ini, antrian ini diberi nama. `event-messages-queue` Antrian bertindak sebagai buffer pesan yang mengumpulkan dan menyimpan pesan masuk. KEDA memantau metrik antrian, seperti jumlah pesan atau panjang antrian, dan secara otomatis menskalakan aplikasi berdasarkan metrik ini.
- Peran IAM untuk aplikasi — Dalam pola ini, peran ini diberi nama`keda-identity`. `keda-operator`Peran mengasumsikan peran ini. Peran ini memungkinkan akses ke antrian Amazon SQS.
- Peran IAM untuk operator KEDA - Dalam pola ini, peran ini diberi nama. `keda-operator` Operator KEDA menggunakan peran ini untuk melakukan panggilan AWS API yang diperlukan. Peran ini memiliki izin untuk mengambil `keda-identity` peran. Karena hubungan kepercayaan antara peran `keda-operator` dan `keda-identity` peran, `keda-operator` peran tersebut memiliki izin Amazon SQS.

Melalui sumber daya kustom `TriggerAuthentication` dan `ScaledObject` Kubernetes, operator menggunakan `keda-identity` peran tersebut untuk terhubung dengan antrean Amazon SQS. Berdasarkan ukuran antrian, KEDA secara otomatis menskalakan penerapan aplikasi. Ini menambahkan 1 pod untuk setiap 5 pesan yang belum dibaca dalam antrian. Dalam konfigurasi default, jika tidak ada pesan yang belum dibaca dalam antrean Amazon SQS, aplikasi akan menurunkan skala ke 0 pod. Operator KEDA memantau antrian pada interval yang Anda tentukan.

Gambar berikut menunjukkan cara Anda menggunakan Amazon EKS Pod Identity untuk menyediakan `keda-operator` peran dengan akses aman ke antrean Amazon SQS.

Diagram menunjukkan alur kerja berikut:

1. Anda menginstal agen Amazon EKS Pod Identity di cluster Amazon EKS.
2. Anda menerapkan operator KEDA di namespace KEDA di cluster Amazon EKS.
3. Anda membuat peran `keda-operator` dan `keda-identity` IAM di target Akun AWS.
4. Anda membangun hubungan kepercayaan antara peran IAM.
5. Anda menyebarkan aplikasi di `security` namespace.

6. Operator KEDA melakukan polling pesan dalam antrian Amazon SQS.
7. KEDA memulai HPA, yang secara otomatis menskalakan aplikasi berdasarkan ukuran antrian.

Alat

Layanan AWS

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.

Alat-alat lainnya

- KEDA adalah autoscaler berbasis peristiwa berbasis Kubernetes.

Repositori kode

Kode untuk pola ini tersedia di GitHub Event-driven auto scaling menggunakan EKS Pod Identity dan REPOSITORI KEDA.

Praktik terbaik

Sebaiknya Anda untuk mematuhi praktik terbaik berikut:

- Praktik terbaik Amazon EKS
- Praktik terbaik keamanan di IAM
- Praktik terbaik Amazon SQS

Epik

Buat AWS sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk operator KEDA.	1. Masuk ke Konsol Manajemen AWS, lalu buka konsol IAM. 2. Di panel navigasi, pilih Peran. 3. Pilih Buat peran. 4. Pilih jenis peran kebijakan kepercayaan kustom. 5. Di bagian Kebijakan kepercayaan khusus, masukkan kebijakan kepercayaan khusus berikut untuk peran ini: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "pods.eks.amazonaws.com" }, "Action": [ "sts:AssumeRole", "sts:TagSession"`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>]
	        }
	    ]
	}</pre>
	6. Pada halaman Tambahkan izin, pilih Berikutnya. Anda tidak menambahkan kebijakan apa pun ke peran ini.
	7. Untuk Nama peran, masukkan `keda-operator` .
	8. Pilih Buat peran.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk aplikasi sampel.	1. Di konsol IAM, di panel navigasi, pilih Peran. 2. Pilih Buat peran. 3. Pilih jenis peran kebijakan kepercayaan kustom. 4. Di bagian Kebijakan kepercayaan khusus, masukkan kebijakan kepercayaan khusus berikut untuk peran ini. Ganti `<account number>` dengan nomor akun target Anda: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "pods.eks.amazonaws.com", "AWS": "arn:aws:iam::<account number>:role/keda-operator" }, "Action": [ "sts:AssumeRole", "sts:TagSession" ]`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre> }] }</pre> 5. Pada halaman Tambahkan izin, tambahkan kebijakan AWS terkelola berikut ke peran: • `AmazonSQSReadOnlyAccess` • `AWSLambdaSQSQueueExecutionRole` 6. Pilih Berikutnya. 7. Untuk Nama peran, masukkan `keda-identity` . 8. Pilih Buat peran.	
Membuat antrean Amazon SQS.	1. Buka konsol Amazon SQS. 2. Pilih Buat antrean. 3. Untuk Jenis, pilih Standar. 4. Pada Buat halaman antrian, untuk Nama, masukkan`event-messages-queue` . 5. Pilih Buat antrean. Anda tidak mengubah pengaturan default untuk antrian ini.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR.	1. Buka konsol Amazon ECR. 2. Pilih Buat repositori. 3. Untuk nama Repositori, masukkan. `keda-pod-identity-registry` 4. Pilih Buat repositori. Anda tidak mengubah pengaturan default untuk repositori ini.	AWS Umum

Siapkan kluster Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan agen Amazon EKS Pod Identity.	Untuk cluster Amazon EKS target, siapkan agen Amazon EKS Pod Identity. Ikuti petunjuk dalam Mengatur Agen Identitas Pod Amazon EKS di dokumentasi Amazon EKS.	AWS DevOps

Menyebarkan KEDA.	1. Masukkan perintah berikut untuk menerapkan KEDA pada cluster Amazon EKS target:
	`# Add Helm Repo for Keda`
	`helm repo add kedacore https://kedacore.github.io/charts`
	`# Update Helm repo`
	`helm repo update`
	`# Install Keda`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`helm install keda kedacore/keda --namespace keda --create-namespace`
	
	Untuk informasi selengkapnya, lihat Menerapkan dengan Helm di dokumentasi KEDA.
	
	2. Setelah penerapan berhasil, dalam output, validasi bahwa tiga penerapan dibuat untuk operator KEDA. Berikut ini adalah contoh output:
	
	`NAME READY UP-TO-DATE AVAILABLE AGE`
	`keda-admission-webhooks 1/1 1 1 89s`
	`keda-operator 1/1 1 1 89s`
	`keda-operator-metrics-apiserver 1/1 1 1 89s`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan peran IAM ke akun layanan Kubernetes.	Ikuti petunjuk dalam Menetapkan peran IAM ke akun layanan Kubernetes di dokumentasi Amazon EKS. Gunakan nilai berikut: • Untuk peran IAM, masukkan`keda-operator` . • Untuk namespace Kubernetes, masukkan. `keda` • Untuk akun layanan Kubernetes, masukkan. `keda-operator`	AWS DevOps
Buat namespace.	Masukkan perintah berikut untuk membuat `security` namespace di cluster Amazon EKS target: `kubectl create ns security`	DevOps insinyur

Men-deploy aplikasi sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning file aplikasi.	Masukkan perintah berikut untuk mengkloning penskalaan auto Event-driven menggunakan EKS Pod Identity dan repositori KEDA dari: GitHub	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/aws-samples/event-driven-autoscaling-using-podidentity-and-keda.git`	

membuat gambar Docker.	1. Masukkan perintah berikut untuk menavigasi ke repositori kloning:
	`cd event-driven-autoscaling-using-podidentity-and-keda`
	2. Masukkan perintah berikut untuk membangun image Docker untuk aplikasi sampel:
	`docker build -t keda-pod-identity-registry .`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Dorong gambar Docker ke Amazon ECR.	1. Di terminal tempat Anda membuat gambar Docker, masukkan perintah berikut untuk masuk ke Amazon ECR. Ganti <AWS_REGION> dan <AWS_ACCOUNT_ID> dengan nilai dari AWS lingkungan Anda:
	`aws ecr get-login-password \ --region <AWS_REGION> | docker login \ --username AWS \ --password-stdin <AWS_ACCOUNT_ID>.dkr.ecr.<AWS_REGION>.amazonaws.com`
	2. Masukkan perintah berikut untuk menandai gambar. Ganti <AWS_REGION> dan <AWS_ACCOUNT_ID> dengan nilai dari AWS lingkungan Anda:
	`docker tag keda-pod-identity-registry:latest <AWS_ACCOUNT_ID>.dkr.ecr.<AWS_REGION>.amazonaws.com/keda-pod-identity-registry:latest`
	3. Masukkan perintah berikut untuk mendorong gambar
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ke Amazon ECR. Ganti `<AWS_REGION>` dan `<AWS_ACCOUNT_ID>` dengan nilai dari AWS lingkungan Anda: `docker push <AWS_ACCOUNT_ID>.dkr.ecr.<AWS_REGION>.amazonaws.com/keda-pod-identity-registry:latest` **Note** Anda dapat menemukan perintah push dengan menavigasi ke halaman repositori Amazon ECR dan kemudian memilih perintah View push.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Deploy aplikasi sampel.	1. Di repositori kloning, buka file deploy.yaml. 2. Ganti <AWS_ACCOUNT_ID> dan <AWS_REGION> dengan nilai-nilai dari lingkungan Anda. 3. Simpan dan tutup file deploy.yaml. 4. Masukkan perintah berikut untuk menerapkan aplikasi sampel pada kluster Amazon EKS target: `kubectl apply -f deploy.yaml` Perintah ini membuat akun penyebaran dan layanan di cluster.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tetapkan peran IAM ke akun layanan aplikasi.	Lakukan salah satu hal berikut untuk mengaitkan peran `keda-identity` IAM dengan akun layanan untuk aplikasi sampel:
	• Ikuti petunjuk dalam Menetapkan peran IAM ke akun layanan Kubernetes di dokumentasi Amazon EKS. Gunakan nilai berikut:
	  • Untuk peran IAM, masukkan `keda-identity` .
	  • Untuk namespace Kubernetes, masukkan. `security`
	  • Untuk akun layanan Kubernetes, masukkan. `my-sqs-read-msgs`
	• Masukkan AWS CLI perintah berikut. Ganti `<cluster-name>` dengan nama kluster Amazon EKS target, dan ganti `<role-ARN>` dengan Nama Sumber Daya Amazon (ARN) peran: `keda-identity`
	`aws eks create-pod-identity-association \`
	`  --cluster-name <cluster-name> \`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--role-arn <role-ARN> \ --namespace security \ --service-account my-sqs-read-msgs`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan `ScaledObject` dan `TriggerAuthentication`.	1. Di repositori kloning, buka file keda.yaml. 2. Ganti `{{AWS_ACCOUNT_ID}}` dengan ID target Anda Akun AWS. 3. Ganti `{{AWS_REGION}}` dengan target Anda Wilayah AWS. 4. (Opsional) Pada baris 21—24, perbarui parameter untuk kebijakan `ScaledObject` penskalaan. Lihat berikut ini untuk informasi selengkapnya tentang parameter ini: • Interval Pemungutan Suara • periode CoolDownperiod • idleReplicaCount • minReplicaCount 5. Simpan dan tutup file keda.yaml. 6. Masukkan perintah berikut untuk menyebarkan `ScaledObject` dan `TriggerAuthentication` sumber daya: `kubectl -n security apply -f keda.yaml`	DevOps insinyur

Uji penskalaan otomatis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kirim pesan ke antrian Amazon SQS.	1. Masukkan perintah berikut untuk menavigasi ke repositori kloning:
	`cd event-driven-autoscaling-using-podidentity-and-keda`
	2. Masukkan perintah berikut untuk mengirim pesan pengujian ke antrean Amazon SQS:
	`python sqs_send_msg.py`
	Skrip sqs_send_msg.py bertindak sebagai aplikasi yang menghasilkan pesan untuk menguji penskalaan otomatis.
	Note
	Jika Anda menjalankan Python 3, masukkan. `python3 sqs_send_msg.py`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Pantau pod aplikasi.	1. Di terminal yang berbeda, masukkan perintah berikut untuk memantau pod:
	
	`kubectl -n security get po`
	
	2. Untuk setiap 5 pesan yang belum dibaca dalam antrean Amazon SQS, KEDA menambahkan satu pod. Pada output dari perintah sebelumnya, konfirmasikan bahwa pod baru sedang ditambahkan. Berikut ini adalah contoh output:
	
	`kubectl -n security get po`
	`NAME READY STATUS RESTARTS AGE`
	`q-read-797f4c7589-2bj76 1/1 Running 0 2s`
	`q-read-797f4c7589-4zxph 1/1 Running 0 49s`
	`q-read-797f4c7589-cg9dt 1/1 Running 0 18s`
	`q-read-797f4c7589-slc69 1/1`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Running 0 33s` 3. Ketika Anda selesai menguji, di terminal asli, masukkan CTRL+C (Windows) atau CMD + C (macOS). Ini menghentikan skrip python sqs_send_msg.py.	

Pemecahan masalah

Isu	Solusi

Operator KEDA tidak dapat menskalakan aplikasi.	Masukkan perintah berikut untuk memeriksa log peran `keda-operator` IAM:
	`kubectl logs -n keda -l app=keda-operator -c keda-operator`
	Jika ada kode `HTTP 403` respons, maka aplikasi dan scaler KEDA tidak memiliki izin yang cukup untuk mengakses antrian Amazon SQS. Selesaikan langkah-langkah berikut:
	1. Periksa kebijakan dan pernyataan IAM untuk `keda-identity` peran untuk mengonfirmasi bahwa akses baca antrian diberikan.
	2. Validasi hubungan kepercayaan antara peran IAM. Berikut ini adalah contohnya:
	`{ "Version": "2012-10-17",`

Isu	Solusi

	<pre> "Statement": [
	        {
	            "Effect": "Allow",
	            "Principal": {
	                "Service": "pods.eks
	.amazonaws.com"
	            },
	            "Action": [
	                "sts:AssumeRole",
	                "sts:TagSession"
	            ]
	        }
	    ]
	}</pre>
	Jika ada `Assume-Role` kesalahan, maka peran IAM node Amazon EKS tidak dapat mengambil peran IAM yang ditentukan. `TriggerAuthentication` Selesaikan langkah-langkah berikut:
	1. Masukkan perintah berikut untuk menghapus `keda-operator` pod dan membuat yang baru:
	<pre>kubectl delete pod keda-operator-
	<alphenumeric-value> --namespace
	 keda</pre>
	2. Masukkan perintah berikut untuk memeriksa identitas yang diasumsikan oleh pod:
	<pre>kubectl describe pod <keda-operator-
	pod-name> --namespace keda</pre>
	3. Ketika pod berhasil di-restart, konfirmasikan bahwa dua variabel berikut ditambahkan ke deskripsi pod:

Isu	Solusi
	• `AWS_CONTAINER_CREDENTIALS_FULL_URI` • `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`

Sumber daya terkait

- Siapkan Agen Identitas Pod Amazon EKS (dokumentasi Amazon EKS)
- Menyebarkan KEDA (dokumentasi KEDA)
- ScaledObject spesifikasi (dokumentasi KEDA)
- Otentikasi dengan TriggerAuthentication (dokumentasi KEDA)

Merampingkan penerapan PostgreSQL di Amazon EKS dengan menggunakan PGO

Shalaka Dengale, Amazon Web Services

Ringkasan

Pola ini mengintegrasikan Operator Postgres dari Crunchy Data (PGO) dengan Amazon Elastic Kubernetes Service (Amazon EKS) untuk merampingkan penerapan PostgreSQL di lingkungan cloud-native. PGO menyediakan otomatisasi dan skalabilitas untuk mengelola database PostgreSQL di Kubernetes. Saat Anda menggabungkan PGO dengan Amazon EKS, PGO akan membentuk platform yang kuat untuk menerapkan, mengelola, dan menskalakan database PostgreSQL secara efisien.

Integrasi ini memberikan manfaat utama berikut:

- Penyebaran otomatis: Menyederhanakan penyebaran dan manajemen cluster PostgreSQL.
- Definisi sumber daya kustom (CRDs): Menggunakan primitif Kubernetes untuk manajemen PostgreSQL.
- Ketersediaan tinggi: Mendukung failover otomatis dan replikasi sinkron.
- Pencadangan dan pemulihan otomatis: Merampingkan proses pencadangan dan pemulihan.
- Penskalaan horisontal: Memungkinkan penskalaan dinamis cluster PostgreSQL.
- Peningkatan versi: Memfasilitasi peningkatan bergulir dengan waktu henti minimal.
- Keamanan: Menerapkan enkripsi, kontrol akses, dan mekanisme otentikasi.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- AWS CLI Config, untuk menghubungkan AWS sumber daya dari baris perintah.
- eksctl, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- `kubectl`, diinstal dan dikonfigurasi untuk mengakses sumber daya di kluster Amazon EKS Anda. Untuk informasi selengkapnya, lihat Mengatur kubectl dan eksctl di dokumentasi Amazon EKS.

- Terminal komputer Anda dikonfigurasi untuk mengakses kluster Amazon EKS. Untuk informasi selengkapnya, lihat Mengonfigurasi komputer untuk berkomunikasi dengan klaster Anda di dokumentasi Amazon EKS.

Versi produk

- Kubernetes versi 1.21—1.24 atau yang lebih baru (lihat dokumentasi PGO).
- PostgreSQL versi 10 atau yang lebih baru. Pola ini menggunakan PostgreSQL versi 16.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Tumpukan teknologi target

- Amazon EKS
- Amazon Virtual Private Cloud (Amazon VPC)
- Amazon Elastic Compute Cloud (Amazon EC2)

Arsitektur target

Pola ini membangun arsitektur yang berisi cluster Amazon EKS dengan tiga node. Setiap node berjalan pada satu set EC2 instance di backend. Pengaturan PostgreSQL ini mengikuti arsitektur replika utama, yang sangat efektif untuk kasus penggunaan read-heavy. Arsitektur mencakup komponen-komponen berikut:

- Container database primer (pg-primary) menghosting instance PostgreSQL utama di mana semua operasi penulisan diarahkan.
- Container replika sekunder (pg-replica) menghosting instance PostgreSQL yang mereplikasi data dari database utama dan menangani operasi baca.

- PgBounceradalah pooler koneksi ringan untuk database PostgreSQL yang disertakan dengan PGO. Itu berada di antara klien dan server PostgreSQL, dan bertindak sebagai perantara untuk koneksi database.
- PGO mengotomatiskan penerapan dan pengelolaan klaster PostgreSQL di lingkungan Kubernetes ini.
- Patroni adalah alat open-source yang mengelola dan mengotomatiskan konfigurasi ketersediaan tinggi untuk PostgreSQL. Ini termasuk dengan PGO. Saat Anda menggunakan Patroni dengan PGO di Kubernetes, Patroni memainkan peran penting dalam memastikan ketahanan dan toleransi kesalahan klaster PostgreSQL. Untuk informasi lebih lanjut, lihat dokumentasi Patroni.

Alur kerja mencakup langkah-langkah ini:

- Menyebarkan operator PGO. Anda menerapkan operator PGO di klaster Kubernetes Anda yang berjalan di Amazon EKS. Ini dapat dilakukan dengan menggunakan manifes Kubernetes atau bagan Helm. Pola ini menggunakan manifes Kubernetes.
- Tentukan instance PostgreSQL. Saat operator berjalan, Anda membuat resource kustom (CRs) untuk menentukan status instans PostgreSQL yang diinginkan. Ini termasuk konfigurasi seperti penyimpanan, replikasi, dan pengaturan ketersediaan tinggi.
- Manajemen operator. Anda berinteraksi dengan operator melalui objek API Kubernetes seperti CRs membuat, memperbarui, atau menghapus instance PostgreSQL.
- Pemantauan dan pemeliharaan. Anda dapat memantau kesehatan dan kinerja instans PostgreSQL yang berjalan di Amazon EKS. Operator sering menyediakan metrik dan pencatatan untuk tujuan pemantauan. Anda dapat melakukan tugas pemeliharaan rutin seperti peningkatan dan penambalan seperlunya. Untuk informasi selengkapnya, lihat Memantau performa klaster Anda dan melihat log di dokumentasi Amazon EKS.
- Penskalaan dan pencadangan: Anda dapat menggunakan fitur yang disediakan oleh operator untuk menskalakan instance PostgreSQL dan mengelola cadangan.

Pola ini tidak mencakup operasi pemantauan, pemeliharaan, dan pencadangan.

Otomatisasi dan skala

- Anda dapat menggunakan CloudFormation untuk mengotomatiskan pembuatan infrastruktur. Untuk informasi selengkapnya, lihat Membuat resource Amazon EKS dengan CloudFormation dokumentasi Amazon EKS.

- Anda dapat menggunakan GitVersion atau Jenkins membangun nomor untuk mengotomatiskan penyebaran instance database.

Alat

Layanan AWS

- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

Alat-alat lainnya

- eksctl adalah alat baris perintah sederhana untuk membuat cluster di Amazon EKS.
- kubectl adalah utilitas baris perintah untuk menjalankan perintah terhadap klaster Kubernetes.
- PGO mengotomatiskan dan menskalakan pengelolaan database PostgreSQL di Kubernetes.

Praktik terbaik

Ikuti praktik terbaik ini untuk memastikan penerapan yang lancar dan efisien:

- Amankan kluster EKS Anda. Terapkan praktik terbaik keamanan untuk kluster EKS Anda, seperti menggunakan peran AWS Identity and Access Management (IAM) untuk akun layanan (IRSA), kebijakan jaringan, dan grup keamanan VPC. Batasi akses ke server API cluster EKS, dan enkripsi komunikasi antara node dan server API dengan menggunakan TLS.
- Pastikan kompatibilitas versi antara PGO dan Kubernetes berjalan di Amazon EKS. Beberapa fitur PGO mungkin memerlukan versi Kubernetes tertentu atau memperkenalkan batasan kompatibilitas. Untuk informasi selengkapnya, lihat Komponen dan Kompatibilitas dalam dokumentasi PGO.
- Rencanakan alokasi sumber daya untuk penyebaran PGO Anda, termasuk CPU, memori, dan penyimpanan. Pertimbangkan persyaratan sumber daya dari PGO dan instance PostgreSQL yang dikelolanya. Pantau penggunaan sumber daya dan skala sumber daya sesuai kebutuhan.

- Desain untuk ketersediaan tinggi. Rancang penerapan PGO Anda untuk ketersediaan tinggi guna meminimalkan waktu henti dan memastikan keandalan. Terapkan beberapa replika PGO di beberapa Availability Zone untuk toleransi kesalahan.
- Menerapkan prosedur pencadangan dan pemulihan untuk database PostgreSQL Anda yang dikelola PGO. Gunakan fitur yang disediakan oleh PGO atau solusi cadangan pihak ketiga yang kompatibel dengan Kubernetes dan Amazon EKS.
- Siapkan pemantauan dan pencatatan untuk penerapan PGO Anda untuk melacak kinerja, kesehatan, dan peristiwa. Gunakan alat seperti Prometheus untuk memantau metrik dan Grafana untuk visualisasi. Konfigurasikan logging untuk menangkap log PGO untuk pemecahan masalah dan audit.
- Konfigurasikan jaringan dengan benar untuk memungkinkan komunikasi antara instans PGO, PostgreSQL, dan layanan lainnya di klaster Kubernetes Anda. Gunakan fitur jaringan VPC Amazon dan plugin jaringan Kubernetes seperti Calico atau Amazon VPC CNI untuk penegakan kebijakan jaringan dan isolasi lalu lintas.
- Pilih opsi penyimpanan yang sesuai untuk database PostgreSQL Anda, dengan mempertimbangkan faktor-faktor seperti kinerja, daya tahan, dan skalabilitas. Gunakan volume Amazon Elastic Block Store (Amazon EBS) AWS atau layanan penyimpanan terkelola untuk penyimpanan persisten. Untuk informasi selengkapnya, lihat Menyimpan volume Kubernetes dengan Amazon EBS di dokumentasi Amazon EKS.
- Gunakan alat infrastruktur sebagai kode (IAc) seperti CloudFormation untuk mengotomatiskan penerapan dan konfigurasi PGO di Amazon EKS. Tentukan komponen infrastruktur—termasuk kluster EKS, jaringan, dan sumber daya PGO—sebagai kode untuk konsistensi, pengulangan, dan kontrol versi.

Epik

Membuat peran IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM.	1. Buat peran IAM dengan menggunakan perintah berikut di AWS CLI: `aws iam create-role \`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--role-name {YourRoleName} \ --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "eks.amazonaws.com" }, "Action": "sts:AssumeRole" }] }' && \ aws iam attach-role-policy \ --role-name {YourRoleName}\ --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy && \ aws iam attach-role-policy \ --role-name {YourRoleName}\ --policy-arn arn:aws:iam::aws:policy/AmazonEKSServicePolicy && \ aws iam attach-role-policy \`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--role-name {YourRoleName}\ --policy-arn arn:aws:iam::aws:policy/CloudWatchFullAccess`
	2. Tinjau peran dalam Konsol Manajemen AWS:
	a. Buka konsol IAM.
	b. Pilih Peran, dan cari nama peran yang Anda buat.
	c. Validasi bahwa kebijakan berikut terlampir:
	`AmazonEKSClusterPolicy`
	`AmazonEKSServicePolicy`
	`CloudWatchFullAccess`	

Buat kluster Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat klaster Amazon EKS.	Jika Anda sudah menerapkan cluster, lewati langkah ini. Jika tidak, terapkan kluster Amazon EKS di saat ini Akun AWS dengan menggunak an`eksctl`, Terraform, atau. CloudFormation Pola ini	Administrator AWS, administrator Terraform atau eksctl, administrator Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	digunakan `eksctl` untuk penyebaran cluster.
	
	Note
	Pola ini menggunakan Amazon EC2 sebagai grup simpul untuk Amazon EKS. Jika Anda ingin menggunakan AWS Fargate, lihat `managedNodeGroups` konfigurasi dalam dokumentasi eksctl.
	
	1. Gunakan file `eksctl` input berikut untuk menghasilkan cluster.
	
	`sample-cluster.yaml` :
	
	`apiVersion: eksctl.io/v1alpha5`
	`kind: ClusterConfig`
	`metadata:`
	`  name: postgresql`
	`  region: us-east-1`
	`  version: "1.29"`
	`accessConfig:`
	`  authenticationMode: API_AND_CONFIG_MAP`
	`availabilityZones:`
	`- us-east-1a`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>- us-east-1b
	- us-east-1c
	nodeGroups:
	  - name: ng-1
	    instanceType:
	 m5.16xlarge
	    desiredCapacity:
	 2
	  - name: ng-2
	    instanceType:
	 m5.16xlarge
	    desiredCapacity:
	 2
	  - name: ng-3
	    instanceType:
	 m5.16xlarge
	    desiredCapacity:
	 2
	vpc:
	  cidr: 192.168.0
	.0/16
	  clusterEndpoints:
	    publicAccess:
	 true
	  nat:
	    gateway: HighlyAva
	ilable
	iamIdentity
	Mappings:
	  - arn: arn:aws:i
	am::<account-id>:r
	ole/<role-name> #
	 update the IAM role
	 ARN created in step 1
	    username: <user-
	name> # Enter the
	 user name per your
	 choice
	    noDuplicateARNs:
	 false</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Jalankan perintah berikut untuk membuat cluster (berikan path file ke `sample-cluster.yaml` file Anda): `eksctl create cluster -f sample-cluster.yaml`	
Validasi status cluster.	Jalankan perintah berikut untuk melihat status node saat ini di cluster: `kubectl get nodes` Jika Anda menemukan kesalahan, lihat bagian pemecahan masalah dokumentasi Amazon EKS.	Administrator AWS, administrator Terraform atau eksctl, administrator Kubernetes

Buat penyedia identitas OIDC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan penyedia IAM OIDC.	Sebagai prasyarat untuk driver Amazon EBS Container Storage Interface (CSI), Anda harus memiliki penyedia IAM OpenID Connect (OIDC) yang sudah ada untuk klaster Anda. Aktifkan penyedia IAM OIDC dengan menggunakan perintah berikut:	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`eksctl utils associate -iam-oidc-provider --region={region} --cluster={YourClusterNameHere} --approve` Untuk informasi selengkapnya tentang langkah ini, lihat dokumentasi Amazon EKS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk driver Amazon EBS CSI.	Gunakan `eksctl` perintah berikut untuk membuat peran IAM untuk driver CSI: `eksctl create iamservic eaccount \ --region {RegionName} \ --name ebs-csi-c ontroller-sa \ --namespace kube-syst em \ --cluster {YourClus terNameHere} \ --attach-policy-ar n arn:aws:iam::aws:p olicy/service-role /AmazonEBSCSIDrive rPolicy \ --approve \ --role-only \ --role-name AmazonEKS _EBS_CSI_DriverRole` Jika Anda menggunakan drive Amazon EBS terenkripsi, Anda harus mengonfigurasi kebijakan lebih lanjut. Untuk petunjuk, lihat dokumentasi driver Amazon EBS SCI.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan driver Amazon EBS CSI.	Gunakan `eksctl` perintah berikut untuk menambahkan driver Amazon EBS CSI: `eksctl create addon \ --name aws-ebs-csi-driver \ --cluster <YourClusterName> service-account-role-arn arn:aws:iam::$(aws sts get-caller-identity \ --query Account \ --output text):role/AmazonEKS_EBS_CSI_DriverRole \ --force`	Administrator AWS

Instal PGO

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori PGO.	Kloning GitHub repositori untuk PGO: `git clone https://github.com/CrunchyData/postgres-operator-examples.git`	AWS DevOps
Berikan detail peran untuk pembuatan akun layanan.	Untuk memberikan akses klaster Amazon EKS ke AWS sumber daya yang diperlukan, tentukan Nama Sumber Daya Amazon (ARN) peran OIDC yang Anda buat sebelumny	Administrator AWS, administrator Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	a dalam file. `service_account.yaml` File ini terletak di folder namespace repositori.
	
	`cd postgres-operator-examples`
	
	`---`
	`metadata:`
	`  annotations:`
	`  eks.amazonaws.com/role-arn: arn:aws:iam::<accountId>:role/<role_name> # Update the OIDC role ARN created earlier`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat namespace dan prasyarat PGO.	1. Jalankan perintah berikut untuk membuat namespace :
	`kubectl apply -k kustomize/install/namespace`
	Ini menetapkan namespace khusus untuk PGO. Jika perlu, Anda dapat memodifikasi `namespace.yml` file dan menetapkan namespace nama yang berbeda.
	2. Jalankan perintah berikut untuk menerapkan konfigurasi default ke cluster:
	`kubectl apply --server-side -k kustomize/install/default`
	`kustomize/install/default` menyediakan konfigurasi default untuk Kubernetes role-based access control (RBAC), custom resource definition (CRD), dan file Kubernetes Manager.
		Administrator Kunernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi pembuatan pod.	Verifikasi bahwa namespace dan konfigurasi default telah dibuat: `kubectl get pods -n postgres-operator`	Administrator AWS, administrator Kubernetes
Verifikasi PVCs.	Gunakan perintah berikut untuk memverifikasi klaim volume persisten (PVCs): `kubectl describe pvc -n postgres-operator`	Administrator AWS, administrator Kubernetes

Membuat dan menyebarkan operator

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat operator.	Merevisi isi file yang terletak di `/kustomize/postgres/postgres.yaml` agar sesuai dengan yang berikut:
	`spec:`
	`  instances:`
	`  - name: pg-1`
	`  replicas: 3`
	`  patroni:`
	`  dynamicConfiguration:`
	`  postgresql:`
	`  pg_hba:`
	`  - "host all all 0.0.0.0/0 trust"`
	`  # this line enabled logical replication`
		Administrator AWS, DBA, administrator Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>with programmatic access
	       - "host all postgres 127.0.0.1/32 md5"
	     synchronous_mode: true
	  users:
	  - name: replicator
	    databases:
	      - testdb
	    options: "REPLICATION"</pre>
	Pembaruan ini melakukan hal berikut:
	• Sesuaikan pengaturan konfigurasi PostgreSQL untuk memfasilitasi akses ke instance PostgreSQL.
	• Sertakan konfigurasi untuk pengguna replikasi , pengguna database, dan superuser untuk mengaktifkan replikasi streaming , akses database, dan manajemen cluster.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan operator.	Menerapkan operator PGO untuk mengaktifkan pengelolaan dan pengoperasian database PostgreSQL yang efisien di lingkungan Kubernetes: `kubectl apply -k kustomize/postgres`	Administrator AWS, DBA, administrator Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi penyebaran.	1. Validasi bahwa operator telah dikerahkan:
	`kubectl get pods -n postgres-operator --selector=postgres-operator.crunchydata.com/instance-set \ -L postgres-operator.crunchydata.com/role`
	2. Verifikasi bahwa sumber daya layanan yang terkait dengan pod operator telah dibuat:
	`kubectl get svc -n postgres-operator`
	Dari output perintah, perhatikan replika utama (`primary_pod_name`) dan baca replika (`read_pod_name`). Anda akan menggunakan ini di langkah selanjutnya.
		Administrator AWS, DBA, administrator Kubernetes

Verifikasi replikasi streaming

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tulis data ke replika utama.	Gunakan perintah berikut untuk terhubung ke replika utama PostgreSQL dan menulis data ke database:	Administrator AWS, administrator Kubernetes

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`kubectl exec -it <primary_pod_name> bash -n postgres-operator`
	
	`psql`
	
	`CREATE TABLE customers (firstname text, customer_id serial, date_created timestamp );`
	`\dt`	

Konfirmasikan bahwa replika baca memiliki data yang sama.	Hubungkan ke replika baca PostgreSQL dan periksa apakah replikasi streaming berfungsi dengan benar:
	
	`kubectl exec -it {read_pod_name} bash -n postgres-operator`
	
	`psql`
	
	`\dt`
	
	Replika baca harus memiliki tabel yang Anda buat di replika utama pada langkah sebelumnya.
		Administrator AWS, administrator Kubernetes

Pemecahan masalah

Isu	Solusi

Pod tidak dimulai.	• Gunakan perintah berikut untuk memeriksa status pod:
	`kubectl get pods -n your-namespace`
	• Periksa log untuk kesalahan apa pun:
	`kubectl logs your-pod-name -n your-namespace`
	• Periksa peristiwa pod untuk setiap peristiwa abnormal yang terkait dengan pod Anda:
	`kubectl describe pod your-pod-name -n your-namespace`

Replika secara signifikan berada di belakang database utama.	• Periksa kelambatan replikasi:
	`SELECT * FROM pg_stat_replication;`
	• Pastikan replika memiliki sumber daya CPU dan memori yang cukup. Periksa batas sumber daya:
	`kubectl describe pod your-replica-pod -n your-namespace`
	• Verifikasi bahwa backend penyimpanan berkinerja optimal. Disk yang lambat I/O dapat menyebabkan kelambatan replikasi.

Anda tidak memiliki visibilitas ke kinerja dan kesehatan cluster PostgreSQL.	• Aktifkan CloudWatch Log Amazon dan pastikan log dikirim ke Amazon CloudWatch untuk dianalisis. Untuk informasi selengkap nya, lihat dokumentasi Amazon EKS.

Isu	Solusi
	• Periksa **`pg_stat_activity`** : `SELECT * FROM pg_stat_activity;`

Replikasi tidak bekerja.	• Periksa konfigurasi utama dengan melihat pengaturan replikasi di`postgresql.conf` :
	`wal_level = replica`
	`max_wal_senders = 10`
	`wal_keep_size = 64  # or wal_keep_segments in older versions`
	• Verifikasi yang `pg_hba.conf` menyertakan izin replikasi:
	`host replication replica_user all md5`
	• Periksa konfigurasi replika. Pastikan bahwa `recovery.conf` atau pengaturan yang setara (`standby.signal` dan`primary_conninfo`) diatur dengan benar pada replika.

Sumber daya terkait

- Amazon Elastic Kubernetes Service (Ikhtisar Opsi Deployment di whitepaper AWS)
- CloudFormation(Ikhtisar Opsi Penerapan di whitepaper AWS)
- Memulai Amazon EKS — eksctl (Panduan Pengguna Amazon EKS)
- Siapkan kubectl dan eksctl (Panduan Pengguna Amazon EKS)
- Buat peran untuk federasi OpenID Connect (Panduan Pengguna IAM)
- Mengkonfigurasi pengaturan untuk AWS CLI (Panduan AWS CLI Pengguna)

- Dokumentasi Crunchy Postgres untuk Kubernetes
- Crunch & Learn: Crunchy Postgres untuk Kubernetes 5.0 (video)

Sederhanakan otentikasi aplikasi dengan TLS timbal balik di Amazon ECS dengan menggunakan Application Load Balancer

Olawale Olaleye dan Shamanth Devagari, Amazon Web Services

Ringkasan

Pola ini membantu Anda menyederhanakan otentikasi aplikasi dan menurunkan beban keamanan dengan TLS bersama di Amazon Elastic Container Service (Amazon ECS) Service Elastic Container ECS) dengan menggunakan Application Load Balancer (ALB). Dengan ALB, Anda dapat mengautentikasi sertifikat klien X.509 dari. AWS Private Certificate Authority Kombinasi yang kuat ini membantu mencapai komunikasi yang aman antara layanan Anda, mengurangi kebutuhan akan mekanisme otentikasi yang kompleks dalam aplikasi Anda. Selain itu, pola menggunakan Amazon Elastic Container Registry (Amazon ECR) Registry (Amazon ECR) untuk menyimpan gambar kontainer.

Contoh dalam pola ini menggunakan gambar Docker dari galeri publik untuk membuat beban kerja sampel pada awalnya. Selanjutnya, gambar Docker baru dibuat untuk disimpan di Amazon ECR. Untuk sumbernya, pertimbangkan sistem berbasis Git seperti GitHub,, atau Bitbucket GitLab, atau gunakan Amazon Simple Storage Service Amazon S3 (Amazon S3). Untuk membuat gambar Docker, pertimbangkan AWS CodeBuild untuk menggunakan gambar berikutnya.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan akses untuk menyebarkan AWS CloudFormation tumpukan. Pastikan Anda memiliki izin pengguna atau peran AWS Identity and Access Management (IAM) untuk digunakan. CloudFormation
- AWS Command Line Interface (AWS CLI) diinstal. Konfigurasikan AWS kredensyal Anda di mesin lokal Anda atau di lingkungan Anda dengan menggunakan AWS CLI atau dengan mengatur variabel lingkungan dalam file. `~/.aws/credentials`
- OpenSSL diinstal.
- Docker diinstal.
- Keakraban dengan yang Layanan AWS dijelaskan dalam Alat.
- Pengetahuan tentang Docker dan NGINX.

Batasan

- Mutual TLS untuk Application Load Balancer hanya mendukung sertifikat klien X.509v3. Sertifikat klien X.509v1 tidak didukung.
- CloudFormation Template yang disediakan dalam repositori kode pola ini tidak termasuk penyediaan CodeBuild proyek sebagai bagian dari tumpukan.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Docker versi 27.3.1 atau yang lebih baru
- AWS CLI versi 2.14.5 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan komponen arsitektur untuk pola ini.

Diagram menunjukkan alur kerja berikut:

1. Buat repositori Git, dan komit kode aplikasi ke repositori.
2. Buat otoritas sertifikat pribadi (CA) di AWS Private CA.
3. Buat CodeBuild proyek. CodeBuildproject Ini dipicu oleh perubahan komit dan membuat gambar Docker dan menerbitkan gambar yang dibangun ke Amazon ECR.
4. Salin rantai sertifikat dan badan sertifikat dari CA, dan unggah bundel sertifikat ke Amazon S3.
5. Buat toko kepercayaan dengan bundel CA yang Anda unggah ke Amazon S3. Kaitkan toko kepercayaan dengan pendengar TLS bersama di Application Load Balancer (ALB).
6. Gunakan CA pribadi untuk menerbitkan sertifikat klien untuk beban kerja kontainer. Juga buat sertifikat TLS pribadi menggunakan AWS Private CA.
7. Impor sertifikat TLS pribadi ke AWS Certificate Manager (ACM), dan gunakan dengan ALB.
8. Beban kerja kontainer `ServiceTwo` menggunakan sertifikat klien yang dikeluarkan untuk mengautentikasi dengan ALB saat berkomunikasi dengan beban kerja kontainer di. `ServiceOne`

9. Beban kerja kontainer `ServiceOne` menggunakan sertifikat klien yang dikeluarkan untuk mengautentikasi dengan ALB saat berkomunikasi dengan beban kerja kontainer di. `ServiceTwo`

Otomatisasi dan skala

Pola ini dapat sepenuhnya otomatis dengan menggunakan CloudFormation, AWS Cloud Development Kit (AWS CDK) , atau operasi API dari SDK untuk menyediakan AWS sumber daya.

Anda dapat menggunakannya AWS CodePipeline untuk mengimplementasikan pipeline continuous integration dan continuous deployment (CI/CD) menggunakan CodeBuild untuk mengotomatiskan proses pembuatan image container dan menerapkan rilis baru ke layanan cluster Amazon ECS.

Alat

Layanan AWS

- AWS Certificate Manager (ACM) membantu Anda membuat, menyimpan, dan memperbarui sertifikat dan kunci SSL/TLS X.509 publik dan pribadi yang melindungi situs web dan aplikasi Anda. AWS
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat untuk menjalankan, menghentikan, dan mengelola kontainer di cluster. Anda dapat menjalankan tugas dan layanan Anda pada infrastruktur tanpa server yang dikelola oleh. AWS Fargate Atau, untuk kontrol lebih lanjut atas infrastruktur, Anda dapat menjalankan tugas dan layanan Anda di klaster instans Amazon Elastic Compute Cloud (Amazon EC2) yang Anda kelola.
- Amazon ECS Exec memungkinkan Anda berinteraksi langsung dengan kontainer tanpa perlu terlebih dahulu berinteraksi dengan sistem operasi wadah host, membuka port masuk, atau mengelola kunci SSH. Anda dapat menggunakan ECS Exec untuk menjalankan perintah di, atau mendapatkan shell ke, wadah yang berjalan pada EC2 instance Amazon atau di. AWS Fargate

- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh EC2 instans Amazon, kontainer, dan alamat IP, di satu atau beberapa Availability Zone. ELB memantau kesehatan target yang terdaftar, dan mengarahkan lalu lintas hanya ke target yang sehat. ELB menskalakan penyeimbang beban Anda saat lalu lintas masuk Anda berubah seiring waktu. Ini dapat secara otomatis menskalakan ke sebagian besar beban kerja.
- AWS Fargatemembantu Anda menjalankan container tanpa perlu mengelola server atau EC2 instans Amazon. Fargate kompatibel dengan Amazon ECS dan Amazon Elastic Kubernetes Service (Amazon EKS). Anda dapat menjalankan tugas dan layanan Amazon ECS Anda dengan jenis peluncuran Fargate atau penyedia kapasitas Fargate. Untuk melakukannya, paketkan aplikasi Anda dalam wadah, tentukan persyaratan CPU dan memori, tentukan kebijakan jaringan dan IAM, dan luncurkan aplikasi. Setiap tugas Fargate memiliki batas isolasi sendiri dan tidak berbagi kernel yang mendasarinya, sumber daya CPU, sumber daya memori, atau elastic network interface dengan tugas lain.
- AWS Private Certificate Authoritymemungkinkan pembuatan hierarki otoritas sertifikat swasta (CA), termasuk root dan bawahan CAs, tanpa biaya investasi dan pemeliharaan pengoperasian CA lokal.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- GitHub, GitLab, dan Bitbucket adalah beberapa sistem kontrol sumber berbasis Git yang umum digunakan untuk melacak perubahan kode sumber.
- NGINX Open Source adalah penyeimbang beban sumber terbuka, cache konten, dan server web. Pola ini menggunakannya sebagai server web.
- OpenSSL adalah perpustakaan open source yang menyediakan layanan yang digunakan oleh implementasi OpenSSL dari TLS dan CMS.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub MTLS-with-Application-Load-Balancer-in-Amazon-ECS.

Praktik terbaik

- Gunakan Amazon ECS Exec untuk menjalankan perintah atau mendapatkan shell ke wadah yang berjalan di Fargate. Anda juga dapat menggunakan ECS Exec untuk membantu mengumpulkan informasi diagnostik untuk debugging.
- Gunakan grup keamanan dan daftar kontrol akses jaringan (ACLs) untuk mengontrol lalu lintas masuk dan keluar antar layanan. Tugas Fargate menerima alamat IP dari subnet yang dikonfigurasi di cloud pribadi virtual (VPC) Anda.

Epik

Buat repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kode sumbernya.	Untuk mengunduh kode sumber pola ini, fork atau kloning repositori GitHub MTLS-with-Application-Load-Balancer-in-Amazon-ECS.	DevOps insinyur
Buat repositori Git.	Untuk membuat repositori Git untuk memuat Dockerfile dan `buildspec.yaml` file, gunakan langkah-langkah berikut: 1. Buat folder di lingkungan virtual Anda. Beri nama dengan nama proyek Anda. 2. Buka terminal di mesin lokal Anda, dan arahkan ke folder ini. 3. Untuk mengkloning repositori MTLS-with-Application-Load-Balancer-in-Amazon-ECS ke direktori	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	proyek Anda, masukkan perintah berikut: `git clone https://github.com/aws-samples/mTLS-with-Application-Load-Balancer-in-Amazon-ECS.git`	

Buat CA dan hasilkan sertifikat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CA pribadi di AWS Private CA.	Untuk membuat otoritas sertifikat pribadi (CA), jalankan perintah berikut di terminal Anda. Ganti nilai dalam variabel contoh dengan nilai Anda sendiri. `export AWS_DEFAULT_REGION="us-west-2"` `export SERVICES_DOMAIN="www.example.com"` `export ROOT_CA_ARN=`aws acm-pca create-certificate-authority \` `--certificate-authority-type ROOT \` `--certificate-authority-configuration \` `"KeyAlgorithm=RSA_2048,`	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> SigningAlgorithm=S
	HA256WITHRSA,
	    Subject={
	        Country=US,
	        State=WA,
	        Locality=
	Seattle,
	        Organizat
	ion=Build on AWS,
	        Organizat
	ionalUnit=mTLS Amazon
	 ECS and ALB Example,
	        CommonName=
	${SERVICES_DOMAIN}}" \
	        --query
	 CertificateAuthori
	tyArn --output text`</pre> Untuk detail selengkapnya, lihat Membuat CA pribadi AWS Private CA di dalam AWS dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat dan instal sertifikat CA pribadi Anda.	Untuk membuat dan menginstal sertifikat untuk CA root pribadi Anda, jalankan perintah berikut di terminal Anda:
	1. Buat permintaan penandatanganan sertifikat (CSR).
	`ROOT_CA_CSR=`aws acm-pca get-certificate-authority-csr \ --certificate-authority-arn ${ROOT_CA_ARN} \ --query Csr --output text``
	2. Terbitkan sertifikat root.
	`AWS_CLI_VERSION=$( aws --version 2>&1 | cut -d/ -f2 | cut -d. -f1) [[ ${AWS_CLI_VERSION} -gt 1 ]] && ROOT_CA_CSR="$(echo ${ROOT_CA_CSR} | base64)" ROOT_CA_CERT_ARN= `aws acm-pca issue-certificate \ --certificate-authority-arn ${ROOT_CA_ARN} \ --template-arn arn:aws:acm-pca::: template/RootCACertificate/V1 \`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre> --signing- algorithm SHA256WIT HRSA \ --validity Value=10,Type=YEARS \ --csr "${ROOT_C A_CSR}" \ --query Certifica teArn --output text`</pre> 3. Ambil sertifikat akar. <pre>ROOT_CA_CERT=`aws acm-pca get-certi ficate \ --certificate-arn ${ROOT_CA_CERT_ARN} \ --certificate- authority-arn ${ROOT_CA_ARN} \ --query Certifica te --output text` # store for later use aws acm-pca get-certi ficate \ --certificate-arn ${ROOT_CA_CERT_ARN} \ --certificate- authority-arn ${ROOT_CA_ARN} \ --query Certifica te --output text > ca-cert.pem</pre> 4. Impor sertifikat CA root untuk menginstalnya di CA.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`[[ ${AWS_CLI_VERSION} -gt 1 ]] && ROOT_CA_CERT="$(echo ${ROOT_CA_CERT} | base64)"`
	
	`aws acm-pca import-certificate-authority-certificate \`
	`--certificate-authority-arn $ROOT_CA_ARN \`
	`--certificate "${ROOT_CA_CERT}"`
	
	Untuk detail selengkapnya, lihat Menginstal sertifikat CA di AWS dokumentasi.	

Minta sertifikat terkelola.	Untuk meminta sertifikat pribadi AWS Certificate Manager untuk digunakan dengan ALB pribadi Anda, gunakan perintah berikut: `export TLS_CERTIFICATE_ARN=`aws acm request-certificate \` `--domain-name "*.${DOMAIN_DOMAIN}" \` `--certificate-authority-arn ${ROOT_CA_ARN} \` `--query CertificateArn --output text``	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Gunakan CA pribadi untuk mengeluarkan sertifikat klien.	• Untuk membuat permintaan penandatanganan sertifikat (CSR) untuk dua layanan, gunakan AWS CLI perintah berikut:
	
	`openssl req -out client_csr1.pem -new -newkey rsa:2048 -nodes -keyout client_private-key1.pem`
	
	`openssl req -out client_csr2.pem -new -newkey rsa:2048 -nodes -keyout client_private-key2.pem`
	
	Perintah ini mengembalikan CSR dan kunci pribadi untuk dua layanan.
	
	• Untuk mengeluarkan sertifikat untuk layanan, jalankan perintah berikut untuk menggunakan CA pribadi yang Anda buat:
	
	`SERVICE_ONE_CERT_ARN=`aws acm-pca issue-certificate \`
		DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--certificate-authority-arn ${ROOT_CA_ARN} \` `--csr fileb://client_csr1.pem \` `--signing-algorithm "SHA256WITHRSA" \` `--validity Value=5,Type="YEARS" --query CertificateArn --output text` `echo "SERVICE_ONE_CERT_ARN: ${SERVICE_ONE_CERT_ARN}"` `aws acm-pca get-certificate \` `--certificate-authority-arn ${ROOT_CA_ARN} \` `--certificate-arn ${SERVICE_ONE_CERT_ARN} \` `| jq -r '.Certificate' > client_cert1.cert` `SERVICE_TWO_CERT_ARN=`aws acm-pca issue-certificate \` `--certificate-authority-arn ${ROOT_CA_ARN} \` `--csr fileb://client_csr2.pem \` `--signing-algorithm "SHA256WITHRSA" \` `--validity Value=5,Type="YEARS" --query CertificateArn --output text`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`echo "SERVICE_TWO_CERT_ARN: ${SERVICE_TWO_CERT_ARN}"`
	
	`aws acm-pca get-certificate \`
	`--certificate-authority-arn ${ROOT_CA_ARN} \`
	`--certificate-arn ${SERVICE_TWO_CERT_ARN} \`
	`| jq -r '.Certificate' > client_cert2.cert`
	
	Untuk informasi selengkapnya, lihat Menerbitkan sertifikat entitas akhir pribadi dalam dokumentasi. AWS	

Menyediakan layanan AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ketentuan Layanan AWS dengan CloudFormation template.	Untuk menyediakan virtual private cloud (VPC), Amazon ECS cluster, layanan Amazon ECS, Application Load Balancer, dan Amazon Elastic Container Registry (Amazon ECR), gunakan template. CloudFormation	DevOps insinyur
Dapatkan variabel.	Verifikasi bahwa Anda memiliki kluster Amazon ECS	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dengan dua layanan yang berjalan. Untuk mengambil detail sumber daya dan menyimpannya sebagai variabel, gunakan perintah berikut:
	
	`export LoadBalancerDNS=$(aws cloudformation describe-stacks --stack-name ecs-mtls \ --output text \ --query 'Stacks[0].Outputs[?OutputKey==`LoadBalancerDNS`].OutputValue')`
	
	`export ECRRepositoryUri=$(aws cloudformation describe-stacks --stack-name ecs-mtls \ --output text \ --query 'Stacks[0].Outputs[?OutputKey==`ECRRepositoryUri`].OutputValue')`
	
	`export ECRRepositoryServiceOneUri=$(aws cloudformation describe-stacks --stack-name ecs-mtls \ --output text \ --query 'Stacks[0].Outputs[?OutputKey==`ECRRepositoryServiceOneUri`].OutputValue')`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`export ECRReposi toryServiceTwoUri= $(aws cloudformation describe-stacks -- stack-name ecs-mtls \ --output text \ --query 'Stacks[0 ].Outputs[?OutputK ey==`ECRRepository ServiceTwoUri`].Ou tputValue')`
	
	`export ClusterName= $(aws cloudformation describe-stacks -- stack-name ecs-mtls \ --output text \ --query 'Stacks[0 ].Outputs[?OutputK ey==`ClusterName`] .OutputValue')`
	
	`export BucketName= $(aws cloudformation describe-stacks -- stack-name ecs-mtls \ --output text \ --query 'Stacks[0 ].Outputs[?OutputK ey==`BucketName`]. OutputValue')`
	
	`export Service1L istenerArn=$(aws cloudformation describe-stacks -- stack-name ecs-mtls \ --output text \ --query 'Stacks[0 ].Outputs[?OutputK ey==`Service1Liste`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nerArn`].OutputVal ue')  export Service2L istenerArn=$(aws cloudformation describe-stacks -- stack-name ecs-mtls \ --output text \ --query 'Stacks[0 ].Outputs[?OutputK ey==`Service2Liste nerArn`].OutputVal ue')	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodeBuild proyek.	Untuk menggunakan CodeBuild proyek untuk membuat gambar Docker untuk layanan Amazon ECS Anda, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan buka CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/. 2. Buat proyek baru. Untuk Sumber, pilih repositori Git yang Anda buat. Untuk informasi tentang berbagai jenis integrasi repositori Git, lihat Bekerja dengan koneksi dalam dokumenta si. AWS 3. Konfirmasikan bahwa mode Privileged diaktifkan. Untuk membangun gambar Docker, mode ini diperluka n. Jika tidak, gambar tidak akan berhasil dibangun. 4. Gunakan `buildspec .yaml` file kustom yang dibagikan untuk setiap layanan. 5. Berikan nilai untuk nama dan deskripsi proyek. Untuk detail selengkapnya, lihat Membuat proyek build	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS CodeBuild di dalam AWS dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bangun gambar Docker.	Anda dapat menggunakan CodeBuild untuk melakukan proses pembuatan gambar. CodeBuild memerlukan izin untuk berinteraksi dengan Amazon ECR dan bekerja dengan Amazon S3.
	
	Sebagai bagian dari proses, image Docker dibuat dan didorong ke registri Amazon ECR. Untuk detail tentang templat dan kode, lihat Informasi tambahan.
	
	(Opsional) Untuk membangun secara lokal untuk tujuan pengujian, gunakan perintah berikut:
	
	`# login to ECR`
	`aws ecr get-login`
	`-password | docker`
	`login --username`
	`AWS --password-stdin`
	`$ECRRepositoryUri`
	
	`# build image for`
	`service one`
	`cd /service1`
	`aws s3 cp s3://$Buc`
	`ketName/serviceone/`
	`service1/ --recursive`
	`docker build -t`
	`$ECRRepositoryServ`
	`iceOneUri .`
	`docker push $ECRRepos`
	`itoryServiceOneUri`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`# build image for service two`
	`cd ../service2`
	`aws s3 cp s3://$BucketName/servicetwo/service2/ --recursive`
	`docker build -t $ECRRepositoryServiceTwoUri .`
	`docker push $ECRRepositoryServiceTwoUri`	

Aktifkan TLS timbal balik

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah sertifikat CA ke Amazon S3.	Untuk mengunggah sertifikat CA ke bucket Amazon S3, gunakan perintah contoh berikut: `aws s3 cp ca-cert.pem s3://$BucketName/acm-trust-store/`	AWS DevOps, DevOps insinyur
Buat toko kepercayaan.	Untuk membuat toko kepercayaan, gunakan perintah contoh berikut: `TrustStoreArn=`aws elbv2 create-trust-store --name acm-pca-trust-certs \` `--ca-certificates-bundle-s3-bucket $BucketName \`	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--ca-certificates-bundle-s3-key acm-trust-store/ca-cert.pem --query 'TrustStores[].TrustStoreArn' --output text`	

Unggah sertifikat klien.	Untuk mengunggah sertifikat klien ke Amazon S3 untuk gambar Docker, gunakan perintah contoh berikut:
	`# for service one`
	`aws s3 cp client_cert1.cert s3://$BucketName/serviceone/`
	`aws s3 cp client_private-key1.pem s3://$BucketName/serviceone/`
	
	`# for service two`
	`aws s3 cp client_cert2.cert s3://$BucketName/servicetwo/`
	`aws s3 cp client_private-key2.pem s3://$BucketName/servicetwo/`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ubah pendengar.	Untuk mengaktifkan TLS timbal balik di ALB, modifikasi pendengar HTTPS dengan menggunakan perintah berikut:
	
	`aws elbv2 modify-listener \`
	`  --listener-arn $Service1ListenerArn \`
	`  --certificates CertificateArn=$TLS_CERTIFICATE_ARN_TWO \`
	`  --ssl-policy ELBSecurityPolicy-2016-08 \`
	`  --protocol HTTPS \`
	`  --port 8080 \`
	`  --mutual-authentication Mode=verify,TrustStoreArn=$TrustStoreArn,IgnoreClientCertificateExpiry=false`
	
	`aws elbv2 modify-listener \`
	`  --listener-arn $Service2ListenerArn \`
	`  --certificates CertificateArn=$TLS_CERTIFICATE_ARN_TWO \`
	`  --ssl-policy ELBSecurityPolicy-2016-08 \`
	`  --protocol HTTPS \`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--port 8090 \`
	`--mutual-authentic ation Mode=veri fy,TrustStoreArn=$ TrustStoreArn,Igno reClientCertificat eExpiry=false`
	
	Untuk informasi selengkap nya, lihat Mengonfigurasi TLS timbal balik pada Applicati on Load Balancer dalam dokumentasi. AWS	

Perbarui layanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui definisi tugas Amazon ECS.	Untuk memperbarui definisi tugas Amazon ECS, ubah `image` parameter dalam revisi baru. Untuk mendapatkan nilai untuk masing-masing layanan, perbarui definisi tugas dengan Uri gambar Docker baru yang Anda buat di langkah sebelumnya: `echo $ECRRepositoryServ iceOneUri` atau `echo $ECRRepositoryServ iceTwoUri`	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"containerDefiniti
	ons": [
	{
	"name":
	"nginx",
	"image":
	"public.ecr.aws/ng
	inx/nginx:latest",
	# <----- change to new
	Uri
	"cpu": 0,</pre>
	Untuk informasi selengkap nya, lihat Memperbarui definisi tugas Amazon ECS menggunakan konsol dalam AWS dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui layanan Amazon ECS.	Perbarui layanan dengan definisi tugas terbaru. Definisi tugas ini adalah cetak biru untuk gambar Docker yang baru dibangun, dan berisi sertifikat klien yang diperlukan untuk otentikasi TLS bersama. Untuk memperbarui layanan, gunakan prosedur berikut: 1. Buka konsol Amazon ECS di https://console.aws.amazon.com/ecs/v2. 2. Pada halaman Clusters, pilih cluster. 3. Pada halaman detail cluster, di bagian Layanan, pilih kotak centang di sebelah layanan, lalu pilih Perbarui. 4. Agar layanan Anda memulai penerapan baru, pilih Paksa penerapan baru. 5. Untuk definisi Tugas, pilih keluarga definisi tugas dan revisi terbaru. 6. Pilih Perbarui. Ulangi langkah-langkah untuk layanan lainnya.	Administrator AWS, AWS DevOps, DevOps insinyur

Akses aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin URL aplikasi.	Gunakan konsol Amazon ECS untuk melihat tugas. Ketika status tugas telah diperbaru i ke Running, pilih tugas. Di bagian Tugas, salin ID tugas.	Administrator AWS, AWS DevOps

Uji aplikasi Anda.	Untuk menguji aplikasi Anda, gunakan ECS Exec untuk mengakses tugas.
	1. Untuk layanan satu, gunakan perintah berikut:
	`container="nginx"`
	`ECS_EXEC_TASK_ARN ="<TASK ARN>" aws ecs execute-c ommand --cluster $ClusterName \ --task $ECS_EXEC _TASK_ARN \ --container $container \ --interactive \ --command "/bin/ bash"`
	2. Dalam wadah tugas satu layanan, gunakan perintah berikut untuk memasukka n penyeimbang beban internal `url` dan port pendengar yang menunjuk ke layanan dua. Kemudian
		Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	tentukan jalur ke sertifikat klien untuk menguji aplikasi:
	
	`curl -kvs https://<internal-alb-url>:8090 --key /usr/local/share/ca-certificates/client.key --cert /usr/local/share/ca-certificates/client.crt`
	
	3. Dalam wadah tugas dua layanan, gunakan perintah berikut untuk memasukkan penyeimbang beban internal `url` dan port pendengar yang menunjuk ke service one. Kemudian tentukan jalur ke sertifikat klien untuk menguji aplikasi:
	
	`curl -kvs https://<internal-alb-url>:8080 --key /usr/local/share/ca-certificates/client.key --cert /usr/local/share/ca-certificates/client.crt`
	
	Note
	`-k`Bendera dalam perintah curl (sebagai bagian dari`-kvs`) menonaktifkan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	validasi sertifikat SSL. Anda dapat menghapus tanda ini saat menggunakan sertifikat SSL yang cocok dengan nama domain Anda, memungkinkan validasi sertifikat yang tepat.	

Sumber daya terkait

Dokumentasi Amazon ECS

- Membuat definisi tugas Amazon ECS menggunakan konsol
- Membuat Image Kontainer untuk Digunakan di Amazon ECS
- Cluster Amazon ECS
- Amazon ECS untuk AWS Fargate
- Praktik terbaik jaringan Amazon ECS
- Parameter definisi layanan Amazon ECS

AWS Sumber daya lainnya

- Bagaimana cara menggunakan CA AWS pribadi untuk mengonfigurasi mTL pada Application Load Balancer? (AWS re:Post)

Informasi tambahan

Mengedit Dockerfile

Kode berikut menunjukkan perintah yang Anda edit di Dockerfile untuk layanan 1:

```
FROM public.ecr.aws/nginx/nginx:latest
```

```
WORKDIR /usr/share/nginx/html
RUN echo "Returning response from Service 1: Ok" > /usr/share/nginx/html/index.html
ADD client_cert1.cert client_private-key1.pem /usr/local/share/ca-certificates/
RUN chmod -R 400 /usr/local/share/ca-certificates/
```

Kode berikut menunjukkan perintah yang Anda edit di Dockerfile untuk layanan 2:

```
FROM public.ecr.aws/nginx/nginx:latest
WORKDIR /usr/share/nginx/html
RUN echo "Returning response from Service 2: Ok" > /usr/share/nginx/html/index.html
ADD client_cert2.cert client_private-key2.pem /usr/local/share/ca-certificates/
RUN chmod -R 400 /usr/local/share/ca-certificates/
```

Jika Anda membuat image Docker dengan CodeBuild, `buildspec` file menggunakan nomor CodeBuild build untuk mengidentifikasi versi gambar secara unik sebagai nilai tag. Anda dapat mengubah `buildspec` file agar sesuai dengan kebutuhan Anda, seperti yang ditunjukkan dalam kode `buildspec` kustom berikut:

```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --
username AWS --password-stdin $ECR_REPOSITORY_URI
      - COMMIT_HASH=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c 1-7)
      - IMAGE_TAG=${COMMIT_HASH:=latest}
  build:
    commands:
        # change the S3 path depending on the service
      - aws s3 cp s3://$YOUR_S3_BUCKET_NAME/serviceone/ $CodeBuild_SRC_DIR/ --
recursive
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $ECR_REPOSITORY_URI:latest .
      - docker tag $ECR_REPOSITORY_URI:latest $ECR_REPOSITORY_URI:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker images...
      - docker push $ECR_REPOSITORY_URI:latest
```

```
      - docker push $ECR_REPOSITORY_URI:$IMAGE_TAG
      - echo Writing image definitions file...
      # for ECS deployment reference
      - printf '[{"name":"%s","imageUri":"%s"}]' $CONTAINER_NAME $ECR_REPOSITORY_URI:
$IMAGE_TAG > imagedefinitions.json

artifacts:
  files:
    - imagedefinitions.json
```

Lebih banyak pola

- Menilai kesiapan aplikasi untuk migrasi ke AWS Cloud dengan menggunakan CAST Highlight
- Mengotomatiskan penghapusan AWS CloudFormation tumpukan dan sumber daya terkait
- Otomatiskan manajemen pipeline dinamis untuk menerapkan solusi perbaikan terbaru di lingkungan Gitflow dengan menggunakan dan AWS Service Catalog AWS CodePipeline
- Membuat CI/CD pipeline dan kluster Amazon ECS secara otomatis untuk layanan mikro menggunakan AWS CDK
- Buat dan dorong gambar Docker ke Amazon ECR menggunakan GitHub Actions dan Terraform
- Kontainerisasi beban kerja mainframe yang telah dimodernisasi oleh Blu Age
- Buat parser log khusus untuk Amazon ECS menggunakan router log Firelens
- Terapkan sistem agen di Amazon Bedrock dengan framework CreWai dengan menggunakan Terraform
- Menerapkan lingkungan untuk aplikasi Blu Age kontainer dengan menggunakan Terraform
- Menerapkan logika preprocessing ke dalam model MLdalam satu titik akhir menggunakan pipeline inferensi di Amazon SageMaker
- Terapkan beban kerja dari DevOps pipeline Azure ke kluster Amazon EKS pribadi
- Menerapkan diagnostik Kubernetes yang didukung AI dan pemecahan masalah dengan integrasi K8SGPT dan Amazon Bedrock
- Kelola blue/green penerapan layanan mikro ke beberapa akun dan Wilayah dengan menggunakan layanan kode AWS dan kunci Multi-wilayah AWS KMS
- Mengelola aplikasi kontainer lokal dengan menyiapkan Amazon ECS Anywhere dengan AWS CDK
- Bermigrasi dari Oracle WebLogic ke Apache Tomcat (ToMee) di Amazon ECS
- Siapkan ruang data minimum yang layak untuk berbagi data antar organisasi
- Memodernisasi aplikasi Formulir Web ASP.NET di AWS
- Pantau repositori Amazon ECR untuk izin wildcard menggunakan AWS dan AWS Config CloudFormation
- Memantau aktivitas aplikasi dengan menggunakan Wawasan CloudWatch Log
- Siapkan CI/CD pipeline untuk beban kerja hybrid di Amazon ECS Anywhere dengan menggunakan AWS CDK dan GitLab
- Siapkan end-to-end enkripsi untuk aplikasi di Amazon EKS menggunakan cert-manager dan Let's Encrypt

- Sederhanakan penerapan aplikasi multi-tenant Amazon EKS dengan menggunakan Flux
- Merampingkan alur kerja pembelajaran mesin dari pengembangan lokal hingga eksperimen yang dapat diskalakan dengan menggunakan SageMaker AI dan Hydra
- Struktur proyek Python dalam arsitektur heksagonal menggunakan AWS Lambda
- Uji AWS infrastruktur dengan menggunakan LocalStack dan Tes Terraform
- Koordinasikan ketergantungan sumber daya dan eksekusi tugas dengan menggunakan konstruksi AWS Fargate WaitCondition kait
- Gunakan agen Amazon Bedrock untuk mengotomatiskan pembuatan kontrol entri akses di Amazon EKS melalui petunjuk berbasis teks

Nirserver

Topik

- Buat aplikasi seluler React Native tanpa server dengan menggunakan AWS Amplify
- Mengelola penghuni di beberapa produk SaaS pada satu bidang kontrol
- Konsolidasikan pembuatan URL presigned Amazon S3 dan unduhan objek dengan menggunakan titik akhir yang terkait dengan alamat IP statis
- Membuat EventBridge koneksi Amazon lintas akun di suatu organisasi
- Mengirimkan data DynamoDB ke Amazon S3 menggunakan Kinesis Data Streams dan Firehose dengan AWS CDK
- Menerapkan versi API berbasis jalur dengan menggunakan domain khusus di Amazon API Gateway
- Impor pustaka psycopg2 AWS Lambda untuk berinteraksi dengan database PostgreSQL Anda
- Integrasikan Amazon API Gateway dengan Amazon SQS untuk menangani REST asinkron APIs
- Memproses peristiwa secara asinkron dengan Amazon API Gateway dan AWS Lambda
- Memproses peristiwa secara asinkron dengan Amazon API Gateway dan Amazon DynamoDB Streams
- Memproses peristiwa secara asinkron dengan Amazon API Gateway, Amazon SQS, dan AWS Fargate
- Jalankan tugas AWS Systems Manager Automation secara sinkron dari AWS Step Functions
- Jalankan pembacaan paralel objek S3 dengan menggunakan Python dalam suatu fungsi AWS Lambda
- Kirim data telemetri dari AWS Lambda ke OpenSearch untuk analitik dan visualisasi waktu nyata
- Siapkan router sel tanpa server untuk arsitektur berbasis sel
- Siapkan akses pribadi ke bucket Amazon S3 melalui titik akhir VPC
- Memecahkan masalah status AWS Step Functions dengan menggunakan Amazon Bedrock
- Lebih banyak pola

Buat aplikasi seluler React Native tanpa server dengan menggunakan AWS Amplify

Deekshitulu Pentakota, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara membuat backend tanpa server untuk aplikasi seluler React Native dengan menggunakan AWS Amplify dan layanan AWS berikut:

- AWS AppSync
- Amazon Cognito
- Amazon DynamoDB

Setelah Anda mengonfigurasi dan menerapkan backend aplikasi dengan menggunakan Amplify, Amazon Cognito mengautentikasi pengguna aplikasi dan memberi wewenang kepada mereka untuk mengakses aplikasi. AWS AppSync kemudian berinteraksi dengan aplikasi frontend dan dengan tabel DynamoDB backend untuk membuat dan mengambil data.

> **Note**
>
> Pola ini menggunakan aplikasi "ToDoList" sederhana sebagai contoh, tetapi Anda dapat menggunakan prosedur serupa untuk membuat aplikasi seluler React Native apa pun.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Amplify Command Line Interface (Amplify CLI), diinstal dan dikonfigurasi
- XCode (versi apa pun)
- Microsoft Visual Studio (versi apa pun, editor kode apa pun, editor teks apa pun)
- Keakraban dengan Amplify
- Keakraban dengan Amazon Cognito

- Keakraban dengan AWS AppSync
- Keakraban dengan DynamoDB
- Keakraban dengan Node.js
- Keakraban dengan npm
- Keakraban dengan React dan React Native
- Keakraban dengan JavaScript dan ECMAScript 6 () ES6
- Keakraban dengan GraphQL

Arsitektur

Diagram berikut menunjukkan contoh arsitektur untuk menjalankan backend aplikasi seluler React Native di AWS Cloud:

Diagram menunjukkan arsitektur berikut:

1. Amazon Cognito mengautentikasi pengguna aplikasi dan mengizinkan mereka untuk mengakses aplikasi.
2. Untuk membuat dan mengambil data, AWS AppSync menggunakan GraphQL API untuk berinteraksi dengan aplikasi frontend dan tabel DynamoDB backend.

Alat

Layanan AWS

- AWS Amplify adalah seperangkat alat dan fitur yang dibuat khusus yang membantu pengembang web dan seluler frontend dengan cepat membangun aplikasi full-stack di AWS.
- AWS AppSync menyediakan antarmuka GraphQL yang dapat diskalakan yang membantu pengembang aplikasi menggabungkan data dari berbagai sumber, termasuk Amazon DynamoDB, AWS Lambda, dan HTTP. APIs
- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.

Kode

Kode untuk aplikasi sampel yang digunakan dalam pola ini tersedia di ios-todo-app repositori GitHub aws-amplify-react-native-. Untuk menggunakan file sampel, ikuti instruksi di bagian Epics dari pola ini.

Epik

Buat dan jalankan aplikasi React Native Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan lingkungan pengembangan React Native.	Untuk instruksi, lihat Menyiapkan lingkungan pengembangan dalam dokumentasi React Native.	Pengembang aplikasi

Buat dan jalankan aplikasi seluler ToDoList React Native di iOS Simulator.	1. Buat direktori proyek aplikasi seluler React Native baru di lingkungan lokal Anda dengan menjalankan perintah berikut di jendela terminal baru:
	`npx react-native init ToDoListAmplify`
	2. Arahkan ke direktori root proyek dengan menjalankan perintah berikut:
	`cd ToDoListAmplify`
	3. Jalankan aplikasi dengan menjalankan perintah berikut:
	`npx react-native run-ios`
		Pengembang aplikasi

Inisialisasi lingkungan backend baru untuk aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat layanan backend yang diperlukan untuk mendukung aplikasi di Amplify.	1. Di lingkungan lokal Anda, jalankan perintah berikut dari direktori root proyek (ToDoListAmplify):
	`amplify init`
	2. Sebuah prompt muncul yang meminta Anda untuk memberikan informasi tentang aplikasi. Masukkan informasi yang diperlukan berdasarkan kasus penggunaan Anda. Kemudian, tekan Enter.
	
	Untuk penyiapan ToDoList aplikasi yang digunakan dalam pola ini, terapkan konfigurasi contoh berikut.
	
	Contoh pengaturan konfigurasi aplikasi React Native Amplify
	
	`? Name: ToDoListAmplify`
	
	`? Environment: dev`
	
	`? Default editor: Visual Studio Code`
	
	`? App type: javascript`
	
	`? Javascript framework : react-native`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`? Source Directory Path: src`
	
	`? Distribution Directory Path: /`
	
	`? Build Command: npm run-script build`
	
	`? Start Command: npm run-script start`
	
	`? Select the authentication method you want to use: AWS profile`
	
	`? Please choose the profile you want to use: default`
	
	Untuk informasi selengkapnya, lihat Membuat backend Amplify baru di dokumentasi Amplify Dev Center.
	
	Note
	`amplify init`Perintah menyediakan sumber daya berikut dengan menggunakan AWS CloudFormation:
	
	• Peran AWS Identity and Access Management (IAM) untuk pengguna yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	diautentikasi dan tidak diautentikasi (Peran Auth dan Peran Unauth) • Bucket Amazon Simple Storage Service (Amazon S3) untuk penerapan (untuk contoh aplikasi pola ini, Amplify-meta.json) • Lingkungan backend di Amplify Hosting	

Tambahkan autentikasi Amazon Cognito ke aplikasi Amplify React Native

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat layanan otentikasi Amazon Cognito.	1. Di lingkungan lokal Anda, jalankan perintah berikut dari direktori root proyek (ToDoListAmplify): `amplify add auth` 2. Muncul prompt yang meminta Anda untuk memberikan informasi tentang pengaturan konfigurasi layanan otentikasi. Masukkan informasi yang diperlukan berdasarkan kasus penggunaan Anda. Kemudian, tekan Enter. Untuk penyiapan ToDoList aplikasi yang digunakan dalam	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	pola ini, terapkan konfigurasi contoh berikut.
	
	Contoh pengaturan konfigurasi layanan otentikasi
	
	`? Do you want to use the default authentication and security configuration? \`
	`Default configuration`
	
	`? How do you want users to be able to sign in? \`
	`Username`
	
	`? Do you want to configure advanced settings? \`
	`No, I am done`
	
	Note
	`amplify add auth`Perintah membuat folder, file, dan file dependensi yang diperlukan dalam folder lokal (amplify) dalam direktori root proyek. Untuk pengaturan ToDoList aplikasi yang digunakan dalam pola	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ini, aws-exports.js dibuat untuk tujuan ini.	

Menerapkan layanan Amazon Cognito ke AWS Cloud.	1. Dari direktori root proyek, jalankan perintah Amplify CLI berikut:
	`amplify push`
	2. Prompt untuk mengonfir masi penerapan muncul. Masukkan Ya. Kemudian, tekan Enter.
	Note
	Untuk melihat layanan yang diterapkan di project Anda, buka konsol Amplify dengan menjalankan perintah berikut:
	`amplify console`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal library Amplify yang diperlukan untuk React Native dan CocoaPods dependensi untuk iOS.	1. Instal pustaka klien sumber terbuka Amplify yang diperlukan dengan menjalankan perintah berikut dari direktori root proyek:
	`npm install aws-amplify aws-amplify-react-native \ amazon-cognito-identity-js @react-native-community/netinfo \ @react-native-async-storage/async-storage`
	2. Instal CocoaPods dependensi yang diperlukan untuk iOS dengan menjalankan perintah berikut:
	`npx pod-install`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Impor dan konfigurasikan layanan Amplify.	Dalam file titik masuk aplikasi (misalnya, App.js), impor dan muat file konfigurasi layanan Amplify dengan memasukkan baris kode berikut:
	`import Amplify from 'aws-amplify'`
	`import config from './src/aws-exports'`
	`Amplify.configure(config)`
	Note
	Jika Anda menerima kesalahan setelah mengimpor layanan Amplify di file titik masuk aplikasi, hentikan aplikasi. Kemudian, buka XCode dan pilih ToDoListAmplify.xc workspace dari folder iOS proyek dan jalankan aplikasi.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file titik masuk aplikasi Anda untuk menggunakan komponen Higher-order (HOC) withAuthenticator.	**Note** `withAuthenticator` HOC menyediakan alur kerja login, pendaftaran, dan lupa kata sandi di aplikasi Anda hanya dengan menggunakan beberapa baris kode. Untuk informasi selengkapnya, lihat Opsi 1: Menggunakan komponen UI pra-build di Amplify Dev Center. Juga, komponen tingkat tinggi dalam dokumentasi React. 1. Dalam file titik masuk aplikasi (misalnya, App.js), impor `withAuthenticator` HOC dengan memasukkan baris kode berikut: `import { withAuthenticator } from 'aws-amplify-react-native'`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Ekspor WithAuthenticator HOC dengan memasukkan kode berikut:
	`export default withAuthenticator(App)`
	
	denganContoh kode HOC Authenticator
	
	`import Amplify from 'aws-amplify'`
	`import config from './src/aws-exports'`
	`Amplify.configure(config)`
	`import { withAuthenticator } from 'aws-amplify-react-native';`
	
	`const App = () => {`
	`  return null;`
	`};`
	
	`export default withAuthenticator(App);`
	
	Note
	Di iOS Simulator, aplikasi menampilkan layar login yang disediakan oleh	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	layanan Amazon Cognito.	
Uji pengaturan layanan otentikasi.	Di iOS Simulator, lakukan hal berikut: 1. Buat akun baru di aplikasi dengan menggunakan alamat email asli. Kode verifikasi kemudian dikirim ke email yang terdaftar. 2. Verifikasi pengaturan akun dengan menggunakan kode yang Anda terima di email verifikasi. 3. Masukkan nama pengguna dan kata sandi yang Anda buat. Kemudian, pilih Masuk. Layar selamat datang muncul. **Note** Anda juga dapat membuka konsol Amazon Cognito dan memeriksa apakah pengguna baru telah dibuat di Identity Pool atau tidak.	Pengembang aplikasi

Connect AWS AppSync API dan database DynamoDB ke aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database AWS AppSync API dan DynamoDB.	1. Tambahkan AWS AppSync API ke aplikasi Anda dan sediakan database DynamoDB secara otomatis dengan menjalankan perintah Amplify CLI berikut dari direktori root proyek: `amplify add api` 2. Muncul prompt yang meminta Anda untuk memberikan informasi tentang pengaturan konfigurasi API dan database. Masukkan informasi yang diperluka n berdasarkan kasus penggunaan Anda. Kemudian, tekan Enter. Amplify CLI membuka file skema GraphQL di editor teks Anda. Untuk penyiapan ToDoList aplikasi yang digunakan dalam pola ini, terapkan konfigurasi contoh berikut. Contoh pengaturan konfigurasi API dan database	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	? Please select from one of the below mentioned services: \ GraphQL ? Provide API name: todolistamplify ? Choose the default authorization type for the API \ Amazon Cognito User Pool Do you want to use the default authentication and security configura tion ? Default configuration How do you want users to be able to sign in? \ Username Do you want to configure advanced settings? \ No, I am done. ? Do you want to configure advanced settings for the GraphQL API \ No, I am done. ? Do you have an annotated GraphQL schema? \ No ? Choose a schema template: \	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Single object with fields (e.g., "Todo" with ID, name, description)
	
	? Do you want to edit the schema now? \
	Yes
	
	Contoh skema GraphQL
	
	type Todo @model {
	id: ID!
	name: String!
	description: String
	}	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan AWS AppSync API.	1. Di direktori root proyek, jalankan perintah Amplify CLI berikut: `amplify push` 2. Muncul prompt yang meminta Anda untuk memberikan informasi lebih lanjut tentang pengaturan konfigura si API dan database. Masukkan informasi yang diperlukan berdasarkan kasus penggunaan Anda. Kemudian, tekan Enter. Aplikasi Anda sekarang dapat berinteraksi dengan AWS AppSync API. Untuk penyiapan ToDoList aplikasi yang digunakan dalam pola ini, terapkan konfigurasi contoh berikut. Contoh pengaturan konfigurasi AWS AppSync API **Note** Konfigurasi berikut membuat GraphQL API di AppSync AWS dan tabel Todo di Dynamo DB.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	? Are you sure you want to continue? Yes ? Do you want to generate code for your newly created GraphQL API Yes ? Choose the code generation language target javascript ? Enter the file name pattern of graphql queries, mutations and subscriptions src/graphql/**/*.js ? Do you want to generate/update all possible GraphQL operations - \ queries, mutations and subscriptions Yes ? Enter maximum statement depth \ [increase from default if your schema is deeply nested] 2	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hubungkan frontend aplikasi ke AWS AppSync API.	Untuk menggunakan ToDoList aplikasi contoh yang disediakan dalam pola ini, salin kode dari file App.js di ios-todo-app GitHub repositori aws-amplify-react-native-. Kemudian, integrasikan kode contoh ke lingkungan lokal Anda. Kode contoh yang disediakan dalam file App.js repositori melakukan hal berikut: • Menunjukkan formulir untuk membuat ToDo Item dengan bidang Judul dan Deskripsi • Menampilkan daftar item yang harus dilakukan (Judul dan Deskripsi) • Memposting dan mengambil data dengan menggunakan metode `aws-amplify`	Pengembang aplikasi

Sumber daya terkait

- AWS Amplify
- Amazon Cognito
- AWS AppSync
- Amazon DynamoDB
- Bereaksi (Dokumentasi React)

Mengelola penghuni di beberapa produk SaaS pada satu bidang kontrol

Ramanna Avancha, Kishan Kavala, Anusha Mandava, dan Jenifer Pascal, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara mengelola siklus hidup penyewa di beberapa produk perangkat lunak sebagai layanan (SaaS) pada satu bidang kontrol di AWS Cloud. Arsitektur referensi yang disediakan dapat membantu organisasi mengurangi implementasi fitur bersama yang berlebihan di seluruh produk SaaS masing-masing dan memberikan efisiensi tata kelola dalam skala besar.

Perusahaan besar dapat memiliki beberapa produk SaaS di berbagai unit bisnis. Produk-produk ini sering perlu disediakan untuk digunakan oleh penyewa eksternal pada tingkat langganan yang berbeda. Tanpa solusi penyewa umum, administrator TI harus menghabiskan waktu mengelola fitur yang tidak terdiferensiasi di beberapa SaaS APIs, alih-alih berfokus pada pengembangan fitur produk inti.

Solusi penyewa umum yang disediakan dalam pola ini dapat membantu memusatkan pengelolaan banyak fitur produk SaaS bersama organisasi, termasuk yang berikut:

- Keamanan
- Penyediaan penyewa
- Penyimpanan data penyewa
- Komunikasi penyewa
- Manajemen produk
- Pencatatan dan pemantauan metrik

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Pengetahuan tentang Amazon Cognito atau penyedia identitas pihak ketiga (iDP)
- Pengetahuan tentang Amazon API Gateway
- Pengetahuan tentang AWS Lambda
- Pengetahuan tentang Amazon DynamoDB
- Pengetahuan tentang AWS Identity and Access Management (IAM)

- Pengetahuan tentang AWS Step Functions
- Pengetahuan tentang AWS CloudTrail dan Amazon CloudWatch
- Pengetahuan tentang pustaka dan kode Python
- Pengetahuan tentang SaaS APIs, termasuk berbagai jenis pengguna (organisasi, penyewa, administrator, dan pengguna aplikasi), model berlangganan, dan model isolasi penyewa
- Pengetahuan tentang persyaratan SaaS multi-produk organisasi Anda dan langganan multi-penyewa

Batasan

- Integrasi antara solusi penyewa umum dan produk SaaS individu tidak tercakup dalam pola ini.
- Pola ini menerapkan layanan Amazon Cognito hanya dalam satu Wilayah AWS.

Arsitektur

Tumpukan teknologi target

- Amazon API Gateway
- Amazon Cognito
- AWS CloudTrail
- Amazon CloudWatch
- Amazon DynamoDB
- IAM
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Notification Service (Amazon SNS)
- Fungsi AWS Step

Arsitektur target

Diagram berikut menunjukkan contoh alur kerja untuk mengelola siklus hidup penyewa di beberapa produk SaaS pada satu bidang kontrol di AWS Cloud.

Diagram menunjukkan alur kerja berikut:

1. Pengguna AWS memulai penyediaan penyewa, penyediaan produk, atau tindakan terkait administrasi dengan melakukan panggilan ke titik akhir API Gateway.
2. Pengguna diautentikasi oleh token akses yang diambil dari kumpulan pengguna Amazon Cognito, atau IDP lain.
3. Tugas penyediaan atau administrasi individu dijalankan oleh fungsi Lambda yang terintegrasi dengan titik akhir API Gateway API.
4. Administrasi APIs untuk solusi penyewa umum (untuk penyewa, produk, dan pengguna) mengumpulkan semua parameter input, header, dan token yang diperlukan. Kemudian, administrasi APIs memanggil fungsi Lambda terkait.
5. Izin IAM untuk administrasi APIs dan fungsi Lambda divalidasi oleh layanan IAM.
6. Fungsi Lambda menyimpan dan mengambil data dari katalog (untuk penyewa, produk, dan pengguna) di DynamoDB dan Amazon S3.
7. Setelah izin divalidasi, alur kerja AWS Step Functions dipanggil untuk melakukan tugas tertentu. Contoh dalam diagram menunjukkan alur kerja penyediaan penyewa.
8. Tugas alur kerja AWS Step Functions individual dijalankan dalam alur kerja yang telah ditentukan (state machine).
9. Setiap data penting yang diperlukan untuk menjalankan fungsi Lambda yang terkait dengan setiap tugas alur kerja diambil dari DynamoDB atau Amazon S3. Sumber daya AWS lainnya mungkin perlu disediakan dengan menggunakan templat CloudFormation AWS.
10. Jika diperlukan, alur kerja mengirimkan permintaan untuk menyediakan sumber daya AWS tambahan untuk produk SaaS tertentu ke akun AWS produk tersebut.
11. Ketika permintaan berhasil atau gagal, alur kerja menerbitkan pembaruan status sebagai pesan ke topik Amazon SNS.
12. Amazon SNS berlangganan topik Amazon SNS alur kerja Step Functions.
13. Amazon SNS kemudian mengirimkan pembaruan status alur kerja kembali ke pengguna AWS.
14. Log dari setiap tindakan layanan AWS, termasuk jejak audit panggilan API, dikirim ke CloudWatch. Aturan dan alarm khusus dapat dikonfigurasi CloudWatch untuk setiap kasus penggunaan.
15. Log diarsipkan dalam bucket Amazon S3 untuk tujuan audit.

Otomatisasi dan skala

Pola ini menggunakan CloudFormation template untuk membantu mengotomatiskan penerapan solusi penyewa umum. Template juga dapat membantu Anda dengan cepat meningkatkan atau menurunkan sumber daya terkait.

Untuk informasi selengkapnya, lihat Bekerja dengan CloudFormation templat AWS di Panduan CloudFormation Pengguna AWS.

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.
- AWS CloudTrail membantu Anda mengaudit tata kelola, kepatuhan, dan risiko operasional akun AWS Anda.
- Amazon CloudWatch membantu Anda memantau metrik sumber daya AWS Anda dan aplikasi yang Anda jalankan di AWS secara real time.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi AWS Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis.

Praktik terbaik

Solusi dalam pola ini menggunakan pesawat kontrol tunggal untuk mengelola orientasi beberapa penyewa dan untuk menyediakan akses ke beberapa produk SaaS. Bidang kontrol membantu pengguna administratif mengelola empat bidang khusus fitur lainnya:

- Pesawat keamanan
- Bidang alur kerja
- Pesawat komunikasi
- Pesawat penebangan dan pemantauan

Epik

Konfigurasikan pesawat keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan persyaratan untuk platform SaaS multi-penyewa Anda.	Tetapkan persyaratan terperinci untuk hal-hal berikut: • Penyewa • Pengguna • Peran • Produk SaaS • Langganan • Pertukaran profil	Arsitek cloud, administrator sistem AWS
Siapkan layanan Amazon Cognito.	Ikuti petunjuk di Memulai Amazon Cognito di Panduan Pengembang Amazon Cognito.	Arsitek awan
Konfigurasikan kebijakan IAM yang diperlukan.	Buat kebijakan IAM yang diperlukan untuk kasus penggunaan Anda. Kemudian, petakan kebijakan ke peran IAM di Amazon Cognito. Untuk informasi selengkap nya, lihat Mengelola akses menggunakan kebijakan dan kontrol akses berbasis peran	Administrator cloud, arsitek Cloud, keamanan AWS IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di Panduan Pengembang Amazon Cognito.	
Konfigurasikan izin API yang diperlukan.	Siapkan izin akses API Gateway dengan menggunakan peran dan kebijakan IAM, serta otorisasi Lambda. Untuk petunjuknya, lihat bagian berikut dari Panduan Pengembang Amazon API Gateway: • Kontrol akses ke API dengan izin IAM • Gunakan otorisasi API Gateway Lambda	Administrator cloud, arsitek Cloud

Konfigurasikan bidang data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat katalog data yang diperlukan.	1. Buat tabel DynamoDB untuk menyimpan data untuk katalog pengguna. Pastikan Anda menyertakan atribut dan peran pengguna. Selain itu, pastikan Anda melakukan pemodelan data pada tabel katalog untuk mempertahankan atribut yang diperlukan dan opsional untuk setiap pengguna dan peran.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buat tabel DynamoDB untuk menyimpan data untuk katalog produk. Pastikan Anda memodelkan kasus penggunaan khusus untuk produk SaaS Anda. 3. Buat tabel DynamoDB untuk menyimpan data untuk katalog penyewa. Pastikan Anda menyiapkan model berlangganan untuk penyewa, produk, dan lisensi untuk langganan dan tag multi-SaaS. Untuk informasi selengkapnya, lihat Menyiapkan DynamoDB di Panduan Pengembang Amazon DynamoDB.	

Konfigurasikan bidang kontrol

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda dan API Gateway APIs untuk menjalankan tugas bidang kontrol yang diperlukan.	Buat fungsi Lambda terpisah dan API Gateway APIs untuk menambahkan, menghapus, dan mengelola hal-hal berikut: • Pengguna • Penyewa • Produk	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkap nya, lihat Menggunakan AWS Lambda dengan Amazon API Gateway di Panduan Pengembang AWS Lambda.	

Konfigurasikan bidang alur kerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi tugas yang harus dijalankan oleh alur kerja AWS Step Functions.	Identifikasi dan dokumenta sikan persyaratan alur kerja AWS Step Functions terperinci untuk hal-hal berikut: • Pengguna • Penyewa • Produk **Important** Pastikan bahwa pemangku kepenting an utama menyetujui persyaratan.	Pemilik aplikasi
Buat alur kerja AWS Step Functions yang diperlukan.	1. Buat alur kerja yang diperlukan untuk pengguna, penyewa, dan produk di AWS Step Functions. Untuk informasi selengkapnya, lihat Panduan Pengembang AWS Step Functions.	Pengembang aplikasi, Membangun prospek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Identifikasi mekanisme coba lagi dan penangana n kesalahan. Untuk informasi selengkapnya, lihat Menangani kesalahan , mencoba ulang, dan menambahkan peringata n ke Mesin Status Fungsi Langkah di Blog AWS. 3. Menerapkan langkah-l angkah alur kerja dengan menggunakan fungsi Lambda. Untuk petunjukn ya, lihat Membuat mesin status Step Functions yang menggunakan Lambda di Panduan Pengembang AWS Step Functions. 4. Integrasikan layanan eksternal apa pun dengan AWS Step Functions sesuai kebutuhan. 5. Pertahankan status setiap alur kerja dalam tabel DynamoDB dan komunikas ikan setiap status alur kerja menggunakan Amazon SNS.	

Konfigurasikan bidang komunikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat topik Amazon SNS.	Buat topik Amazon SNS untuk menerima pemberitahuan tentang hal berikut: • Status alur kerja • Kesalahan • Percobaan ulang Untuk informasi selengkap nya, lihat Membuat topik SNS di Panduan Pengembang Amazon SNS.	Pemilik aplikasi, arsitek Cloud
Berlangganan titik akhir untuk setiap topik Amazon SNS.	Untuk menerima pesan yang dipublikasikan ke topik Amazon SNS, Anda harus berlangganan titik akhir untuk setiap topik. Untuk informasi selengkap nya, lihat Berlangganan topik Amazon SNS di Panduan Pengembang Amazon SNS.	Pengembang aplikasi, arsitek Cloud

Konfigurasikan bidang logging dan monitoring

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan logging untuk setiap komponen solusi penyewa umum.	Aktifkan logging di tingkat komponen untuk setiap sumber daya dalam solusi	Pengembang aplikasi, administrator sistem AWS, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	penyewa umum yang Anda buat. Untuk petunjuk, lihat yang berikut ini: • Bagaimana cara mengaktifkan CloudWatch Log untuk memecahkan masalah API Gateway REST API atau WebSocket API saya? (Pusat Pengetahuan AWS) • Logging menggunakan CloudWatch Log (AWS Step Functions Developer Guide) • Pencatatan fungsi AWS Lambda dengan Python (Panduan Pengembang AWS Lambda) • Pencatatan dan pemantauan di Amazon Cognito (Panduan Pengembang Amazon Cognito) • Pemantauan dengan Amazon CloudWatch (Panduan Pengembang Amazon DynamoDB) **Note** Anda dapat mengkonsolidasikan log untuk setiap sumber daya ke	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akun logging terpusat dengan menggunakan kebijakan IAM. Untuk informasi selengkapnya, lihat Pencatatan terpusat dan pagar keamanan beberapa akun.	

Menyediakan dan menyebarkan solusi penyewa umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation template.	Otomatiskan penerapan dan pemeliharaan solusi penyewa umum penuh dan semua komponennya dengan menggunakan templat. CloudFormation Untuk informasi selengkapnya, lihat Panduan CloudFormation Pengguna AWS.	Pengembang aplikasi, DevOps insinyur, CloudFormation pengembang

Sumber daya terkait

- Kontrol akses ke REST API menggunakan kumpulan pengguna Amazon Cognito sebagai otorisasi (Panduan Pengembang Amazon API Gateway)
- Gunakan otorisasi API Gateway Lambda (Panduan Pengembang Amazon API Gateway)
- Kumpulan pengguna Amazon Cognito (Panduan Pengembang Amazon Cognito)
- CloudWatch Konsol lintas wilayah lintas akun (CloudWatch Panduan Pengguna Amazon)

Konsolidasikan pembuatan URL presigned Amazon S3 dan unduhan objek dengan menggunakan titik akhir yang terkait dengan alamat IP statis

Song Jin, Eunhye Jo, dan Jun Soung Lee, Amazon Web Services

Ringkasan

Pola ini menyederhanakan akses ke Amazon Simple Storage Service (Amazon S3) dengan membuat aman, custom presigned untuk unduhan objek. URLs Solusinya menyediakan titik akhir tunggal dengan domain unik dan alamat IP statis. Ini disesuaikan untuk pelanggan yang memerlukan konsolidasi titik akhir API dan Amazon S3 di bawah domain terpadu dengan alamat IP statis. Kasus penggunaan melibatkan pengguna yang mengikuti kebijakan firewall IP dan domain allowlist, membatasi akses API ke domain dan alamat IP tertentu.

Arsitektur menggunakan kunci Layanan AWS, termasuk AWS Global Accelerator, Amazon API Gateway AWS Lambda, Application Load Balancer AWS PrivateLink, dan Amazon S3. Desain ini memusatkan API untuk menghasilkan presigned URLs dan endpoint Amazon S3 di bawah satu domain, ditautkan ke akselerator dengan dua alamat IP statis. Akibatnya, pengguna dapat dengan mudah meminta presigned URLs dan men-download objek Amazon S3 melalui endpoint domain terpadu dengan alamat IP statis.

Arsitektur ini sangat bermanfaat bagi pelanggan dengan kebijakan atau persyaratan kepatuhan yang ketat, seperti yang ada di sektor publik, medis, dan keuangan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Zona yang dihosting publik untuk nama domain kustom Anda
- Domain yang diimpor AWS Certificate Manager (ACM) sesuai Wilayah AWS pilihan Anda

Batasan

- Nama bucket Amazon S3 harus cocok dengan nama domain titik akhir. Persyaratan ini untuk memastikan bahwa titik akhir Amazon S3 dapat dilayani melalui titik akhir API tunggal.
- Nama domain kustom yang digunakan dalam API Gateway harus sejajar dengan nama domain dari titik akhir API tunggal.

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan arsitektur target dan alur kerja untuk pola ini.

Diagram menggambarkan konsep dan alur kerja berikut:

1. Pengguna memulai permintaan untuk menghasilkan URL presigned dengan menggunakan endpoint kustom yang disajikan melalui AWS Global Accelerator, menggunakan nama domain kustom dan alamat IP terkait.
2. Fungsi Lambda menghasilkan URL yang telah ditetapkan sebelumnya, menunjuk ke titik akhir kustom. Ini merespons dengan pengalihan 301 yang berisi URL presigned yang dihasilkan. Melalui URL presigned yang dialihkan, pengguna mengunduh objek secara otomatis dengan menggunakan titik akhir kustom yang disajikan melalui Global Accelerator.

Komponen arsitektur keseluruhan untuk pembuatan URL presigned dan alur kerja pengunduhan objek adalah sebagai berikut:

- Penyediaan alamat IP statis oleh Global Accelerator.
- Pendaftaran alias akselerator sebagai catatan A ke zona host publik Amazon Route 53 dengan nama domain khusus.
- Pembuatan bucket Amazon S3 dengan nama bucket yang cocok dengan nama domain kustom terdaftar.
- Pembuatan titik akhir VPC untuk API Gateway dan layanan Amazon S3.
- Konfigurasi Application Load Balancer yang menghadap ke internal untuk terhubung ke Global Accelerator.
- Penetapan nama domain kustom untuk API Gateway dengan sertifikat ACM terlampir.
- Penerapan API Gateway pribadi yang terintegrasi dengan fungsi Lambda.
- Fungsi Lambda dilengkapi dengan peran AWS Identity and Access Management (IAM) yang dilampirkan (dengan GetObjectizin).

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- Application Load Balancer mendistribusikan lalu lintas aplikasi yang masuk ke beberapa target, seperti instans Amazon Elastic Compute Cloud (Amazon EC2), di beberapa Availability Zone.
- AWS Certificate Manager (ACM) membantu Anda membuat, menyimpan, dan memperbarui sertifikat dan kunci SSL/TLS X.509 publik dan pribadi yang melindungi situs web dan aplikasi Anda. AWS
- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS Global Acceleratoradalah layanan global yang mendukung titik akhir dalam banyak Wilayah AWS. Anda dapat membuat akselerator yang mengarahkan lalu lintas ke titik akhir optimal melalui jaringan AWS global. Hal ini meningkatkan ketersediaan dan performa aplikasi internet Anda yang digunakan oleh khalayak global.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS PrivateLinkmembantu Anda membuat koneksi pribadi searah dari cloud pribadi virtual Anda (VPCs) ke layanan di luar VPC.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Anda dapat menerapkan pola ini dengan menggunakan AWS CDK atau Terraform berdasarkan preferensi Anda. Bagian Epik berisi instruksi untuk kedua metode penerapan. Kode untuk pola ini tersedia di GitHub repositori berikut:

- AWS CDK— s3- -cdk presignedurl-staticips-endpoint-with
- Terraform — s3 - -terraform presignedurl-staticips-endpoint-with

Praktik terbaik

- Untuk meningkatkan keamanan di lingkungan produksi, sangat penting untuk menerapkan mekanisme otorisasi, seperti Amazon Cognito, untuk membatasi akses ke `PresignedUrl` API generasi.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Siapkan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan nama domain.	Tentukan nama domain publik untuk titik akhir Amazon S3 terpadu. Nama domain juga digunakan sebagai nama bucket Amazon S3.	Administrator AWS, Administrator jaringan
Membuat zona yang di-hosting publik.	Buat zona yang dihosting publik di Amazon Route 53. Nama domainnya harus sesuai dengan nama domain yang digunakan di API Gateway.	Administrator AWS, Administrator jaringan
Siapkan sertifikat SSL.	Gunakan AWS Certificate Manager (ACM) untuk	Administrator AWS, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	meminta atau mengimpor sertifikat SSL untuk domain aplikasi web Anda.	

Terapkan pola dengan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan lingkungan pengembangan Terraform.	Untuk mengatur lingkungan pengembangan, lakukan hal berikut: 1. Instal Terraform versi 1.0 atau yang lebih baru. 2. Kloning repositori GitHub s3- presignedurl-staticips-endpoint-with -terraform dengan menjalankan perintah berikut di jendela terminal: `git clone https://github.com/aws-samples/s3-presignedurl-staticips-endpoint-with-terraform.git`	Administrator AWS, Administrator cloud
Memodifikasi `.tfvars` dan **`provider.tf`** file.	1. Buat **`.tfvars`** file yang diperlukan — Buat `apg.tfvars` file di direktori `1.vpc_alb_ga` dan `2.apigw_s3_lambda` direktori. File-file ini akan berisi nilai	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	variabel khusus lingkungan Anda.
	• Untuk `1.vpc_alb_ga/apg.tfvars`, buat file dengan template berikut:
	`hcl`
	`environment_name = "test"`
	`domain = "your-domain.com"`
	`s3_bucket_prefix = "your-bucket-prefix"`
	`aws_region = "ap-northeast-2" # Change to your preferred region`
	• Untuk `2.apigw_s3_lambda/apg.tfvars`, buat file dengan template berikut (Anda harus memperbarui `vpc_id` setelah menyelesaikan langkah 1):
	`hcl`
	`vpc_id = "vpc-xxxxxxxxx" # Update this with the VPC ID output from step 1`
	`environment_name = "test"`
	`domain = "your-domain.com"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	s3_bucket_prefix = "your-bucket-prefix" aws_region = "ap-northeast-2" # Change to your preferred region	

	2. Ubah **provider.tf** file — Di folder `1.vpc_alb_ga` dan `2.apigw_s3_lambda` folder, edit `provider.tf` file agar sesuai dengan AWS konfigurasi lokal Anda:
	`hcl`
	`provider "aws" {`
	`region = "ap-northeast-2" # Change to your preferred region`
	`profile = "apg" # Change to your AWS profile name`
	`}`
	Perhatikan hal berikut:
	• Ganti nilai placeholder dalam `.tfvars` file dengan nilai aktual Anda.
	• Setelah menyelesaikan penerapan Terraform (`1.vpc_alb_ga`) pertama, perbarui `vpc_id` nilainya dengan ID VPC `2.apigw_s`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`3_lambda/apg.tfvars` dari output. • Pastikan bahwa nama AWS profil Anda dalam `provider.tf` file cocok dengan profil yang ada di `~/.aws/credentials` file Anda. • Nilai domain harus sesuai dengan domain zona host Route 53 Anda. • `s3_bucket_prefix` Gabungan dengan domain akan membentuk nama bucket Amazon S3 Anda (misalnya ,`your-bucket-prefix.your-domain.com`). • Jika Anda berubah`aws_region` , Anda mungkin juga perlu memperbarui zona ketersediaan hardcode dalam `main.tf` file agar sesuai dengan pilihan Anda. Wilayah AWS	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Penyediaan sumber daya jaringan.	Untuk menyediakan sumber daya jaringan, jalankan perintah berikut:
	`cd ./2.vpc_alb_ga`
	`terraform init`
	`terraform plan --var-file=apg.tfvars`
	`terraform apply --var-file=apg.tfvars`
	Selama eksekusi `apply` perintah, ketik ya saat diminta.
		Administrator AWS, Administrator cloud

Penyediaan API Gateway, Amazon S3, dan Lambda.	Untuk menyediakan sumber daya jaringan, gunakan perintah berikut:
	`cd ./2.apigw_s3_lambda`
	`terraform init`
	`terraform plan --var-file=apg.tfvars`
	`terraform apply --var-file=apg.tfvars`
		Administrator AWS, Administrator cloud

Terapkan pola dengan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur lingkungan AWS CDK pengembangan.	Untuk mengatur lingkungan pengembangan, lakukan hal berikut: 1. Instal AWS CDK. 2. Kloning repositori GitHub s3- presignedurl-staticips-	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	endpoint-with -cdk dengan menjalankan perintah berikut di jendela terminal: `git clone https://github.com/aws-samples/s3-presignedurl-staticips-endpoint-with-cdk.git`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan domain dalam `config/index.ts` file.	Untuk mengedit opsi variabel konstan, gunakan perintah berikut: `export const options = { certificateArn: '{arn of the acm which created before}', dnsAttr: { zoneName: '{public hosted zone name}', hostedZoneId: 'hosted zone Id', }, domainNamePrefix: '{Prefix for the domain}', presignPath: 'presign', objectsPath: 'objects', };` Dalam perintah, ganti setiap placeholder dengan informasi Anda sendiri: • Ganti `'{arn of the acm which created before}'` dengan sertifikat ACM Anda Nama Sumber Daya Amazon (ARN). • Ganti `'{public hosted zone name}'` dengan nama domain Anda.	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Ganti `'hosted zone Id'` dengan ID zona yang dihosting Route 53 Anda. • Ganti `'{Prefix for the domain}'` dengan awalan domain yang ingin Anda gunakan. • Secara opsional, modifikasi `'presign'` dan `'objects'` jalur jika diperlukan untuk kasus penggunaan spesifik Anda.	

Menyebarkan tumpukan.	Untuk menyebarkan dua tumpukan, satu untuk virtual private cloud (VPC) dan satu lagi untuk aplikasi, gunakan perintah berikut:
	`$ npm install`
	`$ cdk synth`
	`$ cdk deploy --all`
		Administrator AWS, Administrator cloud

Uji polanya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi alamat IP titik akhir.	Untuk memverifikasi bahwa domain untuk pola ini memiliki alamat IP statis, gunakan perintah berikut: `nslookup ${s3-bucket-prefix}.${domain}`	Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file pengujian yang nantinya dapat Anda unduh.	Unggah file pengujian ke `'/objects'` folder di bucket Amazon S3.	Administrator AWS, Administrator cloud
Memanggil API untuk menghasilkan URL presigned.	Untuk menghasilkan URL yang telah ditetapkan sebelumnya, panggil URL dari browser atau klien API (misalnya, Tukang Pos) menggunakan format berikut: `https://${s3-bucket-prefix}.${domain}/presign/objects/${uploaded-filename}` Ganti nilai placeholder di `${s3-bucket-prefix}` dan `${domain}` dengan nilai yang Anda tetapkan pada langkah sebelumnya.	Pemilik aplikasi
Periksa hasilnya.	Hasil yang diharapkan adalah Anda harus menerima kode status pengalihan 301 (Dipindahkan Secara Permanen). Respons ini akan berisi URL yang telah ditetapkan sebelumnya, yang akan secara otomatis memulai pengunduhan file pengujian Anda.	Insinyur uji

Bersihkan dengan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hancurkan sumber daya API Gateway, Amazon S3, dan Lambda.	Untuk menghapus sumber daya, gunakan perintah berikut:
	`cd ./2.apigw_s3_lambda`
	`terraform init`
	`terraform plan --destroy --var-file=apg.tfvars`
	`terraform destroy --var-file=apg.tfvars`
		Administrator AWS, Administrator cloud

Hancurkan sumber daya jaringan.	Untuk menghapus sumber daya jaringan, gunakan perintah berikut:
	`cd ./1.vpc_alb_ga`
	`terraform init`
	`terraform plan --destroy --var-file=apg.tfvars`
	`terraform destroy --var-file=apg.tfvars`
		Administrator AWS, Administrator cloud

Bersihkan dengan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hancurkan tumpukan.	Untuk menghancurkan tumpukan VPC dan aplikasi, gunakan perintah berikut: `$ cdk destroy --all`	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kosongkan dan hapus ember Amazon S3.	Kosongkan dan hapus objek bucket Amazon S3 dan log bucket Amazon S3 yang tidak dihapus secara default.
	
	Nama bucket Amazon S3 adalah `${s3-bucket-prefix}.${domain}` dan. `${s3-bucket-prefix}.${domain}-logs`
	
	Jika Anda lebih suka menggunakan AWS Command Line Interface (AWS CLI) untuk menghapus bucket, gunakan perintah berikut:
	
	`$ aws s3 rm s3://${s3-bucket-prefix}.${domain} --recursive`
	`$ aws s3 rb s3://${s3-bucket-prefix}.${domain} --force`
	`$ aws s3 rm s3://${s3-bucket-prefix}.${domain}-logs --recursive`
	`$ aws s3 rb s3://${s3-bucket-prefix}.${domain}-logs --force`
	
	Ganti `${s3-bucket-prefix}` dan `${domain}` dengan nilai yang Anda tetapkan pada langkah sebelumnya. , /p>
		Administrator AWS, Administrator cloud

Sumber daya terkait

AWS Blog

- Mengakses Amazon API Gateway melalui alamat IP statis yang disediakan oleh AWS Global Accelerator
- Hasilkan URL presigned dalam modular AWS CDK untuk JavaScript
- Hosting Situs Web Statis HTTPS Internal dengan ALB, S3, dan PrivateLink

Membuat EventBridge koneksi Amazon lintas akun di suatu organisasi

Sam Wilson dan Robert Stone, Amazon Web Services

Ringkasan

Sistem terdistribusi besar menggunakan Amazon EventBridge untuk mengkomunikasikan perubahan status antara berbagai akun Amazon Web Services (AWS) dalam suatu AWS Organizations organisasi. Namun, EventBridge umumnya hanya dapat menargetkan endpoint atau konsumen dalam hal yang sama Akun AWS. Pengecualian adalah bus acara di akun yang berbeda. Bus acara itu adalah target yang valid. Untuk mengkonsumsi acara dari bus acara di akun lain, acara harus didorong dari bus acara akun sumber ke bus acara akun tujuan. Untuk menghindari tantangan saat mengelola peristiwa kritis di seluruh aplikasi yang berbeda Akun AWS, gunakan pendekatan yang direkomendasikan yang disajikan dalam pola ini.

Pola ini menggambarkan bagaimana menerapkan arsitektur berbasis peristiwa dengan EventBridge yang melibatkan banyak Akun AWS dalam suatu organisasi. AWS Organizations Pola menggunakan AWS Cloud Development Kit (AWS CDK) Toolkit dan AWS CloudFormation.

EventBridge menawarkan bus acara tanpa server yang membantu Anda menerima, memfilter, mengubah, merutekan, dan mengirimkan acara. Komponen penting dari arsitektur berbasis peristiwa, EventBridge mendukung pemisahan antara produsen pesan dan konsumen pesan tersebut. Dalam satu akun, ini lurus ke depan. Struktur multi-akun memerlukan pertimbangan tambahan untuk acara di bus acara dalam satu akun untuk dikonsumsi di akun lain dalam organisasi yang sama.

Untuk informasi tentang pertimbangan khusus akun bagi produsen dan konsumen, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Sebuah AWS Organizations organisasi dengan setidaknya dua yang terkait Akun AWS
- Peran AWS Identity and Access Management (IAM) di keduanya Akun AWS yang memungkinkan Anda menyediakan infrastruktur di keduanya Akun AWS dengan menggunakan AWS CloudFormation
- Git diinstal secara lokal
- AWS Command Line Interface (AWS CLI) dipasang secara lokal
- AWS CDK diinstal secara lokal dan bootstrap di keduanya Akun AWS

Versi produk

Pola ini telah dibangun dan diuji dengan menggunakan alat dan versi berikut:

- AWS CDK Toolkit 2.126.0
- Node.js 18.19.0
- npm 10.2.3
- Python 3.12

Pola ini harus bekerja dengan versi AWS CDK v2 atau npm apa pun. Node.js versi 13.0.0 hingga 13.6.0 tidak kompatibel dengan. AWS CDK

Arsitektur

Arsitektur target

Diagram berikut menunjukkan alur kerja arsitektur untuk mendorong peristiwa dari satu akun dan menggunakannya di akun lain.

Alur kerja berisi langkah-langkah berikut:

1. AWS Lambda Fungsi Produser di akun Sumber menempatkan acara di bus EventBridge acara akun.
2. EventBridge Aturan lintas akun merutekan acara ke bus EventBridge acara di akun Tujuan.
3. Bus EventBridge acara di akun Tujuan memiliki aturan Lambda target yang memanggil fungsi Lambda Konsumen.

Praktik terbaik adalah menggunakan Dead Letter Queue (DLQ) untuk menangani pemanggilan fungsi Lambda Konsumen yang gagal. Namun, DLQ dihilangkan dari solusi ini untuk kejelasan. Untuk mempelajari lebih lanjut tentang cara menerapkan DLQ dalam alur kerja Anda dan meningkatkan kemampuan alur kerja Anda untuk pulih dari kegagalan, lihat posting blog Menerapkan pola penanganan AWS Lambda kesalahan.

Otomatisasi dan skala

AWS CDK secara otomatis menyediakan arsitektur yang diperlukan. EventBridge dapat menskalakan hingga ribuan catatan per detik tergantung pada Wilayah AWS. Untuk informasi selengkapnya, lihat dokumentasi EventBridge kuota Amazon.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode. Pola ini menggunakan AWS CDK Toolkit, kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan AWS CDK aplikasi.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.

Alat lainnya

- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.
- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub cross-account-eventbridge-in-organization.

Praktik terbaik

Untuk praktik terbaik saat bekerja sama EventBridge, lihat sumber daya berikut:

- Praktik terbaik untuk pola EventBridge acara Amazon
- Praktik terbaik saat mendefinisikan aturan di Amazon EventBridge

Epik

Siapkan lingkungan AWS CDK penyebaran lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kredenal lokal untuk akun Sumber dan akun Tujuan.	Tinjau Menyiapkan konfigurasi dan kredensil baru, dan gunakan metode otentikasi dan kredensi yang paling masuk akal bagi lingkungan Anda. **Important** Pastikan untuk mengonfigurasi AWS CLI untuk kedua akun Sumber dan otentikasi akun Tujuan. Petunjuk ini mengasumsikan bahwa Anda telah mengonfigurasi dua profil AWS secara lokal: `sourceAccount` dan`destinationAccount` .	Pengembang aplikasi

Bootstrap keduanya Akun AWS.	Untuk mem-bootstrap akun, jalankan perintah berikut:
	
	`cdk bootstrap --profile sourceAccount`
	`cdk bootstrap --profile destinationAccount`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning kode pola.	Untuk mengkloning repositori, jalankan perintah berikut:
	`git clone git@github.com:aws-samples/aws-cdk-examples.git`
	Kemudian, ubah direktori ke folder proyek yang baru dikloning:
	`cd aws-cdk-examples/python/cross-account-eventbridge-in-organization`
		Pengembang aplikasi

Terapkan ProducerStack ke akun Sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah `cdk.json` dengan detail akun AWS Organizations dan akun Anda.	Di folder root proyek, buat perubahan berikut ke`cdk.json`: • `organization_id` ID Organizations dari akun yang terlibat dalam penyebaran Anda • `event_bus_name -CrossAccount` , atau Anda dapat mengganti namanya	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `rules[].targets[].arn` Akun AWS ID akun konsumsi (Akun tujuan)	
Menyebarkan ProducerStack sumber daya.	Jalankan perintah berikut dari direktori root proyek: `cdk deploy ProducerStack --profile sourceAccount` Saat diminta, terima peran IAM baru dan izin terkait keamanan lainnya yang dibuat melalui. AWS CloudFormation	Pengembang aplikasi
Verifikasi bahwa ProducerStack sumber daya digunakan.	Untuk memverifikasi sumber daya, lakukan hal berikut: 1. Pada akun Konsol Manajemen AWS for Source, pilih CloudFormation. 2. Dari daftar tumpukan, pilih ProducerStack. 3. Pada tab Stack Info, verifikasi bahwa status tumpukan adalah`CREATE_COMPLETE` . Secara opsional, pada tab Sumber Daya, tinjau sumber daya yang dikonfigurasi.	Pengembang aplikasi

Menyebarkan ConsumerStack ke akun Tujuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan ConsumerStack sumber daya.	Jalankan perintah berikut dari direktori root proyek: `cdk deploy ConsumerStack --profile destinationAccount` Saat diminta, terima peran IAM baru dan izin terkait keamanan lainnya yang dibuat melalui. CloudFormation	Pengembang aplikasi
Verifikasi bahwa ConsumerStack sumber daya digunakan	1. Di konsol untuk akun Tujuan, pilih CloudFormation. 2. Dari daftar tumpukan, pilih ConsumerStack. 3. Pada tab Stack Info, verifikasi bahwa status tumpukan adalah`CREATE_COMPLETE` . Secara opsional, pada tab Sumber Daya, tinjau sumber daya yang dikonfigurasi.	Pengembang aplikasi

Menghasilkan dan mengkonsumsi acara

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memanggil fungsi Lambda Produser.	1. Di konsol untuk akun Sumber, pilih Lambda.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Dari daftar fungsi, pilih ProducerStack- ProducerLambda XXXX (XXXX mewakili urutan karakter yang dihasilkan secara otomatis oleh AWS CDK). 3. Pilih tab Uji. 4. Di bagian Test event, pilih Test. Konten area teks Event JSON dapat berupa JSON yang valid yang disediakan untuk fungsi Lambda sebagai payload. Dalam hal ini, JSON default yang disediakan sudah cukup. 5. Verifikasi bahwa fungsi Eksekusi: pesan berhasil muncul di spanduk hijau di atas bagian Acara uji.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa acara telah diterima.	1. Di konsol untuk akun Tujuan, pilih Lambda. 2. Dari daftar fungsi, pilih ConsumerStack-ConsumerLambda XXXX (XXXX mewakili urutan karakter yang dihasilkan secara otomatis oleh AWS CDK). 3. Pilih tab Pantau. 4. Di bagian Monitor, pilih Lihat CloudWatch log. 5. Di tab yang baru dibuka, pilih nama aliran log dari aliran log terbaru. 6. Verifikasi bahwa pernyataan log seperti berikut ini muncul: `[DEBUG] 2024-04-08T19:08:10.091Z 9c16844a-f9de-444d-b621-86afe64d4cc8 Event: {'version':'0', 'id':'0b9faa96-973f-8be2-ecf8-75e4f328b980', 'detail-type':'TestType', 'source':'Producer', 'account' :'XXXXXXXXXXXX', 'time':'2024-04-08T19:08:09Z',`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`'region':'us-east-1', 'resources': [], 'detail': {'key1':'value1', 'key2':'value2', 'key3':'value3'}}`	

Pembersihan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hancurkan ConsumerStack sumber daya.	Jika Anda menggunakan pola ini sebagai pengujian, bersihkan sumber daya yang digunakan untuk menghindari biaya tambahan. Jalankan perintah berikut dari direktori root proyek: `cdk destroy ConsumerStack --profile destinationAccount` Anda akan diminta untuk mengonfirmasi penghapusan tumpukan.	Pengembang aplikasi
Hancurkan ProducerStack sumber daya.	Jalankan perintah berikut dari direktori root proyek: `cdk destroy ProducerStack --profile sourceAccount`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda akan diminta untuk mengonfirmasi penghapusan tumpukan.	

Pemecahan masalah

Isu	Solusi
Tidak ada acara yang diterima di akun Tujuan.	1. Verifikasi bahwa ID Organizations yang diberikan sudah benar. 2. Verifikasi bahwa akun Sumber adalah bagian dari organisasi yang disediakan. 3. Verifikasi bahwa aturan bus acara di peta akun Sumber ke informasi yang benar di akun Tujuan.
Memanggil fungsi Lambda dari konsol mengembalikan kesalahan berikut: `User: arn:aws:iam::123456789012:user/XXXXX is not authorized to perform: lambda:Invoke`	Hubungi Akun AWS administrator Anda untuk menerima izin `lambda:Invoke` tindakan yang sesuai pada fungsi `ProducerStack-ProducerLambdaXXXX` Lambda.

Sumber daya terkait

Referensi

- AWS Organizations Panduan Pengguna
- Pola EventBridge acara Amazon
- Aturan di Amazon EventBridge

Tutorial dan video

- Tutorial: Membuat dan mengkonfigurasi organisasi

- AWS re: invent 2023 - Pola berbasis peristiwa tingkat lanjut dengan Amazon (01-R) EventBridge COM3

Informasi tambahan

Aturan produsen

Di akun Sumber, bus EventBridge acara dibuat untuk menerima pesan dari produsen (seperti yang ditunjukkan di bagian Arsitektur). Aturan dengan izin IAM yang menyertainya dibuat di bus acara ini. Aturan menargetkan bus EventBridge acara di akun Tujuan berdasarkan `cdk.json` struktur berikut:

```
"rules": [
  {
    "id": "CrossAccount",
    "sources": ["Producer"],
    "detail_types": ["TestType"],
    "targets": [
      {
        "id": "ConsumerEventBus",
        "arn": "arn:aws:events:us-east-2:012345678901:event-bus/CrossAccount"
      }
    ]
  }
]
```

Untuk setiap bus acara konsumsi, pola acara dan bus acara target harus disertakan.

Pola Acara

Pola acara memfilter peristiwa mana yang akan diterapkan aturan ini. Untuk tujuan contoh ini, sumber acara dan catatan `detail_types` mengidentifikasi peristiwa mana yang akan dikirimkan dari bus acara akun Sumber ke bus acara akun Tujuan.

Target bus acara

Aturan ini menargetkan bus acara yang ada di akun lain. Lengkap `arn` (Nama Sumber Daya Amazon) diperlukan untuk mengidentifikasi bus peristiwa target secara unik, dan itu `id` adalah ID logis yang digunakan oleh. AWS CloudFormation Bus acara target tidak perlu benar-benar ada pada saat pembuatan aturan target.

Pertimbangan khusus akun tujuan

Di akun Tujuan, bus EventBridge acara dibuat untuk menerima pesan dari bus acara akun Sumber. Untuk mengizinkan acara dipublikasikan dari akun Sumber, Anda harus membuat kebijakan berbasis sumber daya:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowOrgToPutEvents",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "events:PutEvents",
    "Resource": "arn:aws:events:us-east-2:012345678901:event-bus/CrossAccount",
    "Condition": {
      "StringEquals": {
        "aws:PrincipalOrgID": "o-XXXXXXXXX"
      }
    }
  }]
}
```

Sangat penting untuk memberikan `events:PutEvents` izin, yang memungkinkan akun lain di organisasi yang sama untuk mempublikasikan acara ke bus acara ini. Menyetel `aws:PrincipalOrgId` sebagai ID organisasi memberikan izin yang diperlukan.

Pola acara

Anda dapat memodifikasi pola acara yang disertakan untuk memenuhi kasus penggunaan Anda:

```
rule = events.Rule(
    self,
    self.id + 'Rule' + rule_definition['id'],
    event_bus=event_bus,
    event_pattern=events.EventPattern(
        source=rule_definition['sources'],
        detail_type=rule_definition['detail_types'],
    )
)
```

Untuk mengurangi pemrosesan yang tidak perlu, pola acara harus menentukan bahwa hanya peristiwa yang akan diproses oleh akun Tujuan yang ditransmisikan ke bus acara akun Tujuan.

Kebijakan berbasis sumber daya

Contoh ini menggunakan ID organisasi untuk mengontrol akun mana yang diizinkan untuk menempatkan peristiwa di bus acara akun Tujuan. Pertimbangkan untuk menggunakan kebijakan yang lebih ketat, seperti menentukan akun Sumber.

EventBridge kuota

Ingatlah kuota berikut:

- 300 aturan per bus acara adalah kuota default. Ini dapat diperluas jika perlu, tetapi harus sesuai dengan sebagian besar kasus penggunaan.
- Lima target per aturan adalah maksimum yang diizinkan. Kami merekomendasikan bahwa arsitek aplikasi harus menggunakan aturan yang berbeda untuk setiap akun Destination untuk mendukung kontrol halus atas pola acara.

Mengirimkan data DynamoDB ke Amazon S3 menggunakan Kinesis Data Streams dan Firehose dengan AWS CDK

Shashank Shrivastava dan Daniel Matuki da Cunha, Amazon Web Services

Ringkasan

Pola ini menyediakan contoh kode dan aplikasi untuk mengirimkan catatan dari Amazon DynamoDB ke Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dengan menggunakan Amazon Kinesis Data Streams dan Amazon Data Firehose. Pendekatan pola menggunakan konstruksi AWS Cloud Development Kit (AWS CDK) L3 dan menyertakan contoh cara melakukan transformasi data AWS Lambda sebelum data dikirim ke bucket S3 target di Amazon Web Services ()AWS Cloud.

Kinesis Data Streams merekam modifikasi tingkat item dalam tabel DynamoDB dan mereplikasi mereka ke aliran data Kinesis yang diperlukan. Aplikasi Anda dapat mengakses Kinesis data stream dan melihat perubahan tingkat item dalam hampir waktu nyata. Kinesis Data Streams juga menyediakan akses ke layanan Amazon Kinesis lainnya, seperti Firehose dan Amazon Managed Service untuk Apache Flink. Ini berarti Anda dapat membangun aplikasi yang menyediakan dasbor waktu nyata, menghasilkan peringatan, menerapkan harga dan iklan dinamis, dan melakukan analisis data yang canggih.

Anda dapat menggunakan pola ini untuk kasus penggunaan integrasi data Anda. Misalnya, kendaraan transportasi atau peralatan industri dapat mengirim volume data yang tinggi ke tabel DynamoDB. Data ini kemudian dapat diubah dan disimpan di danau data yang dihosting di Amazon S3. Anda kemudian dapat menanyakan dan memproses data dan memprediksi potensi cacat dengan menggunakan layanan tanpa server seperti Amazon Athena, Amazon Redshift Spectrum, Amazon Rekognition, dan. AWS Glue

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat AWS CLI Memulai dengan AWS CLI dokumentasi.
- Node.js (18.x +) dan npm, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Mengunduh dan menginstal Node.js dan npm di `npm` dokumentasi.

- aws-cdk (2.x +), diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat AWS CDK Memulai dengan AWS CDK dokumentasi.
- Repositori GitHub aws-dynamodb-kinesisfirehose-s3-ingestion, dikloning dan dikonfigurasi pada mesin lokal Anda.
- Data sampel yang ada untuk tabel DynamoDB. Data harus menggunakan format berikut: `{"SourceDataId": {"S": "123"},"MessageData":{"S": "Hello World"}}`

Arsitektur

Diagram berikut menunjukkan contoh alur kerja untuk mengirimkan catatan dari DynamoDB ke Amazon S3 dengan menggunakan Kinesis Data Streams dan Firehose.

Diagram menunjukkan alur kerja berikut:

1. Data dicerna menggunakan Amazon API Gateway sebagai proxy untuk DynamoDB. Anda juga dapat menggunakan sumber lain untuk menyerap data ke DynamoDB.
2. Perubahan tingkat item dihasilkan dalam waktu nyaris real time di Kinesis Data Streams untuk pengiriman ke Amazon S3.
3. Kinesis Data Streams mengirimkan catatan ke Firehose untuk transformasi dan pengiriman.
4. Fungsi Lambda mengonversi catatan dari format rekaman DynamoDB ke format JSON, yang hanya berisi nama dan nilai atribut item rekaman.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS CDK Toolkit adalah kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan AWS CDK aplikasi Anda.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub aws-dynamodb-kinesisfirehose-s3-konsumsi.

Epik

Siapkan dan konfigurasikan kode sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal dependensi.	Di mesin lokal Anda, instal dependensi dari `package.json` file di `sample-application` direktori `pattern/aws-dynamodb-kinesisstreams-s3` dan dengan menjalankan perintah berikut:
	`cd <project_root>/pattern/aws-dynamodb-kinesisstreams-s3`
	`npm install && npm run build`
	`cd <project_root>/sample-application/`
	`npm install && npm run build`
		Pengembang aplikasi, AWS Umum

Hasilkan CloudFormation template.	1. Jalankan perintah `cd <project_root>/sample-application/` .	Pengembang aplikasi, AWS Umum, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Jalankan `cdk synth` perintah untuk menghasil kan CloudFormation template. 3. `AwsDynamodbKinesis firehoseS3Ingestio nStack.template.js on` Output disimpan dalam `cdk.out` direktori. 4. Gunakan AWS CDK atau Konsol Manajemen AWS untuk memproses template di CloudFormation.	

Menyebarkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa dan gunakan sumber daya.	1. Jalankan `cdk diff` perintah untuk mengident ifikasi jenis sumber daya yang dibuat oleh AWS CDK konstruksi. 2. Jalankan `cdk deploy` perintah untuk menyebark an sumber daya.	Pengembang aplikasi, AWS Umum, AWS DevOps

Menyerap data ke dalam tabel DynamoDB untuk menguji solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyerap data sampel Anda ke dalam tabel DynamoDB.	Kirim permintaan ke tabel DynamoDB Anda dengan	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	menjalankan perintah berikut di: AWS CLI
	
	`aws dynamodb put-item --table-name <your_table_name> --item '{"<table_partition_key>": {"S": "<partition_key_ID>"},"MessageData":{"S": "<data>"}}'`
	
	contoh:
	
	`aws dynamodb put-item --table-name SourceData_table --item '{"SourceDataId": {"S": "123"},"MessageData":{"S": "Hello World"}}'`
	
	Secara default, `put-item` tidak mengembalikan nilai apa pun sebagai output jika operasi berhasil. Jika operasi gagal, ia mengembalikan kesalahan. Data disimpan di DynamoDB dan kemudian dikirim ke Kinesis Data Streams dan Firehose.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Anda menggunakan pendekatan yang berbeda untuk menambahkan data ke dalam tabel DynamoDB. Untuk informasi selengkapnya, lihat Memuat data ke dalam tabel dalam dokumentasi DynamoDB.	
Verifikasi bahwa objek baru dibuat di bucket S3.	Masuk ke Konsol Manajemen AWS dan pantau bucket S3 untuk memverifikasi bahwa objek baru dibuat dengan data yang Anda kirim. Untuk informasi selengkapnya, lihat GetObjectdi dokumentasi Amazon S3.	Pengembang aplikasi, AWS Umum

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Jalankan `cdk destroy` perintah untuk menghapus semua sumber daya yang digunakan oleh pola ini.	Pengembang aplikasi, AWS Umum

Sumber daya terkait

- s3- static-site-stack .ts (repositori) GitHub
- aws-apigateway-dynamodb modul (GitHub repositori)
- aws-kinesisstreams-kinesisfirehose-s3 modul (GitHub repositori)
- Ubah pengambilan data untuk DynamoDB Streams (dokumentasi DynamoDB)
- Menggunakan Kinesis Data Streams untuk menangkap perubahan pada DynamoDB (dokumentasi DynamoDB)

Menerapkan versi API berbasis jalur dengan menggunakan domain khusus di Amazon API Gateway

Corey Schnedl, Marcelo Barbosa, Mario Lopez Martinez, Anbazhagan Ponnuswamy, Gaurav Samudra, dan Abhilash Vinod, Amazon Web Services

Ringkasan

Pola ini menunjukkan bagaimana Anda dapat menggunakan fitur pemetaan API domain kustom untuk mengimplementasikan solusi versi API berbasis jalur untuk Amazon API Gateway.

Amazon API Gateway adalah layanan terkelola penuh yang dapat Anda gunakan untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan APIs pada skala apa pun. Dengan menggunakan fitur domain kustom layanan, Anda dapat membuat nama domain kustom yang lebih sederhana dengan lebih intuitif URLs yang dapat Anda berikan kepada pengguna API Anda. Anda dapat menggunakan pemetaan API untuk menghubungkan tahapan API ke nama domain kustom. Setelah Anda membuat nama domain dan mengonfigurasi catatan DNS, Anda menggunakan pemetaan API untuk mengirim lalu lintas ke Anda APIs melalui nama domain kustom Anda.

Setelah API tersedia untuk umum, konsumen menggunakannya. Seiring berkembangnya API publik, kontrak layanannya juga berkembang untuk mencerminkan fitur dan kemampuan baru. Namun, tidak bijaksana untuk mengubah atau menghapus fitur yang ada. Setiap perubahan yang melanggar dapat memengaruhi aplikasi konsumen dan merusaknya saat runtime. Pembuatan versi API penting untuk menghindari kerusakan kompatibilitas mundur dan melanggar kontrak.

Anda memerlukan strategi yang jelas untuk pembuatan versi API untuk membantu konsumen mengadopsinya. Pembuatan versi APIs dengan menggunakan berbasis jalur URLs adalah pendekatan yang paling mudah dan umum digunakan. Dalam jenis pembuatan versi ini, versi secara eksplisit didefinisikan sebagai bagian dari API. URIs Contoh berikut URLs menunjukkan bagaimana konsumen dapat menggunakan URI untuk menentukan versi API untuk permintaan mereka:

```
https://api.example.com/api/v1/orders
```

```
https://api.example.com/api/v2/orders
```

```
https://api.example.com/api/vX/orders
```

Pola ini menggunakan AWS Cloud Development Kit (AWS CDK) untuk membangun, menerapkan, dan menguji contoh implementasi solusi versi berbasis jalur yang dapat diskalakan untuk API Anda.

AWS CDK adalah kerangka pengembangan perangkat lunak open source untuk memodelkan dan menyediakan sumber daya aplikasi cloud Anda menggunakan bahasa pemrograman yang sudah dikenal.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Kepemilikan domain diperlukan untuk menggunakan contoh repositori pola ini dan menggunakan fungsionalitas domain khusus Amazon API Gateway. Anda dapat menggunakan Amazon Route 53 untuk membuat dan mengelola domain untuk organisasi Anda. Untuk informasi tentang cara mendaftar atau mentransfer domain dengan Route 53, lihat Mendaftarkan domain baru di dokumentasi Route 53.
- Sebelum menyiapkan nama domain khusus untuk API, Anda harus memiliki sertifikat SSL/TLS yang siap. AWS Certificate Manager
- Anda harus membuat atau memperbarui catatan sumber daya penyedia DNS Anda untuk memetakan ke titik akhir API Anda. Tanpa pemetaan seperti itu, permintaan API yang terikat untuk nama domain khusus tidak dapat mencapai API Gateway.

Keterbatasan

- Nama domain khusus tidak didukung untuk pribadi APIs.
- Nama domain khusus harus unik dalam Wilayah AWS semua Akun AWS.
- Untuk mengonfigurasi pemetaan API dengan beberapa level, Anda harus menggunakan nama domain kustom Regional dan menggunakan kebijakan keamanan TLS 1.2.
- Dalam pemetaan API, nama domain khusus dan dipetakan APIs harus sama. Akun AWS
- Pemetaan API harus hanya berisi huruf, angka, dan karakter berikut: `$-_.+!*'()/`
- Panjang maksimum jalur dalam pemetaan API adalah 300 karakter.
- Anda dapat memiliki 200 pemetaan API dengan beberapa level untuk setiap nama domain.
- Anda hanya dapat memetakan HTTP APIs ke nama domain kustom Regional dengan kebijakan keamanan TLS 1.2.
- Anda tidak dapat memetakan WebSocket APIs ke nama domain kustom yang sama dengan HTTP API atau REST API.

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Implementasi sampel ini digunakan AWS CDK dalam TypeScript versi 2.149.0.

Arsitektur

Diagram berikut menunjukkan alur kerja arsitektur.

Diagram ini menggambarkan hal sebagai berikut:

1. Pengguna API mengirimkan permintaan ke Amazon API Gateway dengan nama domain khusus.
2. API Gateway secara dinamis merutekan permintaan pengguna ke instance dan tahap API Gateway yang sesuai, berdasarkan jalur yang ditunjukkan dalam URL permintaan. Tabel berikut menunjukkan contoh bagaimana jalur berbasis URL yang berbeda dapat dirutekan ke tahapan tertentu untuk berbagai instance API Gateway.

API	Stage	Jalan	Titik akhir default
Perhitungan APIv1	API	apiv1	Diaktifkan
Perhitungan APIv2	API	apiv2	Diaktifkan
Perhitungan APIv X	API	APIVx	Diaktifkan

3. Instance API Gateway tujuan memproses permintaan dan mengembalikan hasilnya kepada pengguna.

Otomatisasi dan skala

Kami menyarankan Anda menggunakan AWS CloudFormation tumpukan terpisah untuk setiap versi API Anda. Dengan pendekatan ini, Anda dapat memiliki isolasi lengkap antara backend APIs yang dapat dirutekan oleh fitur pemetaan API domain kustom. Keuntungan dari pendekatan ini adalah bahwa versi berbeda dari API Anda dapat digunakan atau dihapus secara independen tanpa

menimbulkan risiko memodifikasi API lain. Pendekatan ini meningkatkan ketahanan melalui isolasi tumpukan. CloudFormation Selain itu, ini memberi Anda opsi back-end yang berbeda untuk API Anda seperti AWS Lambda, AWS Fargate, titik akhir HTTP, dan tindakan. Layanan AWS

Anda dapat menggunakan strategi percabangan Git, seperti Gitflow, dalam kombinasi dengan CloudFormation tumpukan terisolasi untuk mengelola kode sumber yang diterapkan ke versi API yang berbeda. Dengan menggunakan opsi ini, Anda dapat mempertahankan versi API yang berbeda tanpa perlu menduplikasi kode sumber untuk versi baru. Dengan Gitflow, Anda dapat menambahkan tag ke commit dalam repositori git Anda saat rilis dilakukan. Akibatnya, Anda memiliki snapshot lengkap dari kode sumber yang terkait dengan rilis tertentu. Karena pembaruan perlu dilakukan, Anda dapat memeriksa kode dari rilis tertentu, membuat pembaruan, dan kemudian menyebarkan kode sumber yang diperbarui ke CloudFormation tumpukan yang sejajar dengan versi utama yang sesuai. Pendekatan ini mengurangi risiko melanggar versi API lain karena setiap versi API memiliki kode sumber yang terisolasi dan digunakan untuk CloudFormation tumpukan terpisah.

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- AWS Certificate Manager (ACM) membantu Anda membuat, menyimpan, dan memperbarui sertifikat dan kunci SSL/TLS X.509 publik dan pribadi yang melindungi situs web dan aplikasi Anda. AWS
- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak open-source untuk mendefinisikan infrastruktur cloud Anda dalam kode dan menyediakannya. CloudFormation Implementasi sampel pola ini menggunakan AWS CDK in TypeScript. Bekerja dengan AWS CDK in TypeScript menggunakan alat yang sudah dikenal, termasuk Microsoft TypeScript compiler (`tsc`), Node.js, dan node package manager (`npm`). Jika mau, Anda dapat menggunakan Yarn meskipun contoh dalam pola ini digunakan`npm`. Modul yang terdiri dari AWS Construct Library didistribusikan melalui `npm` repositori, npmjs.org.
- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.
- AWS WAFadalah firewall aplikasi web yang membantu Anda memantau permintaan HTTP dan HTTPS yang diteruskan ke sumber daya aplikasi web Anda yang dilindungi.

Alat-alat lainnya

- Bruno adalah klien pengujian API open source yang ramah git-friendly.
- cdk-nag adalah utilitas open source yang memeriksa AWS CDK aplikasi untuk praktik terbaik dengan menggunakan paket aturan.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub path-based-versioning-with-api-gateway.

Praktik terbaik

- Gunakan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang kuat untuk mengotomatiskan pengujian dan penyebaran CloudFormation tumpukan Anda yang dibangun dengan. AWS CDK Untuk informasi lebih lanjut terkait dengan rekomendasi ini, lihat Panduan AWS DevOps Well-Architected.
- AWS WAF adalah firewall terkelola yang mudah diintegrasikan dengan layanan seperti Amazon API Gateway. Meskipun AWS WAF bukan merupakan komponen yang diperlukan agar pola pembuatan versi ini berfungsi, kami merekomendasikan sebagai praktik terbaik keamanan untuk disertakan AWS WAF dengan API Gateway.
- Dorong konsumen API untuk memutakhirkan secara berkala ke versi terbaru API Anda sehingga versi API yang lebih lama dapat dihentikan dan dihapus secara efisien.
- Sebelum menggunakan pendekatan ini dalam pengaturan produksi, terapkan firewall dan strategi otorisasi untuk API Anda.
- Terapkan akses ke pengelolaan sumber AWS daya Anda Akun AWS dengan menggunakan model akses hak istimewa paling sedikit.
- Untuk menerapkan praktik terbaik dan rekomendasi keamanan untuk aplikasi yang dibangun dengan AWS CDK, kami sarankan Anda menggunakan utilitas cdk-nag.

Epik

Persiapkan lingkungan lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori aplikasi sampel, jalankan perintah berikut: `git clone https://github.com/aws-samples/path-based-versioning-with-api-gateway`	Pengembang aplikasi
Arahkan ke repositori kloning.	Untuk menavigasi ke lokasi folder repositori kloning, jalankan perintah berikut: `cd api-gateway-custom-domain-versioning`	Pengembang aplikasi
Instal dependensi yang diperlukan.	Untuk menginstal dependensi yang diperlukan, jalankan perintah berikut: `npm install`	Pengembang aplikasi

Terapkan tumpukan CloudFormation perutean

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai penyebaran tumpukan routing.	Untuk memulai penyebaran tumpukan CloudFormation routing`CustomDomainRouterStack` ,	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	jalankan perintah berikut, ganti `example.com` dengan nama domain yang Anda miliki: `npx cdk deploy CustomDomainRouterStack --parameters PrerequisiteDomainName=example.com` **Note** Penyebaran tumpukan tidak akan berhasil sampai tugas validasi DNS domain berikut berhasil dilakukan.	

Memverifikasi kepemilikan domain

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi kepemilikan domain Anda.	Sertifikat akan tetap dalam status validasi Tertunda sampai Anda membuktikan kepemilikan domain terkait. Untuk membuktikan kepemilikan, tambahkan catatan CNAME ke zona yang dihosting yang terkait dengan domain. Untuk informasi selengkapnya, lihat validasi DNS dalam dokumentasi. AWS Certificate Manager	Pengembang aplikasi, administrator sistem AWS, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Menambahkan catatan yang sesuai memungkinkan `CustomDomainRouterStack` penerapan berhasil.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat catatan alias untuk menunjuk ke domain kustom API Gateway Anda.	Setelah sertifikat diterbitkan dan divalidasi dengan sukses, buat catatan DNS yang mengarah ke URL domain kustom Amazon API Gateway Anda. URL domain kustom dihasilkan secara unik oleh penyediaan domain kustom dan ditentukan sebagai parameter output. CloudFormation Berikut ini adalah contoh catatan: Kebijakan perutean: Perutean sederhana Nama rekam: `demo.api-gateway-custom-domain-versioning.example.com` Alias: Ya Jenis rekaman: Catatan DNS tipe “A” yang menunjuk ke sumber daya AWS Nilai: `d-xxxxxxxxxx.execute-api.xx-xxxx-x.amazonaws.com` TTL (detik): 300	Pengembang aplikasi, administrator sistem AWS, Administrator jaringan

Menerapkan CloudFormation tumpukan dan menjalankan API

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan `ApiStackV1` tumpukan.	Untuk menyebarkan `ApiStackV1` tumpukan, gunakan perintah berikut:
	`npm run deploy-v1`
	Kode CDK berikut menambahkan pemetaan API:
	`var apiMapping = new CfnApiMapping(this, "ApiMapping", { apiId: this.lambdaRestApi.restApiId, domainName: props.customDomainName.domainName, stage: "api", apiMappingKey: "api/v1", });`
		Pengembang aplikasi

Menyebarkan `ApiStackV2` tumpukan.	Untuk menyebarkan `ApiStackV2` tumpukan, gunakan perintah berikut: `npm run deploy-v2`	Pengembang aplikasi
Memanggil API.	Untuk menjalankan API dan menguji titik akhir API menggunakan Bruno, lihat petunjuk di Informasi tambahan.	Pengembang aplikasi

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Untuk menghancurkan sumber daya yang terkait dengan aplikasi sampel ini, gunakan perintah berikut: `npx cdk destroy --all` **Note** Pastikan Anda membersihkan semua catatan DNS Route 53 yang ditambahkan secara manual untuk proses verifikasi kepemilikan domain.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Penerapan `CustomDomainRouterStack` waktu habis karena sertifikat tidak dapat divalidasi.	Pastikan Anda menambahkan catatan CNAME validasi DNS yang tepat seperti yang dijelaskan dalam tugas sebelumnya. Sertifikat baru Anda mungkin terus menampilkan status validasi Tertunda hingga 30 menit setelah menambahkan catatan validasi DNS. Untuk informasi selengkapnya, lihat validasi DNS dalam dokumentasi. AWS Certificate Manager

Sumber daya terkait

- Menerapkan versi API Gateway berbasis header dengan CloudFront Amazon - Posting Blog Komputasi AWS ini menawarkan strategi pembuatan versi berbasis header sebagai alternatif strategi pembuatan versi berbasis jalur yang diuraikan dalam pola ini.
- AWS CDK Workshop — Lokakarya pengantar ini berfokus pada membangun dan menerapkan aplikasi AWS dengan menggunakan. AWS Cloud Development Kit (AWS CDK) Workshop ini mendukung Go, Python, dan. TypeScript

Informasi tambahan

Menguji API Anda dengan Bruno

Kami menyarankan Anda menggunakan Bruno, alat pengujian API open source, untuk memverifikasi bahwa routing berbasis jalur berfungsi dengan baik untuk aplikasi sampel. Pola ini menyediakan koleksi sampel untuk memfasilitasi pengujian API sampel Anda.

Untuk memanggil dan menguji API Anda, gunakan langkah-langkah berikut:

1. Instal Bruno.
2. Buka Bruno.
3. Dalam repositori kode pola ini, pilih Bruno/Sample-API - dan buka koleksi. Gateway-Custom-Domain-Versioning
4. Untuk melihat dropdown Environments di kanan atas antarmuka pengguna (UI), pilih permintaan apa pun dalam koleksi.
5. Di dropdown Environments, pilih Configure.
6. Ganti `REPLACE_ME_WITH_YOUR_DOMAIN` nilainya dengan domain kustom Anda.
7. Pilih Simpan, lalu tutup bagian Konfigurasi.
8. Untuk Sandbox Environment, verifikasi bahwa opsi Aktif dipilih.
9. Panggil API Anda dengan menggunakan tombol -> untuk permintaan yang dipilih.
10. Perhatikan bagaimana validasi (meneruskan nilai non-angka) ditangani di V1 dibandingkan dengan V2.

Untuk melihat tangkapan layar contoh pemanggilan API dan perbandingan validasi V1 dan V2, lihat Menguji API sampel Anda dalam **`README.md`** file di repositori kode pola ini.

Impor pustaka psycopg2 AWS Lambda untuk berinteraksi dengan database PostgreSQL Anda

Louis Hourcade, Amazon Web Services

Ringkasan

Psycopg adalah adaptor database PostgressQL untuk Python. Pengembang menggunakan `psycopg2` perpustakaan untuk menulis aplikasi Python yang berinteraksi dengan database PostgreSQL.

Di Amazon Web Services (AWS), pengembang juga menggunakan AWS Lambdauntuk menjalankan kode untuk aplikasi atau layanan backend. Lambda adalah layanan komputasi tanpa server yang digerakkan oleh peristiwa yang menjalankan kode tanpa perlu menyediakan atau mengelola server.

Secara default, saat Anda membuat fungsi baru yang menggunakan runtime Python yang didukung oleh Lambda, lingkungan runtime Lambda dibuat dari gambar dasar untuk Lambda yang disediakan oleh. AWS Pustaka, seperti `pandas` ataupsycopg2, tidak termasuk dalam gambar dasar. Untuk menggunakan perpustakaan, Anda perlu menggabungkannya dalam paket khusus dan melampirkannya ke Lambda.

Ada beberapa cara untuk menggabungkan dan melampirkan perpustakaan, termasuk yang berikut:

- Terapkan fungsi Lambda Anda dari arsip file.zip.
- Terapkan fungsi Lambda Anda dari gambar kontainer khusus.
- Buat layer Lambda, dan lampirkan ke fungsi Lambda Anda.

Pola ini menunjukkan dua opsi pertama.

Dengan paket penyebaran.zip, menambahkan `pandas` pustaka ke fungsi Lambda Anda relatif mudah. Buat folder di mesin Linux Anda, tambahkan skrip Lambda bersama dengan `pandas` perpustakaan dan dependensi perpustakaan ke folder, zip folder, dan sediakan sebagai sumber untuk fungsi Lambda Anda.

Meskipun menggunakan paket penyebaran.zip adalah praktik umum, pendekatan itu tidak berfungsi untuk perpustakaan. `psycopg2` Pola ini pertama-tama menunjukkan kesalahan yang Anda dapatkan jika Anda menggunakan paket penyebaran .zip untuk menambahkan `psycopg2` pustaka ke fungsi Lambda Anda. Pola kemudian menunjukkan cara menerapkan Lambda dari Dockerfile dan mengedit gambar Lambda untuk membuat perpustakaan berfungsi. `psycopg2`

Untuk informasi tentang tiga sumber daya yang diterapkan pola, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan izin yang cukup untuk menyebarkan AWS sumber daya yang digunakan oleh pola ini
- AWS Cloud Development Kit (AWS CDK) diinstal secara global dengan menjalankan `npm install -g aws-cdk`
- Klien Git
- Python
- Docker

Keterbatasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

- Versi runtime Python yang didukung oleh Lambda
- Psycopg2 versi 2.9.3
- Pandas versi 1.5.2

Arsitektur

Ikhtisar solusi

Untuk mengilustrasikan tantangan yang mungkin Anda hadapi saat menggunakan `psycopg2` pustaka di Lambda, pola tersebut menerapkan dua fungsi Lambda:

- Satu fungsi Lambda dengan runtime Python dibuat dari file.zip. `pandas`Pustaka `psycopg2` dan diinstal dalam paket penyebaran.zip ini dengan menggunakan pip.
- Satu fungsi Lambda dengan runtime Python dibuat dari Dockerfile. Dockerfile menginstal `psycopg2` dan `pandas` pustaka ke dalam gambar kontainer Lambda.

Fungsi Lambda pertama menginstal `pandas` perpustakaan dan dependensinya dalam file.zip, dan Lambda dapat menggunakan pustaka itu.

Fungsi Lambda kedua menunjukkan bahwa dengan membangun gambar kontainer untuk fungsi Lambda Anda, Anda dapat menjalankan dan `pandas` perpustakaan di Lambda. `psycopg2`

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- panda adalah alat open source berbasis Python untuk analisis dan manipulasi data.
- Psycopg adalah adaptor database PostgreSQL untuk bahasa Python yang dirancang untuk aplikasi multithreaded. Pola ini menggunakan Psycopg 2.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di impor-psycopg2 - - repositori pada. in-lambda-to-interact with-postgres-database GitHub

Praktik terbaik

Pola ini memberi Anda contoh kerja penggunaan AWS CDK untuk membuat fungsi Lambda dari Dockerfile. Jika Anda menggunakan kembali kode ini dalam aplikasi Anda, pastikan bahwa sumber daya yang digunakan memenuhi semua persyaratan keamanan. Gunakan alat seperti Checkov, yang memindai konfigurasi infrastruktur cloud untuk menemukan kesalahan konfigurasi sebelum infrastruktur diterapkan.

Epik

Kloning repositori dan konfigurasikan penerapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori.	Untuk mengkloning GitHub repositori pada mesin lokal Anda, jalankan perintah berikut:
	`git clone https://github.com/aws-samples/import-psycopg2-in-lambda-to-interact-with-postgres-database.git`
	`cd AWS-lambda-psycopg2`
		AWS Umum

Konfigurasikan penerapan Anda.	Edit `app.py` file dengan informasi tentang Anda Akun AWS:
	`aws_acccount = "AWS_ACCOUNT_ID"`
	`region = "AWS_REGION"`
	`# Select the CPU architecture you are using to build the image (ARM or X86)`
	`architecture = "ARM"`
		AWS Umum

Bootstrap akun AWS Anda dan terapkan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap Anda Akun AWS.	Jika Anda belum mem-bootstrap lingkungan AWS Anda,	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	jalankan perintah berikut dengan AWS kredensyal akun Anda: AWS `cdk bootstrap aws://<tooling-account-id>/<aws-region>`	
Menyebarkan kode.	Untuk menyebarkan AWS CDK aplikasi, jalankan perintah berikut: `cdk deploy AWSLambdaPyscopg2`	AWS Umum

Uji fungsi Lambda dari AWS Management Console

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji fungsi Lambda yang dibuat dari file.zip.	Untuk menguji fungsi Lambda yang dibuat dari file.zip, lakukan hal berikut: 1. Masuk ke konsol, dan buka konsol Lambda di. https://console.aws.amazon.com/lambda/ 2. Pilih fungsi `lambda-from-zip` Lambda. 3. Buat acara pengujian untuk memanggil fungsi. 4. Saat dipanggil, fungsi tersebut akan memunculk	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an kesalahan yang menyertakan pesan berikut: `"errorMessage": Unable to import module 'lambda_code': libpq.so.5: cannot open shared object, "stackTrace": [] "errorType": Runtime.ImportModuleError",` 5. Buka CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/. CloudWatch Log menunjukkan bahwa `pandas` pustaka telah berhasil diimpor tetapi impor `psycopg2` pustaka gagal. Karena Lambda tidak menemukan pustaka PostgreSQL yang diperlukan dalam gambar default, Lambda tidak dapat menggunakan pustaka. `psycopg2`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji fungsi Lambda yang dibuat dari Dockerfile.	Untuk menggunakan `psycopg2` pustaka dalam fungsi Lambda Anda, Anda harus mengedit Lambda Amazon Machine Image (AMI).
	
	Untuk menguji fungsi Lambda yang dibuat dari Dockerfile, lakukan hal berikut:
	
	1. Masuk ke konsol, dan buka konsol Lambda.
	2. Pilih fungsi `lambda-from-docker` Lambda.
	3. Buat acara pengujian untuk memanggil fungsi.
	4. Ketika dipanggil, fungsi harus berjalan dengan sukses.
	
	Kode berikut menunjukkan Dockerfile yang dibuat AWS CDK template:
	
	`# Start from lambda Python3.13 image`
	`FROM public.ecr.aws/lambda/python:3.13`
	
	`# Copy the lambda code, together with its requirements`
	`COPY lambda/requirements.txt ${LAMBDA_TASK_ROOT}`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`COPY lambda/lambda_code .py ${LAMBDA_TASK_ROOT }`
	
	`# Install postgresql- devel in your image RUN yum install -y gcc postgresql-devel`
	
	`# install the requireme nts for the Lambda code RUN pip3 install -r requirements.txt --target "${LAMBDA _TASK_ROOT}"`
	
	`# Command can be overwritten by providing a different command in the template directly. CMD ["lambda_code.hand ler"]`
	
	Dockerfile mengambil gambar AWS Lambda yang disediaka n untuk runtime Python dan menginstal postgresql-devel, yang berisi pustaka yang diperlukan untuk mengkompi lasi aplikasi yang berintera ksi langsung dengan server manajemen PostgreSQL. Dockerfile juga menginsta l `pandas` dan `psycopg2` perpustakaan, yang ditunjukk an dalam file. `requireme nts.txt`	

Sumber daya terkait

- AWS CDK dokumentasi
- AWS Lambda dokumentasi

Informasi tambahan

Dalam pola ini, AWS CDK template menyediakan AWS tumpukan dengan tiga sumber daya:

- Peran AWS Identity and Access Management (IAM) untuk fungsi Lambda.
- Fungsi Lambda dengan runtime Python. Fungsi ini digunakan dari paket `Constructs/lambda/lambda_deploy.zip` penyebaran.
- Fungsi Lambda dengan runtime Python. Fungsi ini digunakan dari Dockerfile di bawah folder `Constructs`

Skrip untuk kedua fungsi Lambda memeriksa apakah `psycopg2` pustaka `pandas` dan pustaka berhasil diimpor:

```
import pandas
print("pandas successfully imported")

import psycopg2
print("psycopg2 successfully imported")

def handler(event, context):
    """Function that checks whether psycopg2  and pandas are successfully imported or
 not"""
    return {"Status": "psycopg2 and pandas successfully imported"}
```

Paket `lambda_deploy.zip` penerapan dibangun dengan skrip `Constructs/lambda/build.sh` bash. Skrip ini membuat folder, menyalin skrip Lambda, menginstal `pandas` dan `psycopg2` pustaka, dan menghasilkan file.zip. Untuk membuat file.zip sendiri, jalankan skrip bash ini dan gunakan kembali tumpukan. AWS CDK

Dockerfile dimulai dengan image dasar yang AWS disediakan untuk Lambda dengan runtime Python. Dockerfile menginstal `pandas` dan `psycopg2` pustaka di atas gambar default.

Integrasikan Amazon API Gateway dengan Amazon SQS untuk menangani REST asinkron APIs

Natalia Colantonio Favero dan Gustavo Martim, Amazon Web Services

Ringkasan

Saat Anda menerapkan REST APIs, terkadang Anda perlu mengekspos antrian pesan yang dapat dipublikasikan oleh aplikasi klien. Misalnya, Anda mungkin memiliki masalah dengan latensi pihak ketiga APIs dan penundaan respons, atau Anda mungkin ingin menghindari waktu respons kueri database atau menghindari penskalaan server ketika ada sejumlah besar konkuren. APIs Dalam skenario ini, aplikasi klien yang mempublikasikan ke antrian hanya perlu mengetahui bahwa API menerima data—bukan apa yang terjadi setelah data diterima.

Pola ini membuat endpoint REST API dengan menggunakan Amazon API Gateway untuk mengirim pesan ke Amazon Simple Queue Service (Amazon SQS). Ini menciptakan easy-to-implement integrasi antara dua layanan yang menghindari akses langsung ke antrian SQS.

Prasyarat dan batasan

- AWS Akun aktif

Arsitektur

Diagram menggambarkan langkah-langkah ini:

1. Minta titik akhir POST REST API dengan menggunakan alat seperti Postman, API lain, atau teknologi lainnya.
2. API Gateway memposting pesan, yang diterima di badan permintaan, pada antrian.
3. Amazon SQS menerima pesan dan mengirimkan jawaban ke API Gateway dengan kode sukses atau gagal.

Alat

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.

Epik

Buat antrian SQS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat antrian.	Untuk membuat antrean SQS yang menerima pesan dari REST API: 1. Masuk ke Anda Akun AWS. 2. Buka konsol Amazon SQS di. https://console.aws.amazon.com/sqs/ 3. Pilih Buat antrean. 4. Pada halaman Create queue, pilih yang benar Wilayah AWS dari daftar dropdown Region. 5. Untuk Type, pertahankan pengaturan default (Standar). 6. Masukkan Nama untuk antrian Anda. 7. Simpan nilai default untuk semua pengaturan lainnya. 8. Pilih Buat antrean.	Pengembang aplikasi

Menyediakan akses ke Amazon SQS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM.	Peran IAM ini memberikan sumber daya API Gateway akses penuh ke Amazon SQS. 1. Buka konsol IAM di https://console.aws.amazon.com/iam/. 2. Dalam panel navigasi, pilih Roles (Peran), lalu Create role (Buat peran). 3. Untuk jenis entitas Tepercaya, pilih Layanan AWS. 4. Untuk kasus Penggunaan, pilih API Gateway dari daftar tarik-turun, lalu pilih Berikutnya, Berikutnya. 5. Untuk nama Peran, masukkan AWSGatewayRoleForSQS dan deskripsi opsional, lalu pilih Buat peran. 6. Di panel Peran, cari AWSGatewayRoleForSQS, dan pilih kotak centang. 7. Di bagian Kebijakan izin, pilih Tambahkan izin, Lampirkan kebijakan. 8. Cari Amazon SQSFull Access dan pilih. 9. Pilih Tambahkan izin.	Pengembang aplikasi, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	10. Di bagian Ringkasan AWSGateway RoleForSQS, salin Nomor Sumber Daya Amazon (ARN). Anda akan menggunakan ID ini di langkah selanjutnya.	

Buat REST API

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat REST API.	Ini adalah REST API tempat permintaan HTTP dikirim. 1. Buka konsol API Gateway di https://console.aws.amazon.com/apigateway/. 2. Di bagian REST API, pilih Build. 3. Untuk nama API, masukkan nama dan deskripsi opsional untuk API Anda, simpan semua pengaturan default lainnya, lalu pilih Buat API.	Pengembang aplikasi
Hubungkan API Gateway ke Amazon SQS.	Langkah ini memungkinkan pesan mengalir dari dalam badan permintaan HTTP ke Amazon SQS. 1. Di konsol API Gateway, pilih API yang Anda buat.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pada halaman Resources , di bagian Methods, pilih Create method. 3. Untuk jenis Metode, pilih POST. 4. Untuk jenis Integrasi, pilih Layanan AWS. 5. Untuk Wilayah AWS, pilih Wilayah tempat Anda membuat antrean SQS. 6. Untuk Layanan AWS, pilih Simple Queue Service (SQS). 7. Untuk metode HTTP, pilih POST. 8. Untuk tipe Action, pilih Use path override. 9. <name of SQS queue>Untuk Path override, masukkan/<AWS account ID>. 10. Untuk peran Eksekusi, tempel ARN dari peran yang Anda buat sebelumnya. 11. Pilih metode Buat.	

Uji REST API

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji REST API.	Jalankan tes untuk memeriksa konfigurasi yang hilang:
	1. Di konsol API Gateway, pilih REST API yang Anda buat.
	2. Di panel Resources, pilih metode POST.
	3. Pilih tab Uji. (Gunakan panah kanan jika tab tidak ditampilkan.)
	4. Untuk badan Permintaan, tempel kode JSON berikut:
	`{ "message": "lorem ipsum" }`
	5. Pilih Uji.
	Anda akan menerima kesalahan yang mirip dengan berikut ini:
	`<UnknownOperationException/>`
		Pengembang aplikasi

Ubah integrasi API untuk meneruskan permintaan dengan benar ke Amazon SQS.	Selesaikan konfigurasi untuk memperbaiki kesalahan integrasi:	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Di konsol API Gateway, pilih API yang Anda buat, lalu pilih POST. 2. Bagian Eksekusi Metode menunjukkan pemetaan visual antara API Gateway dan Amazon SQS. Dari bagian ini, pilih Permintaan integrasi, lalu pilih Edit. 3. Perluas bagian header HTTP, lalu pilih parameter Add request header. • Untuk Nama, tentukan Content-Type. • Untuk Dipetakan dari, masukkan x-www-form-urlencoded'aplikasi/ '. Pastikan untuk menyertakan tanda kutip tunggal. • Pilih kotak centang Caching. 4. Perluas bagian Templat pemetaan. • Pilih Tambahkan templat pemetaan. • Untuk jenis Konten, masukkan aplikasi/json. • Untuk badan Template, tempel kode ini: `Action=SendMessage&MessageBody=$input.body`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Pilih Simpan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji dan validasi pesan di Amazon SQS.	Jalankan tes untuk mengonfirmasi bahwa tes berhasil diselesaikan:
	1. Di konsol API Gateway, pilih REST API yang Anda buat.
	2. Di panel Resources, pilih metode POST.
	3. Pilih tab Uji. (Gunakan panah kanan jika tab tidak ditampilkan.)
	4. Untuk badan Permintaan, tempel kode JSON berikut:
	`{ "message": "lorem ipsum" }`
	5. Pilih Uji.
	6. Buka konsol Amazon SQS.
	7. Di panel navigasi, pilih Antrian, lalu pilih antrian Anda.
	8. Pilih Kirim dan terima pesan.
	9. Pilih Poll untuk pesan.
	10. Pilih Pesan. Ini harus menampilkan yang berikut:
	`Body { "message": "lorem ipsum" }`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji API Gateway dengan karakter khusus.	Jalankan pengujian yang menyertakan karakter khusus (seperti &) yang tidak dapat diterima dalam pesan:
	1. Di konsol API Gateway, pilih API Anda.
	2. Ulangi tes dari langkah sebelumnya dengan menggunakan kode JSON berikut:
	`{ "message": "lorem ipsum &" }`
	3. Pilih Uji.
	Anda akan menerima kesalahan seperti berikut:
	`{ "Error": { "Code": "AccessDenied", "Message": "Access to the resource https://sqs.us-east-2.amazonaws.com/976166761794/Apg2 is denied.", "Type": "Sender" }, "RequestId": "e83c9c67-bcf6-5e9a-91e9-c737094b17ab"`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>}</pre> Ini karena karakter khusus tidak didukung secara default di badan pesan. Pada langkah berikutnya, Anda akan mengonfigurasi API Gateway untuk mendukung karakter khusus. Untuk informasi selengkapnya tentang konversi jenis konten, lihat dokumentasi API Gateway.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ubah konfigurasi API untuk mendukung karakter khusus.	Sesuaikan konfigurasi untuk menerima karakter khusus dalam pesan:
	1. Di konsol API Gateway, pilih API yang Anda buat, lalu pilih POST.
	2. Pilih Permintaan integrasi, lalu pilih Edit.
	3. Ubah Penanganan Konten untuk Mengonversi ke teks.
	4. Di bagian Templat pemetaan:
	• Untuk jenis Konten, masukkan aplikasi/json.
	• Untuk badan Template, tentukan:
	`Action=SendMessage&MessageBody=$util.urlEncode($input.body)`
	• Pilih Simpan.
	5. Pilih tab Uji.
	6. Untuk badan Permintaan, masukkan kode JSON dari sebelumnya:
	`{ " message": "lorem ipsum &" }`
	7. Pilih Uji.
	8. Buka konsol Amazon SQS.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	9. Pilih antrian Anda, lalu pilih Kirim dan terima pesan, Poll untuk pesan, Pesan seperti sebelumnya. Pesan baru harus menyertakan karakter khusus.	

Menerapkan REST API

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan API.	Untuk menerapkan REST API: 1. Buka konsol API Gateway. 2. Pilih API Anda. 3. Pilih Deploy API. Untuk informasi selengkapnya tentang langkah ini, lihat dokumentasi API Gateway.	Pengembang aplikasi
Uji dengan alat eksternal.	Jalankan pengujian dengan alat eksternal untuk mengonfirmasi bahwa pesan berhasil diterima: 1. Buka alat seperti Tukang Pos, Insomnia, atau cURL. 2. Jalankan API Anda. 3. Buka konsol Amazon SQS. 4. Pilih antrian Anda. 5. Muat pesan untuk melihat pesan baru.	Pengembang aplikasi

Pembersihan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus API.	Di konsol API Gateway, pilih API yang Anda buat, lalu pilih Hapus.	Pengembang aplikasi
Hapus peran IAM.	Di konsol IAM, di panel Peran, pilih AWSGatewayRoleForSQS, lalu pilih Hapus.	Pengembang aplikasi
Hapus antrian SQS.	Di konsol Amazon SQS, di panel Antrian, pilih antrian SQS yang Anda buat, lalu pilih Hapus.	Pengembang aplikasi

Sumber daya terkait

- SQS- SendMessage (dokumentasi API Gateway)
- Konversi jenis konten di API Gateway (dokumentasi API Gateway)
- variabel $ util (dokumentasi API Gateway)
- Bagaimana cara mengintegrasikan API Gateway REST API dengan Amazon SQS dan mengatasi kesalahan umum? (AWS Re: artikel posting)

Memproses peristiwa secara asinkron dengan Amazon API Gateway dan AWS Lambda

Andrea Meroni, Mariem Kthiri, Nadim Majed, dan Michael Wallner, Amazon Web Services

Ringkasan

Amazon API Gateway adalah layanan terkelola penuh yang dapat digunakan pengembang untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan APIs pada skala apa pun. Ini menangani tugas-tugas yang terlibat dalam menerima dan memproses hingga ratusan ribu panggilan API bersamaan.

Kuota layanan penting API Gateway adalah batas waktu integrasi. Batas waktu adalah waktu maksimum di mana layanan backend harus mengembalikan respons sebelum REST API mengembalikan kesalahan. Batas keras 29 detik umumnya dapat diterima untuk beban kerja sinkron. Namun, batas itu merupakan tantangan bagi pengembang yang ingin menggunakan API Gateway dengan beban kerja asinkron.

Pola ini menunjukkan contoh arsitektur untuk memproses peristiwa secara asinkron menggunakan API Gateway dan. AWS Lambda Arsitektur mendukung menjalankan tugas pemrosesan dengan durasi hingga 15 menit, dan menggunakan REST API dasar sebagai antarmuka.

Projen digunakan untuk mengatur lingkungan pengembangan lokal dan untuk menyebarkan arsitektur contoh ke target Akun AWS, dalam kombinasi dengan AWS Cloud Development Kit (AWS CDK) Toolkit, Docker, dan Node.js. Projen secara otomatis menyiapkan lingkungan virtual Python dengan pra-komit dan alat yang digunakan untuk jaminan kualitas kode, pemindaian keamanan, dan pengujian unit. Untuk informasi selengkapnya, lihat bagian Alat.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Alat-alat berikut diinstal pada workstation Anda:
  - AWS Cloud Development Kit (AWS CDK) Toolkit versi 2.85.0
  - Docker versi 20.10.21
  - Node.js versi 18.13.0
  - Projen versi 0.71.111

- Python versi 3.9.16

Keterbatasan

- Runtime maksimum suatu pekerjaan dibatasi oleh runtime maksimum untuk fungsi Lambda (15 menit).
- Jumlah maksimum permintaan pekerjaan bersamaan dibatasi oleh konkurensi yang dicadangkan dari fungsi Lambda.

Arsitektur

Diagram berikut menunjukkan interaksi API pekerjaan dengan fungsi Lambda pemrosesan peristiwa dan penanganan kesalahan, dengan peristiwa yang disimpan dalam arsip peristiwa Amazon. EventBridge

Alur kerja tipikal mencakup langkah-langkah berikut:

1. Anda mengautentikasi terhadap AWS Identity and Access Management (IAM) dan mendapatkan kredensi keamanan.
2. Anda mengirim `POST` permintaan HTTP ke titik akhir API `/jobs` lowongan, yang menentukan parameter pekerjaan di badan permintaan.
3. API jobs, yang merupakan API API Gateway REST API, mengembalikan respons HTTP yang berisi pengenal pekerjaan kepada Anda.
4. API pekerjaan memanggil fungsi Lambda pemrosesan peristiwa secara asinkron.
5. Fungsi pemrosesan peristiwa memproses peristiwa, dan kemudian menempatkan hasil pekerjaan di tabel Amazon DynamoDB pekerjaan
6. Anda mengirim `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` pekerjaan, dengan pengenal pekerjaan dari langkah 3 sebagai. `{jobId}`
7. API pekerjaan menanyakan tabel `jobs` DynamoDB untuk mengambil hasil pekerjaan.
8. API pekerjaan mengembalikan respons HTTP yang berisi hasil pekerjaan.
9. Jika pemrosesan peristiwa gagal, fungsi pemrosesan peristiwa mengirimkan acara ke fungsi penanganan kesalahan.
10. Fungsi penanganan kesalahan menempatkan parameter pekerjaan dalam tabel `jobs` DynamoDB.

11 Anda dapat mengambil parameter pekerjaan dengan mengirimkan `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` lowongan.

12 Jika penanganan kesalahan gagal, fungsi penanganan kesalahan akan mengirimkan acara ke arsip EventBridge peristiwa.

Anda dapat memutar ulang acara yang diarsipkan dengan menggunakan. EventBridge

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, fungsi Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Alat-alat lainnya

- autopep8 secara otomatis memformat kode Python berdasarkan panduan gaya Proposal Peningkatan Python (PEP) 8.
- Bandit memindai kode Python untuk menemukan masalah keamanan umum.
- Commitizen adalah pemeriksa dan generator komit Git. `CHANGELOG`
- cfn-lint adalah linter AWS CloudFormation
- Checkov adalah alat analisis kode statis yang memeriksa infrastruktur sebagai kode (IAc) untuk kesalahan konfigurasi keamanan dan kepatuhan.
- jq adalah alat baris perintah untuk mengurai JSON.

- Postman adalah platform API.
- pre-commit adalah manajer Git Hooks.
- Projen adalah generator proyek.
- pytest adalah kerangka kerja Python untuk menulis tes kecil yang dapat dibaca.

Repositori kode

Contoh kode arsitektur ini dapat ditemukan di GitHub Asynchronous Event Processing dengan API Gateway dan repositori Lambda.

Praktik terbaik

- Contoh arsitektur ini tidak termasuk pemantauan infrastruktur yang diterapkan. Jika kasus penggunaan Anda memerlukan pemantauan, evaluasi penambahan Konstruksi Pemantauan CDK atau solusi pemantauan lainnya.
- Contoh arsitektur ini menggunakan izin IAM untuk mengontrol akses ke API pekerjaan. Siapa pun yang berwenang untuk berasumsi `JobsAPIInvokeRole` akan dapat memanggil API pekerjaan. Dengan demikian, mekanisme kontrol akses adalah biner. Jika kasus penggunaan Anda memerlukan model otorisasi yang lebih kompleks, evaluasi menggunakan mekanisme kontrol akses yang berbeda.
- Saat pengguna mengirim `POST` permintaan HTTP ke titik akhir API `/jobs` jobs, data input divalidasi pada dua tingkatan yang berbeda:
  - Amazon API Gateway bertanggung jawab atas validasi permintaan pertama.
  - Fungsi pemrosesan acara melakukan permintaan kedua.

    Tidak ada validasi yang dilakukan saat pengguna melakukan `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` lowongan. Jika kasus penggunaan Anda memerlukan validasi input tambahan dan peningkatan tingkat keamanan, evaluasi menggunakan AWS WAF untuk melindungi API Anda.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori secara lokal, jalankan perintah berikut: `git clone https://github.com/aws-samples/asynchronous-event-processing-api-gateway-lambda-cdk.git`	DevOps insinyur

Siapkan proyek.	Ubah direktori ke root repositori dan atur lingkungan virtual Python dan semua alat dengan menggunakan Projen:
	`cd asynchronous-event-processing-api-gateway-api-gateway-lambda-cdk`
	`npx projen`
		DevOps insinyur

Instal kait pra-komit.	Untuk memasang kait pra-komit, lakukan hal berikut:
	1. Aktifkan lingkungan virtual Python:
	`source .env/bin/activate`
	2. Pasang kait pra-komit:
	`pre-commit install`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`pre-commit install --hook-type commit-msg`	

Menyebarkan contoh arsitektur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap AWS CDK.	Untuk bootstrap AWS CDK di Anda Akun AWS, jalankan perintah berikut: `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen bootstrap`	AWS DevOps
Menyebarkan contoh arsitektur.	Untuk menerapkan arsitektur contoh di Anda Akun AWS, jalankan perintah berikut: `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen deploy`	AWS DevOps

Uji arsitekturnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal prasyarat tes.	Instal di workstation Anda AWS Command Line Interface (AWS CLI), Postman, dan jq. Menggunakan Postman untuk menguji arsitektur contoh ini disarankan tetapi tidak	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	wajib. Jika Anda memilih alat pengujian API alternatif, pastikan alat tersebut mendukung otentikasi AWS Signature Version 4, dan lihat titik akhir API terbuka yang dapat diperiksa dengan mengekspor REST API.	
Asumsikan `JobsAPIInvokeRole`.	Asumsikan `JobsAPIInvokeRole` yang dicetak sebagai output dari perintah deploy: `CREDENTIALS=$(AWS_PROFILE=$<YOUR_AWS_PROFILE> aws sts assume-role \ --no-cli-pager \ --role-arn $<JOBS_API_INVOKE_ROLE_ARN> \ --role-session-name JobsAPIInvoke) export AWS_ACCESS_KEY_ID=$(cat $CREDENTIALS \| jq '.Credentials''.AccessKeyId') export AWS_SECRET_ACCESS_KEY=$(cat $CREDENTIALS \| jq '.Credentials''.SecretAccessKey') export AWS_SESSION_TOKEN==$(cat $CREDENTIALS | jq '.Credentials''.SessionToken')`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Tukang Pos.	1. Untuk mengimpor koleksi Postman yang disertakan dalam repositori, ikuti petunjuk dalam dokumentasi Tukang Pos. 2. Atur `JobsAPI` variabel dengan nilai-nilai berikut: • `accessKey` - Nilai `Credentials.AccessKeyId` atribut dari `assume-role` perintah • `baseUrl`- Nilai `JobsApiJobsAPIEndpoint` output dari perintah penerapan, tanpa garis miring • `region`Nilai Wilayah AWS tempat Anda menerapkan arsitektur contoh • `seconds`- Nilai parameter input untuk pekerjaan contoh. Itu harus bilangan bulat positif • `secretKey` - Nilai `Credentials.SecretAccessKey` atribut dari `assume-role` perintah • `sessionToken` - Nilai `Credentials.Sessio`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`nToken` atribut dari `assume-role` perintah	
Uji contoh arsitektur.	Untuk menguji arsitektur contoh, kirim permintaan ke API pekerjaan. Untuk informasi lebih lanjut, lihat dokumentasi Tukang Pos.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Penghancuran dan pemindahan arsitektur contoh berikutnya gagal karena grup CloudWatch log Amazon Logs `/aws/apigateway/JobsAPIAccessLogs` sudah ada.	1. Jika perlu, ekspor data log Anda ke Amazon S3. 2. Hapus grup CloudWatch log Log`/aws/apigateway/JobsAPIAccessLogs` . 3. Menerapkan ulang contoh arsitektur.

Sumber daya terkait

- Template pemetaan API Gateway dan referensi variabel pencatatan akses
- Siapkan pemanggilan asinkron dari fungsi Lambda backend

Memproses peristiwa secara asinkron dengan Amazon API Gateway dan Amazon DynamoDB Streams

Andrea Meroni, Mariem Kthiri, Nadim Majed, Alessandro Trisolini, dan Michael Wallner, Amazon Web Services

Ringkasan

Amazon API Gateway adalah layanan terkelola penuh yang dapat digunakan pengembang untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan APIs pada skala apa pun. Ini menangani tugas yang terlibat dalam menerima dan memproses hingga ratusan ribu panggilan API bersamaan.

Kuota layanan penting API Gateway adalah batas waktu integrasi. Batas waktu adalah waktu maksimum di mana layanan backend harus mengembalikan respons sebelum REST API mengembalikan kesalahan. Batas keras 29 detik umumnya dapat diterima untuk beban kerja sinkron. Namun, batas itu merupakan tantangan bagi pengembang yang ingin menggunakan API Gateway dengan beban kerja asinkron.

Pola ini menunjukkan contoh arsitektur untuk memproses peristiwa secara asinkron menggunakan API Gateway, Amazon DynamoDB Streams, dan. AWS Lambda Arsitektur mendukung menjalankan tugas pemrosesan paralel dengan parameter input yang sama, dan menggunakan REST API dasar sebagai antarmuka. Dalam contoh ini, menggunakan Lambda sebagai backend membatasi durasi pekerjaan hingga 15 menit. Anda dapat menghindari batas ini dengan menggunakan layanan alternatif untuk memproses peristiwa yang masuk (misalnya, AWS Fargate).

Projen digunakan untuk mengatur lingkungan pengembangan lokal dan untuk menyebarkan arsitektur contoh ke target Akun AWS, dalam kombinasi dengan AWS Cloud Development Kit (AWS CDK) Toolkit, Docker dan Node.js. Projen secara otomatis menyiapkan lingkungan virtual Python dengan pra-komit dan alat yang digunakan untuk jaminan kualitas kode, pemindaian keamanan, dan pengujian unit. Untuk informasi selengkapnya, lihat bagian Alat.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Alat-alat berikut diinstal pada workstation Anda:

- AWS Cloud Development Kit (AWS CDK) Toolkit versi 2.85.0 atau yang lebih baru
- Docker versi 20.10.21 atau yang lebih baru
- Node.js versi 18 atau yang lebih baru
- Projen versi 0.71.111 atau yang lebih baru
- Python versi 3.9.16 atau yang lebih baru

Keterbatasan

- Jumlah maksimum pembaca yang disarankan untuk DynamoDB Streams adalah dua untuk menghindari pelambatan.
- Runtime maksimum suatu pekerjaan dibatasi oleh runtime maksimum untuk fungsi Lambda (15 menit).
- Jumlah maksimum permintaan pekerjaan bersamaan dibatasi oleh konkurensi yang dicadangkan dari fungsi Lambda.

Arsitektur

Arsitektur

Diagram berikut menunjukkan interaksi API pekerjaan dengan DynamoDB Streams dan fungsi Lambda pemrosesan peristiwa dan penanganan kesalahan, dengan peristiwa yang disimpan dalam arsip peristiwa Amazon. EventBridge

Alur kerja tipikal mencakup langkah-langkah berikut:

1. Anda mengautentikasi terhadap AWS Identity and Access Management (IAM) dan mendapatkan kredensi keamanan.
2. Anda mengirim `POST` permintaan HTTP ke titik akhir API `/jobs` lowongan, yang menentukan parameter pekerjaan di badan permintaan.
3. API pekerjaan mengembalikan respons HTTP yang berisi pengenal pekerjaan kepada Anda.
4. API pekerjaan menempatkan parameter pekerjaan di tabel `jobs_table` Amazon DynamoDB.
5. Tabel `jobs_table` DynamoDB aliran DynamoDB memanggil fungsi Lambda pemrosesan peristiwa.

6. Fungsi Lambda pemrosesan peristiwa memproses peristiwa dan kemudian menempatkan hasil pekerjaan di tabel DynamoDB. `jobs_table` Untuk membantu memastikan hasil yang konsisten, fungsi pemrosesan peristiwa menerapkan mekanisme penguncian yang optimis.
7. Anda mengirim `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` pekerjaan, dengan pengenal pekerjaan dari langkah 3 sebagai. `{jobId}`
8. API pekerjaan menanyakan tabel `jobs_table` DynamoDB untuk mengambil hasil pekerjaan.
9. API pekerjaan mengembalikan respons HTTP yang berisi hasil pekerjaan.
10. Jika pemrosesan peristiwa gagal, pemetaan sumber fungsi pemrosesan peristiwa akan mengirimkan peristiwa ke topik Simple Notification Service Amazon (Amazon SNS) yang menangani kesalahan.
11. Topik SNS penanganan kesalahan secara asinkron mendorong acara ke fungsi penanganan kesalahan.
12. Fungsi penanganan kesalahan menempatkan parameter pekerjaan dalam tabel `jobs_table` DynamoDB.

    Anda dapat mengambil parameter pekerjaan dengan mengirimkan `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` lowongan.
13. Jika penanganan kesalahan gagal, fungsi penanganan kesalahan akan mengirimkan acara ke arsip Amazon EventBridge .

    Anda dapat memutar ulang acara yang diarsipkan dengan menggunakan. EventBridge

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, fungsi AWS Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun AWS lainnya.

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

Alat lainnya

- autopep8 secara otomatis memformat kode Python berdasarkan panduan gaya Proposal Peningkatan Python (PEP) 8.
- Bandit memindai kode Python untuk menemukan masalah keamanan umum.
- Commitizen adalah pemeriksa dan generator komit Git. `CHANGELOG`
- cfn-lint adalah linter AWS CloudFormation
- Checkov adalah alat analisis kode statis yang memeriksa infrastruktur sebagai kode (IAc) untuk kesalahan konfigurasi keamanan dan kepatuhan.
- jq adalah alat baris perintah untuk mengurai JSON.
- Postman adalah platform API.
- pre-commit adalah manajer Git Hooks.
- Projen adalah generator proyek.
- pytest adalah kerangka kerja Python untuk menulis tes kecil yang dapat dibaca.

Repositori kode

Contoh kode arsitektur ini dapat ditemukan di GitHub Asynchronous Processing with API Gateway dan DynamoDB Streams repositori.

Praktik terbaik

- Contoh arsitektur ini tidak termasuk pemantauan infrastruktur yang diterapkan. Jika kasus penggunaan Anda memerlukan pemantauan, evaluasi penambahan Konstruksi Pemantauan CDK atau solusi pemantauan lainnya.
- Contoh arsitektur ini menggunakan izin IAM untuk mengontrol akses ke API pekerjaan. Siapa pun yang berwenang untuk berasumsi `JobsAPIInvokeRole` akan dapat memanggil API pekerjaan. Dengan demikian, mekanisme kontrol akses adalah biner. Jika kasus penggunaan

Anda memerlukan model otorisasi yang lebih kompleks, evaluasi menggunakan mekanisme kontrol akses yang berbeda.

- Saat pengguna mengirim `POST` permintaan HTTP ke titik akhir API `/jobs` jobs, data input divalidasi pada dua tingkatan yang berbeda:
  - API Gateway bertanggung jawab atas validasi permintaan pertama.
  - Fungsi pemrosesan acara melakukan permintaan kedua.

  Tidak ada validasi yang dilakukan saat pengguna melakukan `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` lowongan. Jika kasus penggunaan Anda memerlukan validasi input tambahan dan tingkat keamanan yang lebih tinggi, evaluasi penggunaan AWS WAF untuk melindungi API Anda.

- Untuk menghindari pembatasan, dokumentasi DynamoDB Streams mencegah pengguna membaca dengan lebih dari dua konsumen dari pecahan aliran yang sama. Untuk meningkatkan jumlah konsumen, sebaiknya gunakan Amazon Kinesis Data Streams.
- Penguncian optimis telah digunakan dalam contoh ini untuk memastikan pembaruan item yang konsisten dalam tabel `jobs_table` DynamoDB. Bergantung pada persyaratan kasus penggunaan, Anda mungkin perlu menerapkan mekanisme penguncian yang lebih andal, seperti penguncian pesimis.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori secara lokal, jalankan perintah berikut: `git clone https://github.com/aws-samples/asynchronous-event-processing-api-gateway-dynamodb-streams-cdk.git`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan proyek.	Ubah direktori ke root repositori, dan atur lingkungan virtual Python dan semua alat dengan menggunakan Projen:
	`cd asynchronous-event-processing-api-gateway-api-gateway-dynamodb-streams-cdk`
	`npx projen`
		DevOps insinyur

Pasang kait pra-komit.	Untuk memasang kait pra-komit, lakukan hal berikut:
	1. Aktifkan lingkungan virtual Python:
	`source .env/bin/activate`
	2. Pasang kait pra-komit:
	`pre-commit install`
	`pre-commit install --hook-type commit-msg`
		DevOps insinyur

Menyebarkan contoh arsitektur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap AWS CDK.	Untuk bootstrap AWS CDKdi Anda Akun AWS, jalankan perintah berikut:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`AWS_PROFILE=$YOUR_AWS_PROFILE npx projen bootstrap`	
Menyebarkan contoh arsitektur.	Untuk menerapkan arsitektur contoh di Anda Akun AWS, jalankan perintah berikut: `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen deploy`	AWS DevOps

Uji arsitekturnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal prasyarat tes.	Instal di workstation Anda AWS Command Line Interface (AWS CLI), Postman, dan jq. Menggunakan Postman untuk menguji arsitektur contoh ini disarankan tetapi tidak wajib. Jika Anda memilih alat pengujian API alternatif, pastikan alat tersebut mendukung autentikasi AWS Signature Version 4, dan lihat titik akhir API terbuka yang dapat diperiksa dengan mengekspor REST API.	DevOps insinyur
Asumsikan`JobsAPIInvokeRole`.	Asumsikan `JobsAPIInvokeRole` yang dicetak	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sebagai output dari `deploy` perintah: `CREDENTIALS=$(AWS_ PROFILE=$<YOUR_AWS _PROFILE> aws sts assume-role \ --no-cli-pager \ --role-arn $<JOBS_AP I_INVOKE_ROLE_ARN> \ --role-session-name JobsAPIInvoke) export AWS_ACCES S_KEY_ID=$(cat $CREDENTIALS \| jq '.Credentials''.Ac cessKeyId') export AWS_SECRE T_ACCESS_KEY=$(cat $CREDENTIALS \| jq '.Credentials''.Se cretAccessKey') export AWS_SESSI ON_TOKEN==$(cat $CREDENTIALS | jq '.Credentials''.Se ssionToken')`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan Tukang Pos.	• Untuk mengimpor koleksi Postman yang disertakan dalam repositori, ikuti petunjuk dalam dokumentasi Tukang Pos.
	• Atur `JobsAPI` variabel dengan nilai-nilai berikut:
	  • `accessKey` - Nilai `Credentials.AccessKeyId` atribut dari `assume-role` perintah.
	  • `baseUrl`- Nilai `JobsApiJobsAPIEndpoint` output dari `deploy` perintah, tanpa garis miring.
	  • `region`Nilai di Wilayah AWS mana Anda menggunakan arsitektur contoh.
	  • `seconds`- Nilai parameter input untuk pekerjaan contoh. Itu harus bilangan bulat positif.
	  • `secretKey` - Nilai `Credentials.SecretAccessKey` atribut dari `assume-role` perintah.
	  • `sessionToken` - Nilai `Credentials.SessionToken` atribut dari `assume-role` perintah.
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji contoh arsitektur.	Untuk menguji arsitektur contoh, kirim permintaan ke API pekerjaan. Untuk informasi lebih lanjut, lihat dokumentasi Tukang Pos.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Penghancuran dan pemindahan arsitektur contoh berikutnya gagal karena grup CloudWatch log Amazon Logs `/aws/apigateway/JobsAPIAccessLogs` sudah ada.	1. Jika perlu, ekspor data log Anda ke Amazon Simple Storage Service (Amazon S3). 2. Hapus grup CloudWatch log Log`/aws/apigateway/JobsAPIAccessLogs` . 3. Menerapkan ulang contoh arsitektur.

Sumber daya terkait

- Template pemetaan API Gateway dan referensi variabel pencatatan akses
- Ubah pengambilan data untuk DynamoDB Streams
- Penguncian optimis dengan nomor versi
- Menggunakan Kinesis Data Streams untuk menangkap perubahan ke DynamoDB

Memproses peristiwa secara asinkron dengan Amazon API Gateway, Amazon SQS, dan AWS Fargate

Andrea Meroni, Mariem Kthiri, Nadim Majed, Alessandro Trisolini, dan Michael Wallner, Amazon Web Services

Ringkasan

Amazon API Gateway adalah layanan terkelola penuh yang dapat digunakan pengembang untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan APIs pada skala apa pun. Ini menangani tugas-tugas yang terlibat dalam menerima dan memproses hingga ratusan ribu panggilan API bersamaan.

Kuota layanan penting API Gateway adalah batas waktu integrasi. Batas waktu adalah waktu maksimum di mana layanan backend harus mengembalikan respons sebelum REST API mengembalikan kesalahan. Batas keras 29 detik umumnya dapat diterima untuk beban kerja sinkron. Namun, batas itu merupakan tantangan bagi pengembang yang ingin menggunakan API Gateway dengan beban kerja asinkron.

Pola ini menunjukkan contoh arsitektur untuk memproses peristiwa secara asinkron menggunakan API Gateway, Amazon Simple Queue Service (Amazon SQS) dan. AWS Fargate Arsitektur mendukung menjalankan pekerjaan pemrosesan tanpa batasan durasi, dan menggunakan REST API dasar sebagai antarmuka.

Projen digunakan untuk mengatur lingkungan pengembangan lokal dan untuk menyebarkan arsitektur contoh ke target Akun AWS, dalam kombinasi dengan, Docker AWS Cloud Development Kit (AWS CDK), dan Node.js. Projen secara otomatis menyiapkan lingkungan virtual Python dengan pra-komit dan alat yang digunakan untuk jaminan kualitas kode, pemindaian keamanan, dan pengujian unit. Untuk informasi selengkapnya, lihat bagian Alat.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Alat-alat berikut diinstal pada workstation Anda:
  - AWS Cloud Development Kit (AWS CDK) Toolkit versi 2.85.0 atau yang lebih baru
  - Docker versi 20.10.21 atau yang lebih baru

- Node.js versi 18 atau yang lebih baru
- Projen versi 0.71.111 atau yang lebih baru
- Python versi 3.9.16 atau yang lebih baru

Batasan

- Pekerjaan bersamaan dibatasi hingga 500 tugas per menit, yang merupakan jumlah maksimum tugas yang dapat disediakan Fargate.

Arsitektur

Diagram berikut menunjukkan interaksi API jobs dengan tabel `jobs` Amazon DynamoDB, layanan Fargate pemrosesan peristiwa, dan fungsi penanganan kesalahan. AWS Lambda Acara disimpan dalam arsip EventBridge acara Amazon.

Alur kerja tipikal mencakup langkah-langkah berikut:

1. Anda mengautentikasi terhadap AWS Identity and Access Management (IAM) dan mendapatkan kredensyal keamanan.
2. Anda mengirim `POST` permintaan HTTP ke titik akhir API `/jobs` lowongan, yang menentukan parameter pekerjaan di badan permintaan.
3. API jobs, yang merupakan API API Gateway REST API, mengembalikan respons HTTP yang berisi pengenal pekerjaan kepada Anda.
4. API pekerjaan mengirimkan pesan ke antrean SQS.
5. Fargate menarik pesan dari antrian SQS, memproses acara, dan kemudian menempatkan hasil pekerjaan di tabel DynamoDB. `jobs`
6. Anda mengirim `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` pekerjaan, dengan pengenal pekerjaan dari langkah 3 sebagai. `{jobId}`
7. API pekerjaan menanyakan tabel `jobs` DynamoDB untuk mengambil hasil pekerjaan.
8. API pekerjaan mengembalikan respons HTTP yang berisi hasil pekerjaan.
9. Jika pemrosesan acara gagal, antrian SQS mengirimkan acara ke antrian huruf mati (DLQ).
10. Sebuah EventBridge peristiwa memulai fungsi penanganan kesalahan.
11. Fungsi penanganan kesalahan menempatkan parameter pekerjaan dalam tabel `jobs` DynamoDB.

12 Anda dapat mengambil parameter pekerjaan dengan mengirimkan `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` lowongan.

13 Jika penanganan kesalahan gagal, fungsi penanganan kesalahan akan mengirimkan acara ke arsip. EventBridge

Anda dapat memutar ulang acara yang diarsipkan dengan menggunakan. EventBridge

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Fargatemembantu Anda menjalankan container tanpa perlu mengelola server atau instans Amazon Elastic Compute Cloud EC2 (Amazon). Ini digunakan bersama dengan Amazon Elastic Container Service (Amazon ECS).
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, fungsi Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.

Alat-alat lainnya

- autopep8 secara otomatis memformat kode Python berdasarkan panduan gaya Proposal Peningkatan Python (PEP) 8.
- Bandit memindai kode Python untuk menemukan masalah keamanan umum.
- Commitizen adalah pemeriksa dan generator komit Git. `CHANGELOG`

- cfn-lint adalah linter AWS CloudFormation
- Checkov adalah alat analisis kode statis yang memeriksa infrastruktur sebagai kode (IAc) untuk kesalahan konfigurasi keamanan dan kepatuhan.
- jq adalah alat baris perintah untuk mengurai JSON.
- Postman adalah platform API.
- pre-commit adalah manajer Git Hooks.
- Projen adalah generator proyek.
- pytest adalah kerangka kerja Python untuk menulis tes kecil yang dapat dibaca.

Repositori kode

Contoh kode arsitektur ini dapat ditemukan di GitHub Asynchronous Processing dengan API Gateway dan SQS repositori.

Praktik terbaik

- Contoh arsitektur ini tidak termasuk pemantauan infrastruktur yang diterapkan. Jika kasus penggunaan Anda memerlukan pemantauan, evaluasi penambahan Konstruksi Pemantauan CDK atau solusi pemantauan lainnya.
- Contoh arsitektur ini menggunakan izin IAM untuk mengontrol akses ke API pekerjaan. Siapa pun yang berwenang untuk berasumsi `JobsAPIInvokeRole` akan dapat memanggil API pekerjaan. Dengan demikian, mekanisme kontrol akses adalah biner. Jika kasus penggunaan Anda memerlukan model otorisasi yang lebih kompleks, evaluasi menggunakan mekanisme kontrol akses yang berbeda.
- Saat pengguna mengirim `POST` permintaan HTTP ke titik akhir API `/jobs` jobs, data input divalidasi pada dua tingkatan yang berbeda:
  - API Gateway bertanggung jawab atas validasi permintaan pertama.
  - Fungsi pemrosesan acara melakukan permintaan kedua.

    Tidak ada validasi yang dilakukan saat pengguna melakukan `GET` permintaan HTTP ke titik akhir API `/jobs/{jobId}` lowongan. Jika kasus penggunaan Anda memerlukan validasi input tambahan dan tingkat keamanan yang lebih tinggi, evaluasi penggunaan AWS WAF untuk melindungi API Anda.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori secara lokal, jalankan perintah berikut: `git clone https://github.com/aws-samples/asynchronous-event-processing-api-gateway-sqs-cdk.git`	DevOps insinyur

Siapkan proyek.	Ubah direktori ke root repositori, dan atur lingkungan virtual Python dan semua alat dengan menggunakan Projen:
	`cd asynchronous-event-processing-api-gateway-api-gateway-sqs-cdk`
	`npx projen`
		DevOps insinyur

Instal kait pra-komit.	Untuk memasang kait pra-komit, lakukan hal berikut:
	1. Aktifkan lingkungan virtual Python:
	`source .env/bin/activate`
	2. Pasang kait pra-komit:
	`pre-commit install`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`pre-commit install --hook-type commit-msg`	

Menyebarkan contoh arsitektur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap AWS CDK.	Untuk bootstrap AWS CDKdi Anda Akun AWS, jalankan perintah berikut: `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen bootstrap`	AWS DevOps
Menyebarkan contoh arsitektur.	Untuk menerapkan arsitektur contoh di Anda Akun AWS, jalankan perintah berikut: `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen deploy`	AWS DevOps

Uji arsitekturnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal prasyarat tes.	Instal di workstation Anda AWS Command Line Interface (AWS CLI), Postman, dan jq. Menggunakan Postman untuk menguji arsitektur contoh ini disarankan tetapi tidak	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	wajib. Jika Anda memilih alat pengujian API alternatif, pastikan alat tersebut mendukung otentikasi AWS Signature Version 4, dan lihat titik akhir API terbuka yang dapat diperiksa dengan mengekspor REST API.	
Asumsikan `JobsAPIInvokeRole` .	Asumsikan `JobsAPIInvokeRole` yang dicetak sebagai output dari `deploy` perintah: `CREDENTIALS=$(AWS_PROFILE=$<YOUR_AWS_PROFILE> aws sts assume-role \ --no-cli-pager \ --role-arn $<JOBS_API_INVOKE_ROLE_ARN> \ --role-session-name JobsAPIInvoke) export AWS_ACCESS_KEY_ID=$(cat $CREDENTIALS \| jq '.Credentials''.AccessKeyId') export AWS_SECRET_ACCESS_KEY=$(cat $CREDENTIALS \| jq '.Credentials''.SecretAccessKey') export AWS_SESSION_TOKEN==$(cat $CREDENTIALS | jq '.Credentials''.SessionToken')`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan Tukang Pos.	• Untuk mengimpor koleksi Postman yang disertakan dalam repositori, ikuti petunjuk dalam dokumentasi Tukang Pos.
	• Atur `JobsAPI` variabel dengan nilai-nilai berikut:
	  • `accessKey` - Nilai `Credentials.AccessKeyId` atribut dari `assume-role` perintah.
	  • `baseUrl`- Nilai `JobsApiJobsAPIEndpoint` output dari `deploy` perintah, tanpa garis miring.
	  • `region`Nilai di Wilayah AWS mana Anda menggunakan arsitektur contoh.
	  • `seconds`- Nilai parameter input untuk pekerjaan contoh. Itu harus bilangan bulat positif.
	  • `secretKey` - Nilai `Credentials.SecretAccessKey` atribut dari `assume-role` perintah.
	  • `sessionToken` - Nilai `Credentials.SessionToken` atribut dari `assume-role` perintah.
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji contoh arsitektur.	Untuk menguji arsitektur contoh, kirim permintaan ke API pekerjaan. Untuk informasi lebih lanjut, lihat dokumentasi Tukang Pos.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Penghancuran dan pemindahan arsitektur contoh berikutnya gagal karena grup CloudWatch log Amazon Logs `/aws/apigateway/JobsAPIAccessLogs` sudah ada.	1. Jika perlu, ekspor data log Anda ke Amazon Simple Storage Service (Amazon S3). 2. Hapus grup CloudWatch log Log `/aws/apigateway/JobsAPIAccessLogs` . 3. Menerapkan ulang contoh arsitektur.
Penghancuran dan pemindahan arsitektur contoh berikutnya gagal karena grup CloudWatch log Log `/aws/ecs/EventProcessingServiceLogs` sudah ada.	1. Jika perlu, ekspor data log Anda ke Amazon S3. 2. Hapus grup CloudWatch log Log `/aws/ecs/EventProcessingServiceLogs.` 3. Menerapkan ulang contoh arsitektur.

Sumber daya terkait

- Template pemetaan API Gateway dan referensi variabel pencatatan akses
- Bagaimana cara mengintegrasikan API Gateway REST API dengan Amazon SQS dan mengatasi kesalahan umum?

Jalankan tugas AWS Systems Manager Automation secara sinkron dari AWS Step Functions

Elie El khoury, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara berintegrasi AWS Step Functions dengan AWS Systems Manager. Ini menggunakan integrasi layanan AWS SDK untuk memanggil Systems Manager startAutomationExecutionAPI dengan token tugas dari alur kerja mesin status, dan berhenti sampai token kembali dengan panggilan sukses atau gagal. Untuk mendemonstrasikan integrasi, pola ini mengimplementasikan pembungkus dokumen Otomasi (runbook) di sekitar `AWS-RunPowerShellScript` dokumen `AWS-RunShellScript` atau, dan digunakan `.waitForTaskToken` untuk memanggil atau secara sinkron. `AWS-RunShellScript` `AWS-RunPowerShellScript` Untuk informasi selengkapnya tentang integrasi layanan AWS SDK di Step Functions, lihat Panduan AWS Step Functions Pengembang.

Step Functions adalah layanan alur kerja visual dengan kode rendah yang dapat Anda gunakan untuk membangun aplikasi terdistribusi, mengotomatiskan proses TI dan bisnis, serta membangun jaringan data dan pembelajaran mesin dengan menggunakan layanan. AWS Alur kerja mengelola kegagalan, percobaan ulang, paralelisasi, integrasi layanan, dan observabilitas sehingga Anda dapat fokus pada logika bisnis bernilai lebih tinggi.

Otomatisasi, kemampuan AWS Systems Manager, menyederhanakan tugas pemeliharaan, penyebaran, dan remediasi umum Layanan AWS seperti Amazon Elastic Compute Cloud (Amazon), Amazon Relational Database Service (Amazon RDS EC2), Amazon Redshift, dan Amazon Simple Storage Service (Amazon S3). Otomasi memberi Anda kontrol terperinci atas konkurensi otomatisasi Anda. Misalnya, Anda dapat menentukan berapa banyak sumber daya yang akan ditargetkan secara bersamaan, dan berapa banyak kesalahan yang dapat terjadi sebelum otomatisasi dihentikan.

Untuk detail implementasi, termasuk langkah runbook, parameter, dan contoh, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- AWS Akun aktif

- AWS Identity and Access Management Izin (IAM) untuk mengakses Step Functions dan Systems Manager
- Sebuah EC2 instance dengan Systems Manager Agent (SSM Agent) diinstal pada instance
- Profil instans IAM untuk Systems Manager yang dilampirkan ke instance tempat Anda berencana menjalankan runbook
- Peran Step Functions yang memiliki izin IAM berikut (yang mengikuti prinsip hak istimewa terkecil):

```
{
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": "*"
 }
```

Versi produk

- Skema dokumen SSM versi 0.3 atau yang lebih baru
- Agen SSM versi 2.3.672.0 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- AWS Step Functions
- AWS Systems Manager Otomasi

Arsitektur target

Otomatisasi dan skala

- Pola ini menyediakan AWS CloudFormation template yang dapat Anda gunakan untuk menyebarkan runbook pada beberapa instance. (Lihat GitHub Step Functions dan repositori implementasi Systems Manager.)

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar.

Kode

Kode untuk pola ini tersedia di repositori implementasi GitHub Step Functions dan Systems Manager.

Epik

Buat runbook

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh CloudFormation template.	Unduh `ssm-automation-documents.cfn.json` templat dari `cloudformation` folder GitHub repositori.	AWS DevOps
Buat runbook.	Masuk ke Konsol Manajemen AWS, buka CloudFormation konsol, dan terapkan template. Untuk informasi selengkapnya tentang penerapan CloudFormation templat, lihat Membuat	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tumpukan di CloudFormation konsol dalam CloudFormation dokumentasi. CloudFormation Template menyebarkan tiga sumber daya: • `SfnRunCommandByInstanceIds` — Runbook yang memungkinkan Anda menjalankan `AWS-RunShellScript` atau `AWS-RunPowerShellScript` dengan menggunakan instance IDs. • `SfnRunCommandByTargets` — Runbook yang memungkinkan Anda menjalankan `AWS-RunShellScript` atau `AWS-RunPowerShellScript` dengan menggunakan target. • `SSMSyncRole` — Peran IAM diasumsikan oleh runbook.	

Buat mesin status sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat mesin status uji.	Ikuti petunjuk di Panduan AWS Step Functions	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pengembang untuk membuat dan menjalankan mesin status. Untuk definisi, gunakan kode berikut. Pastikan untuk memperbarui `InstanceIds` nilai dengan ID instance Systems Manager-enabled yang valid di akun Anda. `{ "Comment": "A description of my state machine", "StartAt": "StartAutomationWaitForCallBack", "States": { "StartAutomationWaitForCallBack": { "Type": "Task", "Resource": "arn:aws:states:::aws-sdk:ssm:startAutomationExecution.waitForTaskToken", "Parameters": { "DocumentName": "SfnRunCommandByInstanceIds", "Parameters": { "InstanceIds": ["i-1234567890abcdef0"], "taskToken.$": "States.Array($$.Task.Token)", "workingDirectory": [`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"/home/ssm-user/"
	],
	"Commands": [
	"echo \"This is a test running automation waitForTaskToken\" >> automation.log",
	"sleep 100"
	],
	"executionTimeout": [
	"10800"
	],
	"deliveryTimeout": [
	"30"
	],
	"shell": [
	"Shell"
	]
	}
	},
	"End": true
	}
	}
	}</pre> Kode ini memanggil runbook untuk menjalankan dua perintah yang mendemonstrasikan `waitForTaskToken` panggilan ke Systems Manager Automation.
	
	Nilai `shell` parameter (`Shell`atau`PowerShell`) menentukan apakah dokumen Otomasi berjalan `AWS-RunSh`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ellScript` atau`AWS-RunPowerShellScript` . Tugas menulis “Ini adalah pengujian yang menjalankan waitForTask Token otomatisasi” ke dalam `/home/ssm-user/automation.log` file, dan kemudian tidur selama 100 detik sebelum merespons dengan token tugas dan melepaskan tugas berikutnya dalam alur kerja. Jika Anda ingin memanggil `SfnRunCommandByTargets` runbook sebagai gantinya, ganti `Parameters` bagian dari kode sebelumnya dengan yang berikut: `"Parameters": { "Targets": [ { "Key": "InstanceIds", "Values": [ "i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE" ] } ],`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui peran IAM untuk mesin status.	Langkah sebelumnya secara otomatis membuat peran IAM khusus untuk mesin status. Namun, itu tidak memberikan izin untuk memanggil runbook. Perbarui peran dengan menambahkan izin berikut: `{ "Effect": "Allow", "Action": "ssm:StartAutomationExecution", "Resource": "*" }`	AWS DevOps
Validasi panggilan sinkron.	Jalankan state machine untuk memvalidasi panggilan sinkron antara Step Functions dan Systems Manager Automation. Untuk keluaran sampel, lihat bagian Informasi tambahan.	AWS DevOps

Sumber daya terkait

- Memulai dengan AWS Step Functions (Panduan AWS Step Functions Pengembang)
- Tunggu panggilan balik dengan token tugas (Panduan AWS Step Functions Pengembang, pola integrasi layanan)
- panggilan API send_task_success dan send_task_failure (dokumentasi Boto3)
- AWS Systems Manager Otomatisasi (Panduan AWS Systems Manager Pengguna)

Informasi tambahan

Detail implementasi

Pola ini menyediakan CloudFormation template yang menyebarkan dua runbook Systems Manager:

- `SfnRunCommandByInstanceIds`menjalankan `AWS-RunPowerShellScript` perintah `AWS-RunShellScript` or dengan menggunakan instance IDs.
- `SfnRunCommandByTargets`menjalankan `AWS-RunPowerShellScript` perintah `AWS-RunShellScript` or dengan menggunakan target.

Setiap runbook mengimplementasikan empat langkah untuk mencapai panggilan sinkron saat menggunakan `.waitForTaskToken` opsi di Step Functions.

Langkah	Tindakan	Deskripsi
1	`Branch`	Memeriksa nilai `shell` parameter (`Shell`atau`PowerShell`) untuk memutuskan apakah akan dijalankan `AWS-RunShellScript` untuk Linux atau `AWS-RunPowerShellScript` untuk Windows.
2	`RunCommand_Shell` atau `RunCommand_PowerShell`	Mengambil beberapa input dan menjalankan `RunPowerShellScript` perintah `RunShellScript` atau. Untuk informasi selengkapnya, periksa tab Detail untuk dokumen `RunCommand_PowerShell` Otomatisasi `RunCommand_Shell` atau di konsol Systems Manager.
3	`SendTaskFailure`	Berjalan saat langkah 2 dibatalkan atau dibatalkan.

		Ini memanggil Step Functions send_task_failure API, yang menerima tiga parameter sebagai input: token yang dilewatkan oleh mesin status, kesalahan kegagalan, dan deskripsi penyebab kegagalan .
4	`SendTaskSuccess`	Berjalan ketika langkah 2 berhasil. Ini memanggil Step Functions send_task_success API, yang menerima token yang diteruskan oleh mesin status sebagai input.

Parameter buku runbook

`SfnRunCommandByInstanceIds`buku runbook:

Nama parameter	Jenis	Opsional atau diperlukan	Deskripsi
`shell`	String	Diperlukan	Contoh shell untuk memutuskan apakah akan dijalankan `AWS-RunShellScript` untuk Linux atau `AWS-RunPowerShellScript` untuk Windows.
`deliveryTimeout`	Bilangan Bulat	Opsional	Waktu, dalam hitungan detik, untuk menunggu perintah dikirim ke Agen SSM pada

			sebuah instance. Parameter ini memiliki nilai minimum 30 (0,5 menit) dan nilai maksimum 2592000 (720 jam).
`executionTimeout`	String	Opsional	Waktu, dalam hitungan detik, untuk menyelesaikan perintah sebelum dianggap gagal. Nilai defaultnya adalah 3600 (1 jam). Nilai maksimum adalah 172800 (48 jam).
`workingDirectory`	String	Opsional	Jalur direktori kerja pada instans Anda.
`Commands`	StringList	Diperlukan	Skrip shell atau perintah untuk dijalankan.
`InstanceIds`	StringList	Diperlukan	IDs Contoh di mana Anda ingin menjalankan perintah.
`taskToken`	String	Diperlukan	Token tugas yang digunakan untuk respons callback.

`SfnRunCommandByTargets`buku runbook:

Nama	Jenis	Opsional atau diperlukan	Deskripsi

`shell`	String	Diperlukan	Contoh shell untuk memutuskan apakah akan dijalankan `AWS-RunShellScript` untuk Linux atau `AWS-RunPowerShellScript` untuk Windows.
`deliveryTimeout`	Bilangan Bulat	Opsional	Waktu, dalam hitungan detik, untuk menunggu perintah dikirim ke Agen SSM pada sebuah instance. Parameter ini memiliki nilai minimum 30 (0,5 menit) dan nilai maksimum 2592000 (720 jam).
`executionTimeout`	Bilangan Bulat	Opsional	Waktu, dalam hitungan detik, untuk menyelesaikan perintah sebelum dianggap gagal. Nilai defaultnya adalah 3600 (1 jam). Nilai maksimum adalah 172800 (48 jam).
`workingDirectory`	String	Opsional	Jalur direktori kerja pada instans Anda.
`Commands`	StringList	Diperlukan	Skrip shell atau perintah untuk dijalankan.

`Targets`	MapList	Diperlukan	Array kriteria pencarian yang mengidentifikasi instance dengan menggunakan pasangan kunci-nilai yang Anda tentukan. Misalnya: `[{"Key":"InstanceIds","Values":["i-02573cafcfEXAMPLE","i-0471e04240EXAMPLE"]}]`
`taskToken`	String	Diperlukan	Token tugas yang digunakan untuk respons callback.

Keluaran sampel

Tabel berikut memberikan output sampel dari fungsi langkah. Ini menunjukkan bahwa total waktu berjalan lebih dari 100 detik antara langkah 5 (`TaskSubmitted`) dan langkah 6 (`TaskSucceeded`). Ini menunjukkan bahwa fungsi langkah menunggu `sleep 100` perintah selesai sebelum pindah ke tugas berikutnya dalam alur kerja.

ID	Jenis	Langkah	Sumber daya	Waktu Berlalu (ms)	Stempel Waktu
1	`ExecutionStarted`		-	0	11 Mar 2022 02:50:34.303
2	`TaskStateEntered`	`StartAutomationWai`	-	40	11 Mar 2022 02:50:34.343

		tForCallBack			
3	TaskScheduled	StartAutomationWaitForCallBack	-	40	11 Mar 2022 02:50:34.343
4	TaskStarted	StartAutomationWaitForCallBack	-	154	11 Mar 2022 02:50:34.457
5	TaskSubmitted	StartAutomationWaitForCallBack	-	657	11 Mar 2022 02:50:34.960
6	TaskSucceeded	StartAutomationWaitForCallBack	-	103835	11 Mar 2022 02:52:18.138
7	TaskStateExited	StartAutomationWaitForCallBack	-	103860	11 Mar 2022 02:52:18.163
8	ExecutionSucceeded		-	103897	11 Mar 2022 02:52:18.200

Jalankan pembacaan paralel objek S3 dengan menggunakan Python dalam suatu fungsi AWS Lambda

Eduardo Bortoluzzi, Amazon Web Services

Ringkasan

Anda dapat menggunakan pola ini untuk mengambil dan meringkas daftar dokumen dari Amazon Simple Storage Service (Amazon S3) bucket secara real time. Pola ini memberikan contoh kode ke objek baca paralel dari bucket S3 di Amazon Web Services ()AWS. Pola ini menampilkan cara menjalankan tugas I/O terikat secara efisien dengan AWS Lambda fungsi menggunakan Python.

Sebuah perusahaan keuangan menggunakan pola ini dalam solusi interaktif untuk secara manual menyetujui atau menolak transaksi keuangan yang berkorelasi secara real time. Dokumen transaksi keuangan disimpan dalam ember S3 yang terkait dengan pasar. Operator memilih daftar dokumen dari bucket S3, menganalisis nilai total transaksi yang dihitung oleh solusi, dan memutuskan untuk menyetujui atau menolak batch yang dipilih.

Tugas terikat I/O mendukung banyak utas. Dalam contoh kode ini, concurrent.futures.ThreadPoolExecutordigunakan dengan maksimal 30 utas simultan, meskipun fungsi Lambda mendukung hingga 1.024 utas (dengan salah satu utas tersebut menjadi proses utama Anda). Batas ini karena terlalu banyak utas menciptakan masalah latensi karena peralihan konteks dan pemanfaatan sumber daya komputasi. Anda juga perlu meningkatkan koneksi pool maksimum `botocore` sehingga semua thread dapat melakukan download objek S3 secara bersamaan.

Kode contoh menggunakan satu objek 8,3 KB, dengan data JSON, dalam ember S3. Objek dibaca beberapa kali. Setelah fungsi Lambda membaca objek, data JSON diterjemahkan ke objek Python. Pada bulan Desember 2024, hasil setelah menjalankan contoh ini adalah 1.000 pembacaan diproses dalam 2,3 detik dan 10.000 pembacaan diproses dalam 27 detik menggunakan fungsi Lambda yang dikonfigurasi dengan memori 2.304 MB. AWS Lambda mendukung konfigurasi memori dari 128 MB menjadi 10.240 MB (10 GB), meskipun meningkatkan Lambdamemory melebihi 2.304 MB tidak membantu mengurangi waktu untuk menjalankan tugas terikat I/O khusus ini.

Alat AWS Lambda Power Tuning digunakan untuk menguji konfigurasi memori Lambda yang berbeda dan memverifikasi rasio performance-to-cost terbaik untuk tugas tersebut. Untuk hasil tes, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Kemahiran dengan pengembangan Python

Batasan

- Fungsi Lambda dapat memiliki paling banyak 1.024 proses eksekusi atau utas.
- Baru Akun AWS memiliki batas memori Lambda 3.008 MB. Sesuaikan alat AWS Lambda Power Tuning yang sesuai. Untuk informasi selengkapnya, lihat bagian Pemecahan Masalah.
- Amazon S3 memiliki batas 5.500 GET/HEAD permintaan per detik per awalan yang dipartisi.

Versi produk

- Python 3.9 atau yang lebih baru
- AWS Cloud Development Kit (AWS CDK) v2
- AWS Command Line Interface (AWS CLI) versi 2
- AWS Lambda Power Tuning 4.3.6 (opsional)

Arsitektur

Tumpukan teknologi target

- AWS Lambda
- Amazon S3
- AWS Step Functions (jika AWS Lambda Power Tuning digunakan)

Arsitektur target

Diagram berikut menunjukkan fungsi Lambda yang membaca objek dari bucket S3 secara paralel. Diagram juga memiliki alur kerja Step Functions untuk alat AWS Lambda Power Tuning untuk menyempurnakan memori fungsi Lambda. Penyetelan halus ini membantu mencapai keseimbangan yang baik antara biaya dan kinerja.

Otomatisasi dan skala

Fungsi Lambda berskala cepat saat diperlukan. Untuk menghindari kesalahan 503 Slow Down dari Amazon S3 selama permintaan tinggi, kami sarankan untuk membatasi penskalaan.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK) v2 adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode. Infrastruktur contoh dibuat untuk digunakan. AWS CDK
- AWS Command Line InterfaceAWS CLIadalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda. Dalam pola ini, AWS CLI versi 2 digunakan untuk mengunggah contoh file JSON.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service Amazon S3 adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi AWS Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis.

Alat-alat lainnya

- Python adalah bahasa pemrograman komputer tujuan umum. Penggunaan kembali thread idle worker diperkenalkan di Python versi 3.8, dan kode fungsi Lambda dalam pola ini dibuat untuk Python versi 3.9 dan yang lebih baru.

Repositori kode

Kode untuk pola ini tersedia di aws-lambda-parallel-download GitHub repositori.

Praktik terbaik

- AWS CDK Konstruksi ini bergantung pada izin Akun AWS pengguna Anda untuk menerapkan infrastruktur. Jika Anda berencana menggunakan AWS CDK Pipelines atau penerapan lintas akun, lihat Stack synthesizer.

- Aplikasi contoh ini tidak mengaktifkan log akses di bucket S3. Ini adalah praktik terbaik untuk mengaktifkan log akses dalam kode produksi.

Epik

Mempersiapkan lingkungan pengembangan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa versi yang diinstal Python.	Kode ini telah diuji secara khusus pada Python 3.9 dan Python 3.13, dan harus bekerja pada semua versi di antara rilis ini. Untuk memeriksa versi Python Anda, jalankan `python3 -V` di terminal Anda, dan instal versi yang lebih baru jika diperluka n. Untuk memverifikasi bahwa modul yang diperlukan diinstal, jalankan`python3 -c "import pip, venv"`. Tidak ada pesan kesalahan berarti modul diinstal dengan benar dan Anda siap untuk menjalankan contoh ini.	Arsitek awan
Instal AWS CDK.	Untuk menginstal AWS CDK jika belum diinstal, ikuti petunjuk di Memulai dengan AWS CDK. Untuk mengonfir masi bahwa AWS CDK versi yang diinstal adalah 2.0 atau yang lebih baru, jalankan`cdk -version`.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap lingkungan Anda.	Untuk bootstrap lingkungan Anda, jika belum dilakukan , ikuti instruksi di Bootstrap lingkungan Anda untuk digunakan dengan AWS CDK.	Arsitek awan

Kloning repositori contoh

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning versi terbaru dari repositori, jalankan perintah berikut: `git clone --depth 1 --branch v1.2.0 \ git@github.com:aws-samples/aws-lambda-parallel-download.git`	Arsitek awan
Ubah direktori kerja ke repositori kloning.	Jalankan perintah berikut: `cd aws-lambda-parallel-download`	Arsitek awan
Buat lingkungan virtual Python.	Untuk membuat lingkungan virtual Python, jalankan perintah berikut: `python3 -m venv .venv`	Arsitek awan
Aktifkan lingkungan virtual.	Untuk mengaktifkan lingkungan virtual, jalankan perintah berikut:	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`source .venv/bin/activate`	
Instal dependensi.	Untuk menginstal dependensi Python, jalankan perintah: `pip` `pip install -r requirements.txt`	Arsitek awan
Jelajahi kodenya.	(Opsional) Contoh kode yang mengunduh objek dari bucket S3 ada di`resources/parallel.py` . Kode infrastruktur ada di `parallel_download` folder.	Arsitek awan

Terapkan dan uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan aplikasi.	Jalankan `cdk deploy`. Tuliskan AWS CDK outputnya: • `ParallelDownloadStack.LambdaFunctionARN` • `ParallelDownloadStack.SampleS3BucketName`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `ParallelDownloadStack.StateMachineARN`	
Unggah contoh file JSON.	Repositori berisi contoh file JSON sekitar 9 KB. Untuk mengunggah file ke bucket S3 dari tumpukan yang dibuat, jalankan perintah berikut: `aws s3 cp sample.json s3://<ParallelDownloadStack.SampleS3BucketName>` Ganti `<ParallelDownloadStack.SampleS3BucketName>` dengan nilai yang sesuai dari AWS CDK output.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan aplikasi.	Untuk menjalankan aplikasi, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, navigasikan ke konsol Lambda, dan temukan fungsi Lambda yang memiliki ARN dari output. AWS CDK `ParallelDownloadStack.LambdaFunction` ARN 2. Pada tab Uji, ubah Event JSON menjadi yang berikut: `{"objectKey": "sample.json"}` 3. Pilih Uji. 4. Untuk melihat hasilnya, pilih detail. Detailnya akan menunjukkan statistik unduhan paralel, informasi run, dan log.	Arsitek awan
Tambahkan jumlah unduhan.	(Opsional) Untuk menjalankan 1.500 get panggilan objek, gunakan JSON berikut di Event JSON parameter: `Test` `{"repeat": 1500, "objectKey": "sample.json"}`	Arsitek awan

Opsional: Jalankan AWS Lambda Power Tuning

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan alat AWS Lambda Power Tuning.	1. Masuk ke konsol, dan navigasikan ke Step Functions. 2. Temukan mesin negara dengan ARN dari output. AWS CDK `ParallelDownloadStack.StateMachineARN` 3. Pilih Mulai eksekusi, dan tempel JSON berikut: `{ "lambdaARN": "<ParallelDownloadStack.LambdaFunctionARN>", "num": 10, "strategy": "balanced", "payload": {"repeat": 2000, "objectKey": "sample.json"} }` Ingatlah untuk mengganti `<ParallelDownloadStack.LambdaFunctionARN>` dengan nilai dari AWS CDK output. Di akhir proses, hasilnya akan berada di tab input dan output Eksekusi.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat hasil AWS Lambda Power Tuning dalam grafik.	Pada tab Input dan output eksekusi, salin tautan `visualization` properti, dan tempel di tab browser baru.	Arsitek awan

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus objek dari ember S3.	Sebelum Anda menghancurkan sumber daya yang digunakan, Anda menghapus semua objek dari bucket S3: `aws s3 rm s3://<ParallelDownloadStack.SampleS3BucketName> \ --recursive` Ingatlah untuk mengganti `<ParallelDownloadStack.SampleS3BucketName>` dengan nilai dari AWS CDK output.	Arsitek awan
Hancurkan sumber daya.	Untuk menghancurkan semua sumber daya yang dibuat untuk pilot ini, jalankan perintah berikut: `cdk destroy`	Arsitek awan

Pemecahan masalah

Isu	Solusi
`'MemorySize' value failed to satisfy constraint: Member must have value less than or equal to 3008`	Untuk akun baru, Anda mungkin tidak dapat mengonfigurasi lebih dari 3.008 MB di fungsi Lambda Anda. Untuk menguji menggunak an AWS Lambda Power Tuning, tambahkan properti berikut pada input JSON saat Anda memulai eksekusi Step Functions: `"powerValues": [ 512, 1024, 1536, 2048, 2560, 3008 ]`

Sumber daya terkait

- Python — concurrent.futures. ThreadPoolExecutor
- Kuota Lambda - Konfigurasi fungsi, penerapan, dan eksekusi
- Bekerja dengan AWS CDK in Python
- Fungsi pembuatan profil dengan AWS Lambda Power Tuning

Informasi tambahan

Kode

Cuplikan kode berikut melakukan pemrosesan I/O paralel:

```
with ThreadPoolExecutor(max_workers=MAX_WORKERS) as executor:
  for result in executor.map(a_function, (the_arguments)):
    ...
```

`ThreadPoolExecutor`Menggunakan kembali utas saat tersedia.

Pengujian dan hasil

Tes ini dilakukan pada Desember 2024.

Tes pertama memproses 2.500 pembacaan objek, dengan hasil sebagai berikut.

Mulai dari 3.009 MB, tingkat waktu pemrosesan tetap hampir sama untuk setiap peningkatan memori, tetapi biaya meningkat seiring dengan meningkatnya ukuran memori.

Tes lain menyelidiki kisaran antara 1.536 MB dan 3.072 MB memori, menggunakan nilai yang kelipatan 256 MB dan memproses 10.000 pembacaan objek, dengan hasil sebagai berikut.

performance-to-costRasio terbaik adalah dengan konfigurasi Lambda memori 2.304 MB.

Sebagai perbandingan, proses sekuensial 2.500 pembacaan objek membutuhkan waktu 47 detik. Proses paralel menggunakan konfigurasi Lambda 2.304 MB membutuhkan waktu 7 detik, yaitu 85 persen lebih sedikit.

Kirim data telemetri dari AWS Lambda ke OpenSearch untuk analitik dan visualisasi waktu nyata

Tabby Ward, Guy Bachar, dan David Kilzer, Amazon Web Services

Ringkasan

Aplikasi modern menjadi semakin terdistribusi dan didorong oleh peristiwa, yang memperkuat kebutuhan akan pemantauan dan observabilitas waktu nyata. AWS Lambda adalah layanan komputasi tanpa server yang memainkan peran penting dalam membangun arsitektur yang dapat diskalakan dan digerakkan oleh peristiwa. Namun, pemantauan dan pemecahan masalah fungsi Lambda dapat menjadi tantangan jika Anda hanya mengandalkan Amazon CloudWatch Logs, yang dapat memperkenalkan latensi dan periode retensi terbatas.

Untuk mengatasi tantangan ini, AWS memperkenalkan Lambda Telemetry API, yang memungkinkan fungsi Lambda mengirim data telemetri langsung ke alat pemantauan dan observabilitas pihak ketiga. API ini mendukung streaming real-time log, metrik, dan jejak, serta memberikan tampilan yang komprehensif dan tepat waktu tentang kinerja dan kesehatan fungsi Lambda Anda.

Pola ini menjelaskan cara mengintegrasikan API Telemetri Lambda dengan OpenSearch, yang merupakan mesin pencari dan analitik terdistribusi sumber terbuka. OpenSearch menawarkan platform yang kuat dan terukur untuk menelan, menyimpan, dan menganalisis volume data yang besar, yang menjadikannya pilihan ideal untuk data telemetri Lambda. Secara khusus, pola ini menunjukkan cara mengirim log dari fungsi Lambda yang ditulis dengan Python langsung ke cluster dengan menggunakan OpenSearch ekstensi Lambda yang disediakan oleh. AWS Solusi ini fleksibel dan dapat disesuaikan, sehingga Anda dapat membuat ekstensi Lambda Anda sendiri atau mengubah kode sumber sampel untuk mengubah format output sesuai keinginan.

Pola ini menjelaskan cara mengatur dan mengonfigurasi integrasi API Telemetri Lambda dengan OpenSearch, dan mencakup praktik terbaik untuk keamanan, pengoptimalan biaya, dan skalabilitas. Tujuannya adalah untuk membantu Anda mendapatkan wawasan yang lebih dalam tentang fungsi Lambda Anda dan meningkatkan pengamatan keseluruhan aplikasi tanpa server Anda.

> Catatan: Pola ini berfokus pada integrasi Lambda Telemetry API dengan managed. OpenSearc h Namun, prinsip dan teknik yang dibahas juga berlaku untuk self-managed OpenSearch dan Elasticsearch.

Prasyarat dan batasan

Sebelum Anda memulai proses integrasi, pastikan Anda memiliki prasyarat berikut:

Akun AWS: Aktif Akun AWS dengan izin yang sesuai untuk membuat dan mengelola AWS sumber daya berikut:

- AWS Lambda
- AWS Identity and Access Management (IAM)
- OpenSearch Layanan Amazon (jika Anda menggunakan OpenSearch kluster terkelola)

OpenSearch klaster:

- Anda dapat menggunakan OpenSearch kluster yang dikelola sendiri atau layanan terkelola seperti OpenSearch Layanan.
- Jika Anda menggunakan OpenSearch Layanan, siapkan OpenSearch klaster Anda dengan mengikuti petunjuk di Memulai OpenSearch Layanan Amazon di dokumentasi OpenSearch Layanan.
- Pastikan OpenSearch klaster dapat diakses dari fungsi Lambda Anda dan dikonfigurasi dengan pengaturan keamanan yang diperlukan, seperti kebijakan akses, enkripsi, dan otentikasi.
- Konfigurasikan OpenSearch cluster dengan pemetaan dan pengaturan indeks yang diperlukan untuk menyerap data telemetri Lambda. Untuk informasi selengkapnya, lihat Memuat data streaming ke OpenSearch Layanan Amazon di dokumentasi OpenSearch Layanan.

Konektivitas jaringan:

- Pastikan fungsi Lambda Anda memiliki konektivitas jaringan yang diperlukan untuk mengakses cluster. OpenSearch Untuk panduan tentang cara mengonfigurasi setelan virtual private cloud (VPC), lihat Meluncurkan domain OpenSearch Layanan Amazon Anda dalam VPC di dokumentasi Layanan. OpenSearch

Peran dan kebijakan IAM:

- Buat peran IAM dengan izin yang diperlukan untuk fungsi Lambda Anda untuk mengakses OpenSearch klaster dan mengakses kredensil Anda yang disimpan di dalamnya. AWS Secrets Manager

- Lampirkan kebijakan IAM yang sesuai ke peran, seperti `AWSLambdaBasicExecutionRole` kebijakan dan izin tambahan apa pun yang diperlukan untuk berinteraksi. OpenSearch
- Verifikasi bahwa izin IAM yang diberikan ke fungsi Lambda Anda memungkinkannya menulis data ke cluster. OpenSearch Untuk informasi tentang mengelola izin IAM, lihat Mendefinisikan izin fungsi Lambda dengan peran eksekusi dalam dokumentasi Lambda.

Pengetahuan bahasa pemrograman:

- Anda akan memerlukan pengetahuan dasar tentang Python (atau bahasa pemrograman pilihan Anda) untuk memahami dan memodifikasi kode sampel untuk fungsi Lambda dan ekstensi Lambda.

Lingkungan pengembangan:

- Siapkan lingkungan pengembangan lokal dengan alat dan dependensi yang diperlukan untuk membangun dan menerapkan fungsi dan ekstensi Lambda.

AWS CLI atau Konsol Manajemen AWS:

- Instal dan konfigurasikan AWS Command Line Interface (AWS CLI) atau gunakan Konsol Manajemen AWS dengan kredensil yang sesuai untuk berinteraksi dengan yang diperlukan. Layanan AWS

Pemantauan dan pencatatan:

- Biasakan diri dengan pemantauan dan pencatatan praktik terbaik AWS, termasuk layanan seperti Amazon CloudWatch dan AWS CloudTrail untuk tujuan pemantauan dan audit.
- Periksa CloudWatch Log untuk fungsi Lambda Anda untuk mengidentifikasi kesalahan atau pengecualian yang terkait dengan integrasi API Telemetri Lambda. Untuk panduan pemecahan masalah, lihat dokumentasi Lambda Telemetry API.

Arsitektur

Pola ini menggunakan OpenSearch Layanan untuk menyimpan log dan data telemetri yang dihasilkan oleh fungsi Lambda. Pendekatan ini memungkinkan Anda untuk dengan cepat

mengalirkan log langsung ke OpenSearch klaster Anda, yang mengurangi latensi dan biaya yang terkait dengan penggunaan CloudWatch Log sebagai perantara.

> **Note**
>
> Kode ekstensi Lambda Anda dapat mendorong telemetri ke OpenSearch Layanan, baik dengan langsung menggunakan OpenSearch API atau dengan menggunakan pustaka klien. OpenSearch Ekstensi Lambda dapat menggunakan operasi massal yang didukung oleh OpenSearch API untuk menggabungkan peristiwa telemetri bersama-sama dan mengirimkannya ke OpenSearch Layanan dalam satu permintaan.

Diagram alur kerja berikut menggambarkan alur kerja log untuk fungsi Lambda saat Anda menggunakan klaster sebagai OpenSearch titik akhir.

Arsitektur mencakup komponen-komponen ini:

- Fungsi Lambda: Fungsi tanpa server yang menghasilkan log dan data telemetri selama eksekusi.
- Ekstensi Lambda: Ekstensi berbasis Python yang menggunakan API Telemetri Lambda untuk berintegrasi langsung dengan cluster. OpenSearch Ekstensi ini berjalan bersama fungsi Lambda di lingkungan eksekusi yang sama.
- Lambda Telemetry API: API yang memungkinkan ekstensi Lambda mengirim data telemetri, termasuk log, metrik, dan jejak, langsung ke alat pemantauan dan observabilitas pihak ketiga.
- Kluster OpenSearch Layanan Amazon: OpenSearch Kluster terkelola yang di-host AWS. Cluster ini bertanggung jawab untuk menelan, menyimpan, dan mengindeks data log yang dialirkan dari fungsi Lambda melalui ekstensi Lambda.

Alur kerja terdiri dari langkah-langkah ini:

1. Fungsi Lambda dipanggil, dan menghasilkan log dan data telemetri selama pelaksanaannya.
2. Ekstensi Lambda berjalan di samping fungsi untuk menangkap log dan data telemetri dengan menggunakan Lambda Telemetry API.
3. Ekstensi Lambda membuat koneksi aman dengan kluster OpenSearch Layanan dan mengalirkan data log secara real time.

4. Kluster OpenSearch Layanan menyerap, mengindeks, dan menyimpan data log untuk membuatnya tersedia untuk pencarian, analisis, dan visualisasi melalui penggunaan alat seperti Kibana atau aplikasi lain yang kompatibel.

Dengan menghindari CloudWatch Log dan mengirimkan data log langsung ke OpenSearch cluster, solusi ini memberikan beberapa manfaat:

- Streaming dan analisis log waktu nyata, memungkinkan pemecahan masalah yang lebih cepat dan peningkatan pengamatan.
- Mengurangi latensi dan potensi pembatasan retensi yang terkait dengan CloudWatch Log.
- Fleksibilitas untuk menyesuaikan ekstensi Lambda atau membuat ekstensi Anda sendiri untuk format keluaran tertentu atau pemrosesan tambahan.
- Integrasi dengan kemampuan pencarian, analitik, dan visualisasi OpenSearch Layanan untuk analisis dan pemantauan log.

Bagian Epics memberikan step-by-step instruksi untuk menyiapkan ekstensi Lambda, mengonfigurasi fungsi Lambda, dan mengintegrasikan dengan kluster Layanan. OpenSearch Untuk pertimbangan keamanan, strategi pengoptimalan biaya, dan tips untuk memantau dan memecahkan masalah solusi, lihat bagian Praktik terbaik.

Alat

Layanan AWS

- AWS Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon OpenSearch Service adalah layanan terkelola penuh yang disediakan oleh AWS yang memudahkan untuk menyebarkan, mengoperasikan, dan menskalakan OpenSearch cluster di cloud.
- Ekstensi Lambda memperluas fungsionalitas fungsi Lambda Anda dengan menjalankan kode khusus di sampingnya. Anda dapat menggunakan ekstensi Lambda untuk mengintegrasikan Lambda dengan berbagai alat pemantauan, observabilitas, keamanan, dan tata kelola.
- AWS Lambda API telemetri memungkinkan Anda menggunakan ekstensi untuk menangkap data pemantauan dan observabilitas yang disempurnakan langsung dari Lambda dan mengirimkannya ke tujuan pilihan Anda.

- CloudFormationmembantu Anda memodelkan dan mengatur AWS sumber daya sehingga Anda dapat menghabiskan lebih sedikit waktu untuk mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi Anda.

Repositori kode

- AWS Lambda Ekstensi mencakup demo dan proyek sampel dari AWS dan AWS Mitra untuk membantu Anda memulai membangun ekstensi Anda sendiri.
- Contoh Integrasi Telemetri Lambda untuk menyediakan OpenSearch contoh ekstensi Lambda yang menunjukkan cara mengirim log dari fungsi Lambda ke cluster. OpenSearch

Alat lainnya

- OpenSearchadalah mesin pencari dan analitik terdistribusi open-source yang menyediakan platform yang kuat untuk menelan, menyimpan, dan menganalisis volume data yang besar.
- Kibana adalah alat visualisasi dan eksplorasi data sumber terbuka yang dapat Anda gunakan. OpenSearch Perhatikan bahwa implementasi visualisasi dan analitik berada di luar cakupan pola ini. Untuk informasi selengkapnya, lihat dokumentasi Kibana dan sumber daya lainnya.

Praktik terbaik

Saat Anda mengintegrasikan API Telemetri Lambda dengan OpenSearch, pertimbangkan praktik terbaik berikut.

Keamanan dan kontrol akses

- Komunikasi aman: Enkripsi semua komunikasi antara fungsi Lambda Anda dan klaster dengan OpenSearch menggunakan HTTPS. Konfigurasikan SSL/TLS pengaturan yang diperlukan di ekstensi dan OpenSearch konfigurasi Lambda Anda.
- Izin IAM:
  - Ekstensi berjalan di lingkungan eksekusi yang sama dengan fungsi Lambda, sehingga mereka mewarisi tingkat akses yang sama ke sumber daya seperti sistem file, jaringan, dan variabel lingkungan.
  - Berikan izin IAM minimum yang diperlukan ke fungsi Lambda Anda untuk mengakses API Telemetri Lambda dan menulis data ke cluster. OpenSearch Gunakan prinsip hak istimewa terkecil untuk membatasi ruang lingkup izin.

- OpenSearch kontrol akses: Terapkan kontrol akses berbutir halus di OpenSearch klaster Anda untuk membatasi akses ke data sensitif. Gunakan fitur keamanan bawaan, seperti otentikasi pengguna, kontrol akses berbasis peran, dan izin tingkat indeks, di. OpenSearch
- Ekstensi tepercaya: Selalu pasang ekstensi dari sumber tepercaya saja. Gunakan alat infrastruktur sebagai kode (IAc) seperti CloudFormation untuk menyederhanakan proses melampirkan konfigurasi ekstensi yang sama, termasuk izin IAM, ke beberapa fungsi Lambda. Alat IAC juga menyediakan catatan audit ekstensi dan versi yang digunakan sebelumnya.
- Penanganan data sensitif: Saat membuat ekstensi, hindari mencatat data sensitif. Sanitasi muatan dan metadata sebelum mencatat atau menyimpannya untuk tujuan audit.

Optimalisasi biaya

- Pemantauan dan peringatan: Siapkan mekanisme pemantauan dan peringatan untuk melacak volume data yang dikirim OpenSearch dari fungsi Lambda Anda. Ini akan membantu Anda mengidentifikasi dan mengatasi potensi pembengkakan biaya.
- Retensi data: Pertimbangkan dengan cermat periode retensi data yang sesuai untuk data telemetri Lambda Anda di. OpenSearch Periode retensi yang lebih lama dapat meningkatkan biaya penyimpanan, jadi seimbangkan kebutuhan observabilitas Anda dengan pengoptimalan biaya.
- Kompresi dan pengindeksan: Aktifkan kompresi data dan optimalkan strategi OpenSearch pengindeksan Anda untuk mengurangi jejak penyimpanan data telemetri Lambda Anda.
- Mengurangi ketergantungan pada CloudWatch: Dengan mengintegrasikan API Telemetri Lambda secara langsung dengan OpenSearch, Anda berpotensi mengurangi ketergantungan Anda pada CloudWatch Log, yang dapat menghasilkan penghematan biaya. Ini karena Lambda Telemetry API memungkinkan Anda untuk mengirim log langsung ke OpenSearch, yang melewati kebutuhan untuk menyimpan dan memproses data. CloudWatch

Skalabilitas dan keandalan

- Pemrosesan asinkron: Gunakan pola pemrosesan asinkron, seperti Amazon Simple Queue Service (Amazon SQS) atau Amazon Kinesis, untuk memisahkan eksekusi fungsi Lambda dari konsumsi data. OpenSearch Ini membantu menjaga daya tanggap fungsi Lambda Anda dan meningkatkan keandalan sistem secara keseluruhan.
- OpenSearch penskalaan klaster: Pantau kinerja dan pemanfaatan sumber daya OpenSearch klaster Anda, dan tingkatkan atau turunkan sesuai kebutuhan untuk menangani peningkatan volume data telemetri Lambda.

- Failover dan pemulihan bencana: Menerapkan strategi pemulihan bencana yang kuat untuk OpenSearch klaster Anda, termasuk pencadangan reguler dan kemampuan untuk memulihkan data dengan cepat jika terjadi kegagalan.

Observabilitas dan pemantauan

- Dasbor dan visualisasi: Gunakan Kibana atau alat dasbor lainnya untuk membuat dasbor dan visualisasi khusus yang memberikan wawasan tentang kinerja dan kesehatan fungsi Lambda Anda berdasarkan data telemetri di. OpenSearch
- Peringatan dan pemberitahuan: Siapkan peringatan dan notifikasi untuk secara proaktif memantau anomali, kesalahan, atau masalah kinerja di fungsi Lambda Anda. Integrasikan peringatan dan pemberitahuan ini dengan proses manajemen insiden yang ada.
- Penelusuran dan korelasi: Pastikan data telemetri Lambda Anda menyertakan informasi penelusuran yang relevan, seperti permintaan IDs atau korelasi IDs, untuk memungkinkan end-to-end pengamatan dan pemecahan masalah di seluruh aplikasi tanpa server terdistribusi Anda.

Dengan mengikuti praktik terbaik ini, Anda dapat memastikan bahwa integrasi API Telemetri Lambda Anda aman, hemat biaya, dan dapat diskalakan, serta memberikan pengamatan komprehensif untuk aplikasi tanpa server Anda. OpenSearch

Epik

Bangun dan terapkan lapisan ekstensi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kode sumbernya.	Unduh ekstensi sampel dari repositori AWS Lambda Extensions.	Pengembang aplikasi, arsitek Cloud
Arahkan ke `python-example-telemetry-opensearch-extension` folder.	Repositori AWS Lambda Ekstensi yang Anda unduh berisi banyak contoh untuk beberapa kasus penggunaan dan runtime bahasa. Arahkan ke folder python-example-telemetry-opensearch-	Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	extension untuk menggunakan ekstensi OpenSearch Python, yang mengirimkan log ke. OpenSearch	
Tambahkan izin untuk menjalankan titik akhir ekstensi.	Jalankan perintah berikut untuk membuat endpoint ekstensi dapat dieksekusi: `chmod +x python-example-telemetry-opensearch-extension/extension.py`	Pengembang aplikasi, arsitek Cloud
Instal dependensi ekstensi secara lokal.	Jalankan perintah berikut untuk menginstal dependensi lokal untuk kode Python: `pip3 install -r python-example-telemetry-opensearch-extension/requirements.txt -t ./python-example-telemetry-opensearch-extension/` Dependensi ini akan dipasang bersama dengan kode ekstensi.	Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat paket.zip untuk ekstensi untuk menerapkannya sebagai lapisan.	File ekstensi.zip harus berisi direktori root yang disebut`extensions/`, di mana ekstensi yang dapat dieksekusi berada, dan direktori root lain disebut`python-example-telemetry-opensearch-extension/`, di mana logika inti ekstensi dan dependensinya berada.
	
	Buat paket.zip untuk ekstensi:
	
	`chmod +x extensions/python-example-telemetry-opensearch-extension`
	`zip -r extension.zip extensions python-example-telemetry-opensearch-extension`
		Pengembang aplikasi, arsitek Cloud

Terapkan ekstensi sebagai lapisan Lambda.	Publikasikan layer dengan menggunakan file ekstensi.zip Anda dan perintah berikut:
	
	`aws lambda publish-layer-version \`
	`--layer-name "python-example-telemetry-opensearch-extension" \`
	`--zip-file "fileb://extension.zip"`
		Pengembang aplikasi, arsitek Cloud

Integrasikan ekstensi ke dalam fungsi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan lapisan ke fungsi Anda.	1. Masuk ke Konsol Manajemen AWS dan buka halaman Fungsi AWS Lambda konsol. 2. Pilih fungsi Anda. 3. Di bawah Lapisan, pilih Tambah lapisan. 4. Di bawah Pilih layer, pilih Custom layer sebagai sumber layer dan tambahkan layer Anda. Untuk informasi selengkapnya tentang menambahkan layer ke fungsi Lambda Anda, lihat dokumentasi Lambda.	Pengembang aplikasi, arsitek Cloud
Mengatur variabel lingkungan untuk fungsi.	Pada halaman fungsi, pilih tab Konfigurasi dan tambahkan variabel lingkungan berikut ke fungsi Anda: • `URL`— URI OpenSearch titik akhir Anda di mana log Anda akan dikirim. • `AUTH_SECRET` — ARN OpenSearch kredensil Anda disimpan di. AWS Secrets Manager Ini harus disimpan sebagai pasangan kunci-nil ai dan memiliki dua kunci: `username` dan. `password`	Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `PLATFORM_INDEX` ,`FUNCTION_INDEX` , dan `EXTENSION_INDEX` — Nama-nama indeks yang akan menyimpan data telemetri, log fungsi, dan log ekstensi Anda. Pastikan mereka mematuhi kriteria penamaan yang tepat. Jika tidak, indeks Anda tidak akan dibuat. • `DISPATCH_MIN_BATCH_SIZE` — Jumlah peristiwa log yang ingin Anda batch. Namun, ketika fungsi dimatikan, log Anda akan dikirim terlepas dari pengaturan ini.	

Tambahkan pernyataan logging dan uji fungsi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan pernyataan logging ke fungsi Anda.	Tambahkan pernyataan logging ke fungsi Anda dengan menggunakan salah satu mekanisme logging bawaan atau modul logging pilihan Anda. Berikut adalah contoh pesan logging dengan Python:	Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`print("Your Log Message Here")`
	`logger = logging.getLogger(__name__)`
	
	`logger.info("Test Info Log.")`
	`logger.error("Test Error Log.")`	

Uji fungsi Anda.	1. Di halaman fungsi, pilih tab Uji. 2. Buat acara pengujian untuk fungsi Anda dan jalankan pengujian. Untuk informasi selengkapnya, lihat Menguji fungsi Lambda di konsol dalam dokumentasi Lambda. Anda harus melihat Executing function: berhasil jika semuanya berfungsi dengan baik.	Pengembang aplikasi, arsitek Cloud

Lihat log Anda di OpenSearch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kueri indeks Anda.	Di OpenSearch, jalankan perintah berikut untuk menanyakan indeks Anda:	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`SELECT * FROM index-name` Log Anda harus ditampilkan dalam hasil kueri.	

Pemecahan masalah

Isu	Solusi
Masalah konektivitas	• Konfirmasikan bahwa fungsi Lambda Anda memiliki konektivitas jaringan yang diperlukan untuk mengakses cluster. OpenSearch Lihat dokumentasi OpenSearch Layanan untuk panduan tentang mengonfigurasi setelan VPC. • Verifikasi bahwa izin IAM yang diberikan ke fungsi Lambda Anda memungkinkannya menulis data ke cluster. OpenSearch Tinjau dokumentasi Lambda untuk informasi tentang mengelola izin IAM.
Kesalahan konsumsi data	• Periksa CloudWatch Log untuk fungsi Lambda Anda untuk mengidentifikasi kesalahan atau pengecualian yang terkait dengan integrasi API Telemetri Lambda. Lihat dokumentasi API Telemetri Lambda untuk panduan pemecahan masalah. • Verifikasi bahwa OpenSearch cluster dikonfigurasi dengan benar dan memiliki pemetaan dan pengaturan indeks yang diperlukan untuk menyerap data telemetri

Isu	Solusi
	Lambda. Konsultasikan OpenSearch dokumentasi untuk informasi lebih lanjut.

Sumber daya terkait

- Contoh Integrasi Telemetri Lambda untuk (repositori) OpenSearch GitHub
- Menambah fungsi Lambda menggunakan ekstensi Lambda (dokumentasi Lambda)
- API Telemetri Lambda (dokumentasi Lambda)
- Memperkenalkan API AWS Lambda Telemetri (AWS posting blog)
- Mengintegrasikan API AWS Lambda Telemetri dengan Prometheus dan (posting blog) OpenSearch AWS

Informasi tambahan

Mengubah struktur log

Ekstensi mengirimkan log sebagai dokumen bersarang secara OpenSearch default. Ini memungkinkan Anda melakukan kueri bersarang untuk mengambil nilai kolom individual.

Jika keluaran log default tidak memenuhi kebutuhan spesifik Anda, Anda dapat menyesuaikannya dengan memodifikasi kode sumber ekstensi Lambda yang disediakan oleh. AWS AWS mendorong pelanggan untuk menyesuaikan output agar sesuai dengan kebutuhan bisnis mereka. Untuk mengubah output log, cari `dispatch_to_opensearch` fungsi dalam `telemetry_dispatcher.py` file dalam kode sumber ekstensi dan buat perubahan yang diperlukan.

Siapkan router sel tanpa server untuk arsitektur berbasis sel

Mian Tariq dan Ioannis Lioupras, Amazon Web Services

Ringkasan

Sebagai titik masuk ke sistem aplikasi berbasis sel global, router sel bertanggung jawab untuk secara efisien menugaskan pengguna ke sel yang sesuai dan menyediakan titik akhir kepada pengguna. Router sel menangani fungsi-fungsi seperti menyimpan user-to-cell pemetaan, memantau kapasitas sel, dan meminta sel baru bila diperlukan. Penting untuk menjaga fungsionalitas router sel selama potensi gangguan.

Kerangka desain router sel dalam pola ini berfokus pada ketahanan, skalabilitas, dan pengoptimalan kinerja secara keseluruhan. Pola ini menggunakan perutean statis, di mana klien menyimpan titik akhir cache saat login awal dan berkomunikasi langsung dengan sel. Decoupling ini meningkatkan ketahanan sistem dengan membantu memastikan fungsionalitas aplikasi berbasis sel yang tidak terganggu selama gangguan router sel.

Pola ini menggunakan AWS CloudFormation template untuk menyebarkan arsitektur. Untuk detail tentang apa yang diterapkan template, atau untuk menerapkan konfigurasi yang sama dengan menggunakan Konsol Manajemen AWS, lihat bagian Informasi tambahan.

> **Important**
>
> Demonstrasi, kode, dan CloudFormation template yang disajikan dalam pola ini dimaksudkan untuk tujuan penjelasan saja. Materi yang disediakan semata-mata untuk tujuan menggambarkan pola desain dan membantu dalam pemahaman. Demo dan kode tidak siap produksi dan tidak boleh digunakan untuk kegiatan produksi langsung apa pun. Setiap upaya untuk menggunakan kode atau demo di lingkungan produksi sangat tidak dianjurkan dan merupakan risiko Anda sendiri. Kami merekomendasikan konsultasi dengan profesional yang tepat dan melakukan pengujian menyeluruh sebelum menerapkan pola ini atau komponennya dalam pengaturan produksi.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif

- Versi terbaru dari AWS Command Line Interface (AWS CLI)
- Kredensi AWS dengan izin yang diperlukan untuk membuat CloudFormation tumpukan, AWS Lambda fungsi, dan sumber daya terkait

Versi produk

- Python 3.12

Arsitektur

Diagram berikut menunjukkan desain tingkat tinggi dari router sel.

Diagram langkah-langkah melalui alur kerja berikut:

1. Pengguna menghubungi Amazon API Gateway, yang berfungsi sebagai bagian depan untuk titik akhir API router sel.
2. Amazon Cognito menangani otentikasi dan otorisasi.
3. AWS Step Functions Alur kerja terdiri dari komponen-komponen berikut:
   - Orchestrator - `Orchestrator` Penggunaan AWS Step Functions untuk membuat alur kerja, atau mesin negara. Alur kerja dipicu oleh API router sel. `Orchestrator`Mengeksekusi fungsi Lambda berdasarkan jalur sumber daya.
   - Dispatcher - Fungsi `Dispatcher` Lambda mengidentifikasi dan menetapkan satu sel statis per pengguna baru yang terdaftar. Fungsi mencari sel dengan jumlah pengguna paling sedikit, menetapkannya ke pengguna, dan mengembalikan titik akhir.
   - Mapper - `Mapper` Operasi menangani user-to-cell pemetaan dalam database `RoutingDB` Amazon DynamoDB yang dibuat oleh template. CloudFormation Ketika dipicu, `Mapper` fungsi menyediakan pengguna yang sudah ditetapkan dengan endpoint mereka.
   - Scaler - `Scaler` Fungsi melacak hunian sel dan kapasitas yang tersedia. Bila diperlukan, `Scaler` fungsi dapat mengirim permintaan melalui Amazon Simple Queue Service (Amazon SQS) ke layer Provision and Deploy untuk meminta sel baru.
   - Validator - `Validator` Fungsi memvalidasi titik akhir sel dan mendeteksi potensi masalah.
4. `RoutingDB`Menyimpan informasi dan atribut sel (titik akhir API,, status Wilayah AWS, metrik).
5. Bila kapasitas sel yang tersedia melebihi ambang batas, router sel meminta layanan penyediaan dan penerapan melalui Amazon SQS untuk membuat sel baru.

Saat sel baru dibuat, `RoutingDB` akan diperbarui dari layer Provision and Deploy. Namun, proses itu berada di luar cakupan pola ini. Untuk ikhtisar bangunan desain arsitektur berbasis sel dan detail tentang desain router sel yang digunakan dalam pola ini, lihat bagian Informasi tambahan.

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan dapat diskalakan.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi Lambda dan lainnya untuk membangun aplikasi bisnis yang penting. Layanan AWS

Alat-alat lainnya

- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Serverless-Cell-Router.

Praktik terbaik

Untuk praktik terbaik saat membangun arsitektur berbasis sel, lihat panduan Well-Architected AWS berikut:

- Mengurangi Lingkup Dampak dengan Arsitektur Berbasis Sel
- AWS Pilar Keandalan Kerangka Well-Architected REL1: 0-BP04 Gunakan arsitektur sekat untuk membatasi cakupan dampak

Epik

Siapkan file sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode contoh.	Untuk mengkloning Serverless-Cell-Router GitHub repositori ke komputer Anda, gunakan perintah berikut: `git clone https://github.com/aws-samples/Serverless-Cell-Router/`	Developer
Siapkan kredensial AWS CLI sementara.	Konfigurasikan kredensi AWS CLI dengan untuk Anda. Akun AWS Panduan ini menggunakan kredensil sementara yang disediakan oleh baris Perintah Pusat Identitas AWS IAM atau opsi akses terprogram. Ini menetapkan variabel`AWS_ACCESS_KEY_ID` ,`AWS_SECRET_ACCESS_KEY` , dan `AWS_SESSION_TOKEN` AWS	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lingkungan dengan kredensil yang sesuai untuk digunakan dengan. AWS CLI	
Buat ember S3.	Buat bucket S3 yang akan digunakan untuk menyimpan dan mengakses fungsi Serverless-Cell-Router Lambda untuk penerapan oleh template. CloudFormation Untuk membuat bucket S3, gunakan perintah berikut: `aws s3api create-bucket --bucket <bucket name> --region eu-central-1 --create-bucket-configuration LocationConstraint=eu-central-1`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file.zip.	Buat satu file.zip untuk setiap fungsi Lambda yang terletak di direktori Functions. File.zip ini akan digunakan untuk menyebarkan fungsi Lambda. Di Mac, gunakan `zip` perintah berikut:
	`zip -j mapper-scr.zip Functions/Mapper.py`
	`zip -j dispatcher-scr.zip Functions/Dispatcher.py`
	`zip -j scaler-scr.zip Functions/Scaler.py`
	`zip -j cp validator-scr.zip Functions/Validator.py`
	`zip -j dynamodbDummyData-scr.zip Functions/DynamodbDummyData.py`
		Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Salin file.zip ke bucket S3.	Untuk menyalin semua file.zip fungsi Lambda ke bucket S3, gunakan perintah berikut:
	`aws s3 cp mapper-scr.zip s3://<bucket name>`
	`aws s3 cp dispatcher-scr.zip s3://<bucket name>`
	`aws s3 cp scaler-scr.zip s3://<bucket name>`
	`aws s3 cp validator-scr.zip s3://<bucket name>`
	`aws s3 cp dynamodbDummyData-scr.zip s3://<bucket name>`
		Developer

Buat CloudFormation tumpukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan CloudFormation template.	Untuk menyebarkan CloudFormation template, jalankan AWS CLI perintah berikut:
	`aws cloudformation create-stack --stack-name serverless.cell-router \`
	`--template-body file://Serverless-Cell-Router-Stack-v10.yaml \`
		Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--capabilities CAPABILITY_IAM \ --parameters ParameterKey=LambdaFunctionMapperS3KeyParameterSCR,ParameterValue=mapper-scr.zip \ ParameterKey=LambdaFunctionDispatcherS3KeyParameterSCR,ParameterValue=dispatcher-scr.zip \ ParameterKey=LambdaFunctionScalerS3KeyParameterSCR,ParameterValue=scaler-scr.zip \ ParameterKey=LambdaFunctionAddDynamoDBDummyItemsS3KeyParameterSCR,ParameterValue=dynamodbDummyData-scr.zip \ ParameterKey=LambdaFunctionsS3BucketParameterSCR,ParameterValue=<S3 bucket storing lambda zip files> \ ParameterKey=CognitoDomain,ParameterValue=<Cognito Domain Name> \ --region <enter your aws region id, e.g. "eu-central-1">`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa kemajuan.	Masuk ke Konsol Manajemen AWS, buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation/, dan periksa kemajuan pengembangan tumpukan. Ketika statusnya`CREATE_COMPLETE` , tumpukan telah berhasil digunakan.	Developer

Menilai dan memverifikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan sel ke pengguna.	Untuk memulai`Orchestrator` , jalankan perintah curl berikut: `curl -X POST \ -H "Authorization: Bearer {User id_token}" \ https://xxxxxx.execute-api.eu-central-1.amazonaws.com/Cell_Router_Development/cells` `Orchestrator` Pemicu eksekusi `Dispatcher` fungsi. Itu`Dispatcher` , pada gilirannya, memverifikasi keberadaan pengguna. Jika pengguna ditemukan, `Dispatcher` mengembal	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ikan ID sel terkait dan titik akhir URLs. Jika pengguna tidak ditemukan, `Dispatcher` mengalokasikan sel ke pengguna dan mengirimkan ID sel ke `Scaler` fungsi untuk penilaian kapasitas sisa sel yang ditetapkan. Respons `Scaler` fungsi adalah sebagai berikut: `"cellID : cell-0002 , endPoint_1 : https://xxxxx.execute-api.eu-north-1.amazonaws.com/ , endPoint_2 : https://xxxxxxx.execute-api.eu-central-1.amazonaws.com/"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ambil sel pengguna.	Untuk menggunakan `Orchestrator` untuk menjalankan `Mapper` fungsi, jalankan perintah berikut:
	`curl -X POST \ -H "Authorization: Bearer {User id_token} " \ https://xxxxxxxx x.execute-api.eu-c entral-1.amazonaws .com/Cell_Router_D evelopment/mapper`
	`Orchestrator` Pencarian untuk sel yang ditetapka n untuk pengguna dan mengembalikan ID sel dan URLs dalam respons berikut:
	`"cellID : cell-0002 , endPoint_1 : https:// xxxxx.execute- api.eu-north-1 .amazonaws.com/ , endPoint_2 : https:// xxxxxxx.execute-api .eu-central-1.amaz onaws.com/"`
		Developer

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan sumber daya.	Untuk menghindari biaya tambahan di akun Anda, lakukan hal berikut: 1. Kosongkan bucket S3 yang Anda buat untuk fungsi Lambda. 2. Hapus bucket. 3. Hapus CloudFormation tumpukan.	Pengembang aplikasi

Sumber daya terkait

Referensi

- Stabilitas statis menggunakan Zona Ketersediaan
- Batas Isolasi Kesalahan AWS: Stabilitas statis

Video

Physalia: Arsitektur Berbasis Sel untuk Memberikan Ketersediaan Lebih Tinggi di Amazon EBS

https://www.youtube-nocookie.com/embed/6Iknq? RZMFic kontrol=0

Informasi tambahan

Tempat desain arsitektur berbasis sel

Meskipun pola ini berfokus pada router sel, penting untuk memahami seluruh lingkungan. Lingkungan disusun menjadi tiga lapisan diskrit:

- Lapisan Routing, atau Thin layer, yang berisi router sel
- Lapisan sel, terdiri dari berbagai sel
- Lapisan Penyediaan dan Penyebaran, yang menyediakan sel dan menyebarkan aplikasi

Setiap lapisan mempertahankan fungsionalitas bahkan jika terjadi gangguan yang mempengaruhi lapisan lain. Akun AWS berfungsi sebagai batas isolasi kesalahan.

Diagram berikut menunjukkan lapisan pada tingkat tinggi. Lapisan Cell dan layer Provision and Deploy berada di luar cakupan pola ini.

Untuk informasi selengkapnya tentang arsitektur berbasis sel, lihat Mengurangi Lingkup Dampak dengan Arsitektur Berbasis Sel: Perutean sel.

Pola desain cell-router

Router sel adalah komponen bersama di seluruh sel. Untuk mengurangi dampak potensial, penting bagi lapisan Routing untuk menggunakan desain yang sederhana dan dapat diskalakan secara horizontal yang setipis mungkin. Berfungsi sebagai titik masuk sistem, lapisan Routing hanya terdiri dari komponen yang diperlukan untuk secara efisien menetapkan pengguna ke sel yang sesuai. Komponen dalam lapisan ini tidak terlibat dalam pengelolaan atau pembuatan sel.

Pola ini menggunakan routing statis, yang berarti bahwa klien cache endpoint pada login awal dan kemudian membangun komunikasi langsung dengan sel. Interaksi berkala antara klien dan router sel dimulai untuk mengkonfirmasi status saat ini atau mengambil pembaruan apa pun. Decoupling yang disengaja ini memungkinkan operasi tanpa gangguan untuk pengguna yang ada jika terjadi downtime router sel, dan menyediakan fungsionalitas dan ketahanan berkelanjutan dalam sistem.

Dalam pola ini, router sel mendukung fungsionalitas berikut:

- Mengambil data sel dari database sel di lapisan Provision and Deploy dan menyimpan atau memperbarui database lokal.
- Menetapkan sel untuk setiap pengguna aplikasi terdaftar baru dengan menggunakan algoritma penugasan sel.
- Menyimpan user-to-cells pemetaan dalam database lokal.
- Memeriksa kapasitas sel selama penugasan pengguna dan memunculkan acara untuk mesin penjual otomatis ke lapisan Provision and Deploy untuk membuat sel.
- Menggunakan algoritma kriteria pembuatan sel untuk menyediakan fungsi ini.
- Menanggapi permintaan pengguna yang baru terdaftar dengan menyediakan sel statis. URLs Ini URLs akan di-cache pada klien dengan waktu untuk hidup (TTL).
- Menanggapi permintaan pengguna yang ada dari URL yang tidak valid dengan memberikan URL baru atau yang diperbarui.

Untuk lebih memahami router sel demonstrasi yang diatur oleh CloudFormation template, tinjau komponen dan langkah berikut:

1. Siapkan dan konfigurasikan kumpulan pengguna Amazon Cognito.
2. Siapkan dan konfigurasikan API Gateway API untuk router sel.
3. Buat tabel DynamoDB.
4. Buat dan konfigurasikan antrian SQS.
5. Menerapkan`Orchestrator`.
6. Menerapkan fungsi Lambda:`Dispatcher`,,`Scaler`,`Mapper`. `Validator`
7. Keledai dan verifikasi.

Anggapannya adalah bahwa lapisan Provision dan Deploy sudah ditetapkan. Detail implementasinya berada di luar cakupan artefak ini.

Karena komponen ini diatur dan dikonfigurasi oleh CloudFormation template, langkah-langkah berikut disajikan pada tingkat deskriptif dan tinggi. Asumsinya adalah Anda memiliki AWS keterampilan yang diperlukan untuk menyelesaikan pengaturan dan konfigurasi.

1. Siapkan dan konfigurasikan kumpulan pengguna Amazon Cognito

Masuk ke Konsol Manajemen AWS, dan buka konsol Amazon Cognito di. https://console.aws.amazon.com/cognito/ Siapkan dan konfigurasikan kumpulan pengguna Amazon Cognito bernama`CellRouterPool`, dengan integrasi aplikasi, UI yang dihosting, dan izin yang diperlukan.

2. Siapkan dan konfigurasikan API Gateway API untuk router sel

Buka konsol API Gateway di https://console.aws.amazon.com/apigateway/. Siapkan dan konfigurasikan API bernama`CellRouter`, menggunakan otorisasi Amazon Cognito yang terintegrasi dengan kumpulan pengguna Amazon Cognito. `CellRouterPool` Menerapkan elemen-elemen berikut:

- `CellRouter`Sumber daya API, termasuk `POST` metode
- Integrasi dengan alur kerja Step Functions diimplementasikan pada langkah 5
- Otorisasi melalui otorisasi Amazon Cognito
- Permintaan integrasi dan pemetaan respons
- Alokasi izin yang diperlukan

3. Buat tabel DynamoDB

Buka konsol DynamoDB https://console.aws.amazon.com/dynamodb/di, dan buat tabel DynamoDB standar yang disebut dengan konfigurasi berikut: `tbl_router`

- Kunci partisi - `marketId`
- Kunci sortir - `cellId`
- Mode kapasitas - Disediakan
- Point-in-time pemulihan (PITR) - Mati

Pada tab Indeks, buat indeks sekunder global yang disebut`marketId-currentCapacity-index`. Fungsi `Scaler` Lambda akan menggunakan indeks untuk melakukan pencarian efisien untuk sel dengan jumlah pengguna yang ditugaskan terendah.

Buat struktur tabel dengan atribut berikut:

- `marketId`- Eropa
- `cellId`- sel-0002
- `currentCapacity`- 2
- `endPoint_1`- <your endpoint for the first Region>
- `endPoint_2`- <your endpoint for the second Region>
- `IsHealthy`- Benar
- `maxCapacity`- 10
- `regionCode_1` – `eu-north-1`
- `regionCode_2` – `eu-central-1`
- `userIds`- <your email address>

4. Membuat dan mengkonfigurasi antrian SQS

Buka konsol Amazon SQS di https://console.aws.amazon.com/sqs/, dan buat antrean SQS standar yang disebut dikonfigurasi dengan enkripsi kunci **`CellProvisioning`** Amazon SQS.

5. Menerapkan Orkestrator

Kembangkan alur kerja Step Functions `Orchestrator` untuk berfungsi sebagai router. Alur kerja dapat dipanggil melalui API router sel. Alur kerja mengeksekusi fungsi Lambda yang ditunjuk

berdasarkan jalur sumber daya. Integrasikan fungsi langkah dengan API Gateway API untuk router selCellRouter, dan konfigurasikan izin yang diperlukan untuk menjalankan fungsi Lambda.

Diagram berikut menunjukkan alur kerja. Status pilihan memanggil salah satu fungsi Lambda. Jika fungsi Lambda berhasil, alur kerja berakhir. Jika fungsi Lambda gagal, status gagal dipanggil.

6. Menerapkan fungsi Lambda

Menerapkan DispatcherMapper,,Scaler, dan Validator fungsi. Saat Anda mengatur dan mengonfigurasi setiap fungsi dalam demonstrasi, tentukan peran untuk fungsi tersebut dan tetapkan izin yang diperlukan untuk melakukan operasi yang diperlukan pada tabel DynamoDB. tbl_router Selain itu, integrasikan setiap fungsi ke dalam alur kerjaOrchestrator.

Fungsi dispatcher

DispatcherFungsi ini bertanggung jawab untuk mengidentifikasi dan menetapkan sel statis tunggal untuk setiap pengguna terdaftar baru. Ketika pengguna baru mendaftar dengan aplikasi global, permintaan masuk ke Dispatcher fungsi. Fungsi memproses permintaan dengan menggunakan kriteria evaluasi yang telah ditentukan seperti berikut:

1. Wilayah - Pilih sel di pasar tempat pengguna berada. Misalnya, jika pengguna mengakses aplikasi global dari Eropa, pilih sel yang digunakan Wilayah AWS di Eropa.
2. Kedekatan atau latensi - Pilih sel yang paling dekat dengan pengguna Misalnya, jika pengguna mengakses aplikasi dari Belanda, fungsi tersebut mempertimbangkan sel yang menggunakan Frankfurt dan Irlandia. Keputusan mengenai sel mana yang paling dekat didasarkan pada metrik seperti latensi antara lokasi pengguna dan Wilayah sel. Untuk pola contoh ini, informasi diumpankan secara statis dari lapisan Provision and Deploy.
3. Kesehatan - Dispatcher Fungsi memeriksa apakah sel yang dipilih sehat berdasarkan keadaan sel yang disediakan (Sehat = benar atau salah).
4. Kapasitas - Distribusi pengguna didasarkan pada jumlah pengguna paling sedikit dalam logika sel, sehingga pengguna ditugaskan ke sel yang memiliki jumlah pengguna paling sedikit.

> **Note**
>
> Kriteria ini disajikan untuk menjelaskan pola contoh ini saja. Untuk implementasi router sel kehidupan nyata, Anda dapat menentukan kriteria berbasis kasus yang lebih halus dan menggunakan.

`Orchestrator`Memanggil fungsi Dispatcher untuk menetapkan pengguna ke sel. Dalam fungsi demo ini, nilai pasar adalah parameter statis yang didefinisikan sebagai`europe`.

`Dispatcher`Fungsi menilai apakah sel sudah ditetapkan ke pengguna. Jika sel sudah ditetapkan, `Dispatcher` fungsi mengembalikan titik akhir sel. Jika tidak ada sel yang ditetapkan ke pengguna, fungsi akan mencari sel dengan jumlah pengguna paling sedikit, menetapkannya ke pengguna, dan mengembalikan titik akhir. Efisiensi permintaan pencarian sel dioptimalkan dengan menggunakan indeks sekunder global.

Fungsi mapper

`Mapper`Fungsi mengawasi penyimpanan dan pemeliharaan user-to-cell pemetaan dalam database. Sel tunggal dialokasikan untuk setiap pengguna terdaftar. Setiap sel memiliki dua yang berbeda URLs —satu untuk setiap Wilayah AWS. Berfungsi sebagai titik akhir API yang dihosting di API Gateway, ini URLs berfungsi sebagai titik masuk ke aplikasi global.

Ketika `Mapper` fungsi menerima permintaan dari aplikasi klien, ia menjalankan query pada `tbl_router` tabel DynamoDB untuk mengambil user-to-cell pemetaan yang terkait dengan ID email yang disediakan. Jika menemukan sel yang ditetapkan, `Mapper` fungsi segera menyediakan dua URLs sel. `Mapper`Fungsi ini juga secara aktif memonitor perubahan pada sel URLs, dan memulai pemberitahuan atau pembaruan ke pengaturan pengguna.

Fungsi scaler

`Scaler`Fungsi ini mengelola kapasitas sisa sel. Untuk setiap permintaan pendaftaran pengguna baru, `Scaler` fungsi menilai kapasitas sel yang tersedia dari `Dispatcher` fungsi yang ditetapkan untuk pengguna. Jika sel telah mencapai batas yang telah ditentukan sesuai dengan kriteria evaluasi yang ditentukan, fungsi akan memulai permintaan melalui antrian Amazon SQS ke lapisan Penyediaan dan Penerapan, meminta penyediaan dan penerapan sel baru. Penskalaan sel dapat dijalankan berdasarkan serangkaian kriteria evaluasi seperti berikut:

1. Pengguna maksimum - Setiap sel dapat memiliki 500 jumlah maksimum pengguna.

2. Kapasitas buffer - Kapasitas buffer setiap sel adalah 20 persen, yang berarti bahwa setiap sel dapat ditetapkan ke 400 pengguna setiap saat. Kapasitas buffer 20 persen sisanya dicadangkan untuk kasus penggunaan di masa mendatang dan penanganan skenario yang tidak terduga (misalnya, ketika layanan pembuatan dan penyediaan sel tidak tersedia).
3. Pembuatan sel - Segera setelah sel yang ada mencapai 70 persen kapasitas, permintaan dipicu untuk membuat sel tambahan.

> **Note**
>
> Kriteria ini disajikan untuk menjelaskan pola contoh ini saja. Untuk implementasi router sel kehidupan nyata, Anda dapat menentukan kriteria berbasis kasus yang lebih halus dan menggunakan.

`Scaler`Kode demonstrasi dijalankan oleh `Orchestrator` setelah `Dispatcher` berhasil menetapkan sel ke pengguna yang baru terdaftar. Setelah menerima ID sel dari `ScalerDispatcher`, mengevaluasi apakah sel yang ditunjuk memiliki kapasitas yang memadai untuk mengakomodasi pengguna tambahan, berdasarkan kriteria evaluasi yang telah ditentukan. Jika kapasitas sel tidak mencukupi, `Scaler` fungsi mengirimkan pesan ke layanan Amazon SQS. Pesan ini diambil oleh layanan dalam layer Provision and Deploy, memulai penyediaan sel baru.

Fungsi validator

`Validator`Fungsi mengidentifikasi dan menyelesaikan masalah yang berkaitan dengan akses sel. Saat pengguna masuk ke aplikasi global, aplikasi mengambil sel URLs dari pengaturan profil pengguna dan merutekan permintaan pengguna ke salah satu dari dua Wilayah yang ditetapkan di dalam sel. Jika URLs tidak dapat diakses, aplikasi dapat mengirimkan permintaan URL validasi ke router sel. Router sel `Orchestrator` memanggil file. `Validator` `Validator`Memulai proses validasi. Validasi mungkin mencakup, di antara pemeriksaan lainnya, yang berikut:

- Sel referensi silang URLs dalam permintaan dengan URLs disimpan dalam database untuk mengidentifikasi dan memproses pembaruan potensial
- Menjalankan pemeriksaan kesehatan mendalam (misalnya, `HTTP GET` permintaan ke titik akhir sel)

Kesimpulannya, `Validator` fungsi memberikan tanggapan terhadap permintaan aplikasi klien, memberikan status validasi bersama dengan langkah-langkah remediasi yang diperlukan.

`Validator`Ini dirancang untuk meningkatkan pengalaman pengguna. Pertimbangkan skenario di mana pengguna tertentu mengalami kesulitan mengakses aplikasi global karena insiden menyebabkan sel untuk sementara tidak tersedia. Alih-alih menyajikan kesalahan umum, `Validator` fungsi ini dapat memberikan langkah-langkah remediasi instruktif. Langkah-langkah ini mungkin termasuk tindakan berikut:

- Beri tahu pengguna tentang insiden tersebut.
- Berikan perkiraan waktu tunggu sebelum ketersediaan layanan.
- Berikan nomor kontak dukungan untuk mendapatkan informasi tambahan.

Kode demo untuk `Validator` fungsi memverifikasi bahwa sel yang disediakan pengguna URLs dalam permintaan cocok dengan catatan yang disimpan dalam tabel. `tbl_router` `Validator`Fungsi ini juga memeriksa apakah sel-sel sehat.

Siapkan akses pribadi ke bucket Amazon S3 melalui titik akhir VPC

Martin Maritsch, Nicolas Jacob Baer, Gabriel Rodriguez Garcia, Shukhrat Khodjaev, Mohan Gowda Purushothama, dan Joaquin Rinaudo, Amazon Web Services

Ringkasan

Di Amazon Simple Storage Service (Amazon S3), URLs presigned memungkinkan Anda berbagi file dengan ukuran arbitrer dengan pengguna target. Secara default, Amazon S3 presigned dapat URLs diakses dari internet dalam jangka waktu kedaluwarsa, yang membuatnya nyaman digunakan. Namun, lingkungan perusahaan seringkali memerlukan akses ke Amazon S3 yang ditetapkan URLs untuk dibatasi hanya pada jaringan pribadi.

Pola ini menyajikan solusi tanpa server untuk berinteraksi secara aman dengan objek S3 dengan menggunakan presigned URLs dari jaringan pribadi tanpa traversal internet. Dalam arsitektur, pengguna mengakses Application Load Balancer melalui nama domain internal. Lalu lintas dirutekan secara internal melalui Amazon API Gateway dan titik akhir virtual private cloud (VPC) untuk bucket S3. AWS Lambda Fungsi ini menghasilkan presigned URLs untuk unduhan file melalui titik akhir VPC pribadi, yang membantu meningkatkan keamanan dan privasi untuk data sensitif.

Prasyarat dan batasan

Prasyarat

- VPC yang menyertakan subnet yang digunakan dalam Akun AWS yang terhubung ke jaringan perusahaan (misalnya, melalui). AWS Direct Connect

Batasan

- Bucket S3 harus memiliki nama yang sama dengan domain, jadi sebaiknya Anda memeriksa aturan penamaan bucket Amazon S3.
- Contoh arsitektur ini tidak menyertakan fitur pemantauan untuk infrastruktur yang digunakan. Jika kasus penggunaan Anda memerlukan pemantauan, pertimbangkan untuk menambahkan layanan AWS pemantauan.
- Contoh arsitektur ini tidak menyertakan validasi masukan. Jika kasus penggunaan Anda memerlukan validasi input dan tingkat keamanan yang lebih tinggi, pertimbangkan AWS WAF untuk menggunakan untuk melindungi API Anda.

- Contoh arsitektur ini tidak menyertakan akses logging dengan Application Load Balancer. Jika kasus penggunaan Anda memerlukan pencatatan akses, pertimbangkan untuk mengaktifkan log akses penyeimbang beban.

Versi

- Python versi 3.11 atau yang lebih baru
- Terraform versi 1.6 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

Layanan AWS berikut digunakan dalam tumpukan teknologi target:

- Amazon S3 adalah layanan penyimpanan inti yang digunakan untuk mengunggah, mengunduh, dan menyimpan file dengan aman.
- Amazon API Gateway mengekspos sumber daya dan titik akhir untuk berinteraksi dengan bucket S3. Layanan ini berperan dalam menghasilkan presigned URLs untuk mengunduh atau mengunggah data.
- AWS Lambdamenghasilkan presigned URLs untuk men-download file dari Amazon S3. Fungsi Lambda dipanggil oleh API Gateway.
- Amazon VPC menyebarkan sumber daya dalam VPC untuk menyediakan isolasi jaringan. VPC mencakup subnet dan tabel routing untuk mengontrol arus lalu lintas.
- Application Load Balancer merutekan lalu lintas masuk ke API Gateway atau ke endpoint VPC bucket S3. Ini memungkinkan pengguna dari jaringan perusahaan untuk mengakses sumber daya secara internal.
- Titik akhir VPC untuk Amazon S3 memungkinkan komunikasi langsung dan pribadi antara sumber daya di VPC dan Amazon S3 tanpa melintasi internet publik.
- AWS Identity and Access Management (IAM) mengontrol akses ke AWS sumber daya. Izin disiapkan untuk memastikan interaksi yang aman dengan API dan layanan lainnya.

Arsitektur target

Diagram ini menggambarkan hal sebagai berikut:

1. Pengguna dari jaringan perusahaan dapat mengakses Application Load Balancer melalui nama domain internal. Kami berasumsi bahwa ada koneksi antara jaringan perusahaan dan subnet intranet di Akun AWS (misalnya, melalui koneksi). Direct Connect
2. Application Load Balancer merutekan lalu lintas masuk ke API Gateway untuk menghasilkan presigned URLs untuk mengunduh atau mengunggah data ke Amazon S3, atau ke titik akhir VPC bucket S3. Dalam kedua skenario, permintaan dirutekan secara internal dan tidak perlu melintasi internet.
3. API Gateway mengekspos sumber daya dan titik akhir untuk berinteraksi dengan bucket S3. Dalam contoh ini, kami menyediakan titik akhir untuk mengunduh file dari bucket S3, tetapi ini dapat diperluas untuk menyediakan fungsionalitas unggah juga.
4. Fungsi Lambda menghasilkan URL yang telah ditetapkan sebelumnya untuk mengunduh file dari Amazon S3 dengan menggunakan nama domain Application Load Balancer, bukan domain Amazon S3 publik.
5. Pengguna menerima URL yang telah ditetapkan sebelumnya dan menggunakannya untuk mengunduh file dari Amazon S3 dengan menggunakan Application Load Balancer. Penyeimbang beban menyertakan rute default untuk mengirim lalu lintas yang tidak dimaksudkan untuk API menuju titik akhir VPC bucket S3.
6. Titik akhir VPC merutekan URL yang telah ditetapkan sebelumnya dengan nama domain kustom ke bucket S3. Bucket S3 harus memiliki nama yang sama dengan domain.

Otomatisasi dan skala

Pola ini menggunakan Terraform untuk menyebarkan infrastruktur dari repositori kode ke file. Akun AWS

Alat

Alat

- Python adalah bahasa pemrograman komputer tujuan umum.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi dengan AWS layanan melalui perintah di shell baris perintah Anda.

Repositori kode

Kode untuk pola ini tersedia dalam GitHub repositori di private-s3-vpce. https://github.com/aws-samples/

Praktik terbaik

Contoh arsitektur untuk pola ini menggunakan izin IAM untuk mengontrol akses ke API. Siapa pun yang memiliki kredensil IAM yang valid dapat memanggil API. Jika kasus penggunaan Anda memerlukan model otorisasi yang lebih kompleks, Anda mungkin ingin menggunakan mekanisme kontrol akses yang berbeda.

Epik

Menerapkan solusi dalam Akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan AWS kredensil.	Tinjau AWS kredensi Anda dan akses Anda ke akun Anda. Untuk petunjuk, lihat Pengaturan konfigurasi dan file kredensyal dalam AWS CLI dokumentasi.	AWS DevOps, AWS Umum
Kloning repositori.	Kloning GitHub repositori yang disediakan dengan pola ini: `git clone https://github.com/aws-samples/private-s3-vpce`	AWS DevOps, AWS Umum
Konfigurasikan variabel.	1. Di komputer Anda, di GitHub repositori, buka folder`terraform` : `cd terraform` 2. Buka `example.tfvars` file dan sesuaikan	AWS DevOps, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	parameter sesuai dengan kebutuhan Anda.	
Menyebarkan solusi.	1. Di `terraform` folder, jalankan Terraform dan teruskan variabel yang Anda sesuaikan: `terraform apply -var-file="example.tfvars"` 2. Konfirmasikan bahwa sumber daya yang ditampilkan dalam diagram arsitektur berhasil digunakan.	AWS DevOps, AWS Umum

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file uji.	Unggah file ke Amazon S3 untuk membuat skenario pengujian untuk unduhan file. Anda dapat menggunakan konsol Amazon S3 atau perintah berikut: AWS CLI `aws s3 cp /path/to/testfile s3://your-bucket-name/testfile`	AWS DevOps, AWS Umum
Uji fungsionalitas URL yang telah ditetapkan sebelumnya.	1. Kirim permintaan ke Application Load Balancer untuk membuat URL presigned untuk file	AWS DevOps, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengujian dengan menggunakan awscurl: `awscurl https://your-domain-name/api/get_url?key=testfile` Langkah ini membuat tanda tangan yang valid dari kredensil Anda, yang akan divalidasi oleh API Gateway. 2. Parse link dari respon yang Anda terima dari langkah sebelumnya, dan buka URL presigned untuk men-download file.	
Bersihkan.	Pastikan untuk menghapus sumber daya saat tidak lagi diperlukan: `terraform destroy`	AWS DevOps, AWS Umum

Pemecahan masalah

Isu	Solusi
Nama kunci objek S3 dengan karakter khusus seperti tanda angka (#) memecah parameter URL dan menyebabkan kesalahan.	Encode parameter URL dengan benar, dan pastikan bahwa nama kunci objek S3 mengikuti pedoman Amazon S3.

Sumber daya terkait

Amazon S3:

- Berbagi objek dengan presigned URLs
- Mengontrol akses dari titik akhir VPC dengan kebijakan bucket

Amazon API Gateway:

- Menggunakan kebijakan titik akhir VPC secara pribadi APIs di API Gateway

Application Load Balancer:

- Hosting Situs Web Statis HTTPS Internal dengan ALB, S3, dan PrivateLink (AWS posting blog)

Memecahkan masalah status AWS Step Functions dengan menggunakan Amazon Bedrock

Aniket Kurzadkar dan Sangam Kushwaha, Amazon Web Services

Ringkasan

AWS Step Functions Kemampuan penanganan kesalahan dapat membantu Anda melihat kesalahan yang terjadi selama status dalam alur kerja, tetapi masih bisa menjadi tantangan untuk menemukan akar penyebab kesalahan dan men-debugnya. Pola ini membahas tantangan tersebut dan menunjukkan bagaimana Amazon Bedrock dapat membantu Anda mengatasi kesalahan yang terjadi selama status di Step Functions.

Step Functions menyediakan orkestrasi alur kerja, sehingga memudahkan pengembang untuk mengotomatisasi proses. Step Functions juga menyediakan fungsionalitas penanganan kesalahan yang memberikan manfaat sebagai berikut:

- Pengembang dapat membuat aplikasi yang lebih tangguh yang tidak gagal sepenuhnya ketika terjadi kesalahan.
- Alur kerja dapat mencakup logika bersyarat untuk menangani berbagai jenis kesalahan secara berbeda.
- Sistem dapat secara otomatis mencoba kembali operasi yang gagal, mungkin dengan backoff eksponensial.
- Jalur eksekusi alternatif dapat didefinisikan untuk skenario kesalahan, memungkinkan alur kerja beradaptasi dan melanjutkan pemrosesan.

Ketika terjadi kesalahan dalam alur kerja Step Functions, pola ini menunjukkan bagaimana pesan kesalahan dan konteks dapat dikirim ke model dasar (FM) seperti Claude 3 yang didukung oleh Step Functions. FM dapat menganalisis kesalahan, mengkategorikannya, dan menyarankan langkah-langkah perbaikan potensial.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Pemahaman dasar AWS Step Functions dan alur kerja

- Konektivitas Amazon Bedrock API

Batasan

- Anda dapat menggunakan pendekatan pola ini untuk berbagai Layanan AWS. Namun, hasilnya mungkin bervariasi sesuai dengan prompt yang dibuat oleh AWS Lambda yang kemudian dievaluasi oleh Amazon Bedrock.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat layanan AWS menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Diagram menunjukkan alur kerja otomatis untuk penanganan kesalahan dan pemberitahuan di mesin status Step Functions:

1. Pengembang memulai eksekusi mesin negara.
2. Mesin status Step Functions mulai memproses statusnya. Ada dua kemungkinan hasil:
   - (a) Jika semua status berhasil dijalankan, alur kerja dilanjutkan langsung ke Amazon SNS untuk pemberitahuan keberhasilan email.
   - (b) Jika ada status yang gagal, alur kerja berpindah ke kesalahan penanganan fungsi Lambda.
3. Jika terjadi kesalahan, berikut ini terjadi:
   - (a) Fungsi Lambda (penangan kesalahan) dipicu. Fungsi Lambda mengekstrak pesan kesalahan dari data peristiwa yang diteruskan oleh mesin status Step Functions kepadanya. Kemudian fungsi Lambda menyiapkan prompt berdasarkan pesan kesalahan ini dan mengirimkan prompt ke Amazon Bedrock. Prompt meminta solusi dan saran terkait dengan kesalahan spesifik yang ditemui.
   - (b) Amazon Bedrock, yang menampung model AI generatif, memproses prompt input. (Pola ini menggunakan model pondasi Anthropic Claude 3 (FM), yang merupakan salah satu dari banyak yang didukung FMs Amazon Bedrock.) Model AI menganalisis konteks kesalahan. Kemudian model menghasilkan respons yang dapat mencakup penjelasan mengapa kesalahan terjadi, solusi potensial untuk menyelesaikan kesalahan, dan saran untuk menghindari membuat kesalahan yang sama di masa depan.

Amazon Bedrock mengembalikan respons yang dihasilkan AI ke fungsi Lambda. Fungsi Lambda memproses respons, berpotensi memformatnya atau mengekstrak informasi kunci. Kemudian fungsi Lambda mengirimkan respons ke output mesin status.

4. Setelah penanganan kesalahan atau eksekusi berhasil, alur kerja diakhiri dengan memicu Amazon SNS untuk mengirim pemberitahuan email.

Alat

Layanan AWS

- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.

Praktik terbaik

- Mengingat bahwa Amazon Bedrock adalah model AI generatif yang belajar dari data terlatih, ia juga menggunakan data tersebut untuk melatih dan menghasilkan konteks. Sebagai praktik terbaik, sembunyikan informasi pribadi apa pun yang dapat menyebabkan masalah kebocoran data.
- Meskipun AI generatif dapat memberikan wawasan yang berharga, keputusan penanganan kesalahan kritis harus tetap melibatkan pengawasan manusia, terutama di lingkungan produksi.

Epik

Buat mesin status untuk alur kerja Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat mesin negara.	Untuk membuat mesin status yang sesuai dengan alur kerja Anda, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan buka AWS Step Functions konsol. 2. Dari panel navigasi kiri, pilih mesin State. 3. Pilih Buat mesin status. 4. Pilih template sesuai dengan kasus penggunaan Anda, atau pilih Kosong untuk membuat template sesuai dengan kebutuhan Anda.	AWS DevOps

Buat fungsi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda.	Untuk membuat fungsi Lambda, lakukan hal berikut: 1. Di Konsol Manajemen AWS, navigasikan ke AWS Lambda konsol.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di panel navigasi kiri, pilih Functions dan kemudian pilih Create function. 3. Pada halaman Create function, pilih dari opsi untuk membuat fungsi. Kemudian, masukkan nama di Nama fungsi dan pilih bahasa yang sesuai dari daftar dropdown di Runtime. 4. Pilih Buat fungsi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan logika yang diperlukan dalam kode Lambda.	• Untuk menyambung ke Amazon Bedrock API dengan menggunakan AWS SDK for Python (Boto3), gunakan kode berikut.
	
	Kode ini menyiapkan klien untuk Amazon Bedrock, menyiapkan parameter yang diperlukan, dan kemudian mengirimkan permintaan ke model Claude 3 dengan prompt yang ditentukan.
	
	Pola ini memanggil model Claude 3. Untuk informasi selengkapnya tentang semua model foundation yang didukung termasuk model terkait IDs, lihat Model foundation yang didukung di Amazon Bedrock di dokumentasi Amazon Bedrock.
	
	`client = boto3.client( service_name="bedrock-runtime", region_name="selected-region" )`
	
	`# Invoke Claude 3 with the text prompt model_id = "your-model-id" # Select your`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>Model ID, Based on the Model Id, Change the body format
	
	   try:
	       response = client.invoke_model(
	           modelId=model_id,
	           body=json.dumps(
	               {
	
	"anthropic_version": "bedrock-2023-05-31",
	
	"max_tokens": 1024,
	
	"messages": [
	
	{
	
	   "role": "user",
	
	         "content": [{"type": "text", "text": prompt}],
	
	}
	                   ],
	               }
	           ),
	       )</pre>
	• (Opsional) Ganti akun Akun AWS IDs dengan placeholder. IDs Untuk tujuan keamanan, pendekatan ini dapat berguna untuk membersihkan log, pesan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	kesalahan, atau output lain yang mungkin berisi informasi akun sensitif.
	
	Kode berikut akan menemukan kejadian angka 12 digit yang dilampirkan dalam titik dua (yang merupakan format di Akun AWS IDs Amazon Resource Names (ARNs) dan beberapa AWS pengidentifikasi lainnya) dan menggantinya dengan ID akun placehold er. `":123456789012:"`
	
	`def replace_a ccount_id(input_st ring):`
	
	`# Use a regular expression to find the AWS account ID pattern`
	`account_i d_pattern = r'(:\d{12 }:)'`
	
	`# Replace the matched pattern with ":123456789012:"`
	`modified_string = re.sub(account_id_ pattern, ":1234567 89012:", input_str ing)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`return modified_string`	

Integrasikan Step Functions dengan Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Lambda untuk menangani kesalahan di Step Functions.	Untuk mengatur Step Functions untuk menangani kesalahan tanpa mengganggu alur kerja, lakukan hal berikut: 1. Di konsol Step Functions, navigasikan ke mesin status yang Anda buat sebelumnya. 2. Pilih Edit, lalu pilih layanan yang ingin Anda atur penanganan kesalahan dan pilih Penanganan Kesalahan. 3. Pilih Tambahkan penangkap baru, dan untuk status Fallback, pilih Lambda dan kemudian pilih fungsi Lambda yang Anda buat sebelumnya. Untuk informasi selengkapnya, lihat Menangkap kesalahan dalam dokumentasi Step Functions.	AWS DevOps

Pemecahan masalah

Isu	Solusi
Lambda tidak dapat mengakses Amazon Bedrock API (Tidak diizinkan untuk melakukan)	Kesalahan ini terjadi ketika peran Lambda tidak memiliki izin untuk mengakses Amazon Bedrock API. Untuk mengatasi masalah ini, tambahkan `AmazonBedrockFullAccess` kebijakan untuk peran Lambda. Untuk informasi selengkapnya, lihat AmazonBedrockFullAccessdi Panduan Referensi Kebijakan AWS Terkelola.
Kesalahan batas waktu Lambda	Terkadang mungkin diperlukan lebih dari 30 detik untuk menghasilkan respons dan mengirimkannya kembali, tergantung pada prompt. Untuk mengatasi masalah ini, tingkatkan waktu konfigurasi. Untuk informasi selengkapnya, lihat Mengonfigurasi batas waktu fungsi Lambda di Panduan Pengembang.AWS Lambda

Sumber daya terkait

- Amazon Bedrock
- Akses Amazon Bedrock API
- Buat fungsi Lambda pertama Anda
- Mengembangkan alur kerja dengan Step Functions
- AWS Step Functions

Lebih banyak pola

- Akses, kueri, dan gabungkan tabel Amazon DynamoDB menggunakan Athena
- Otomatiskan CodeGuru ulasan Amazon untuk aplikasi AWS CDK Python dengan menggunakan Tindakan GitHub
- Mengotomatiskan penilaian sumber daya AWS
- Otomatiskan penerapan aplikasi bersarang menggunakan AWS SAM
- Otomatiskan penerapan Rantai Pasokan AWS data lake dalam pengaturan multi-repositori dengan menggunakan GitHub Actions, Artifactory, dan Terraform
- Otomatiskan replikasi instans Amazon RDS di seluruh Akun AWS
- Secara otomatis mengarsipkan item ke Amazon S3 menggunakan DynamoDB TTL
- Secara otomatis mendeteksi perubahan dan memulai CodePipeline jaringan pipa yang berbeda untuk monorepo di CodeCommit
- Membangun arsitektur tanpa server multi-tenant di Amazon Service OpenSearch
- Membangun penampil file mainframe tingkat lanjut di AWS Cloud
- Hitung value at risk (VaR) dengan menggunakan layanan AWS
- Salin produk AWS Service Catalog di berbagai akun AWS dan Wilayah AWS
- Buat pipeline CI dinamis untuk proyek Java dan Python secara otomatis
- Menguraikan monolit menjadi layanan mikro dengan menggunakan CQRS dan sumber acara
- Menerapkan aplikasi satu halaman berbasis React ke Amazon S3 dan CloudFront
- Menerapkan API Amazon API Gateway di situs web internal menggunakan endpoint pribadi dan Application Load Balancer
- Menerapkan dan mengelola data lake tanpa server di AWS Cloud dengan menggunakan infrastruktur sebagai kode
- Terapkan kasus penggunaan RAG AWS dengan menggunakan Terraform dan Amazon Bedrock
- Kembangkan asisten berbasis obrolan otomatis sepenuhnya dengan menggunakan agen dan basis pengetahuan Amazon Bedrock
- Kembangkan asisten berbasis obrolan AI generatif tingkat lanjut dengan menggunakan RAG dan prompt ReAct
- Buat kebijakan IAM secara dinamis dengan IAM Access Analyzer dengan menggunakan Step Functions

- Sematkan komponen visual Amazon Quick Sight ke dalam aplikasi web dengan menggunakan otomatisasi Amazon Cognito dan IAc
- Pastikan pencatatan EMR Amazon ke Amazon S3 diaktifkan saat peluncuran
- Perkirakan biaya tabel DynamoDB untuk kapasitas sesuai permintaan
- Buat rekomendasi yang dipersonalisasi dan diberi peringkat ulang menggunakan Amazon Personalize
- Hasilkan data pengujian menggunakan pekerjaan AWS Glue dan Python
- Menerapkan SHA1 hashing untuk data PII saat bermigrasi dari SQL Server ke PostgreSQL
- Menerapkan pola saga tanpa server dengan menggunakan AWS Step Functions
- Tingkatkan kinerja operasional dengan mengaktifkan Amazon DevOps Guru di beberapa Wilayah AWS, akun, dan OUs dengan AWS CDK
- Luncurkan CodeBuild proyek di seluruh akun AWS menggunakan Step Functions dan fungsi proxy Lambda
- Migrasikan beban kerja Apache Cassandra ke Amazon Keyspaces dengan menggunakan AWS Glue
- Pantau penggunaan Gambar Mesin Amazon bersama di beberapa Akun AWS
- Optimalkan penerapan tanpa server multi-akun dengan menggunakan alur kerja dan Tindakan AWS CDK GitHub
- Mengatur pipeline ETL dengan validasi, transformasi, dan partisi menggunakan AWS Step Functions
- Kueri tabel Amazon DynamoDB dengan SQL dengan menggunakan Amazon Athena
- Jalankan beban kerja yang digerakkan oleh peristiwa dan terjadwal dalam skala besar dengan AWS Fargate
- Kirim atribut khusus ke Amazon Cognito dan masukkan ke dalam token
- Sajikan konten statis dalam bucket Amazon S3 melalui VPC dengan menggunakan Amazon CloudFront
- Merampingkan pengembangan dan penerapan bot Amazon Lex dengan menggunakan alur kerja otomatis
- Struktur proyek Python dalam arsitektur heksagonal menggunakan AWS Lambda
- Terjemahkan bahasa alami ke dalam kueri DSL untuk OpenSearch dan kueri Elasticsearch
- Membongkar data dari klaster Amazon Redshift di seluruh akun ke Amazon S3

- Koordinasikan ketergantungan sumber daya dan eksekusi tugas dengan menggunakan konstruksi AWS Fargate WaitCondition kait
- Gunakan agen Amazon Bedrock untuk mengotomatiskan pembuatan kontrol entri akses di Amazon EKS melalui petunjuk berbasis teks

Jaringan

Topik

- Mengotomatiskan pengaturan peering Antar wilayah dengan AWS Transit Gateway
- Memusatkan konektivitas jaringan menggunakan AWS Transit Gateway
- Konfigurasikan enkripsi HTTPS untuk Oracle JD Edwards EnterpriseOne di Oracle WebLogic dengan menggunakan Application Load Balancer
- Connect ke data Application Migration Service dan kontrol pesawat melalui jaringan pribadi
- Buat objek Infoblox menggunakan sumber daya CloudFormation khusus AWS dan Amazon SNS
- Buat arsitektur IPAM Multi-wilayah yang hierarkis dengan menggunakan Terraform AWS
- Kustomisasi CloudWatch peringatan Amazon untuk AWS Network Firewall
- Menerapkan sumber daya di AWS Wavelength Zona dengan menggunakan Terraform
- Migrasikan data DNS secara massal ke zona host pribadi Amazon Route 53
- Ubah header HTTP saat Anda bermigrasi dari F5 ke Application Load Balancer di AWS
- Buat laporan temuan Network Access Analyzer untuk akses internet masuk dalam beberapa Akun AWS
- Mengatur resolusi DNS untuk jaringan hybrid di lingkungan multi-akun AWS
- Verifikasi bahwa penyeimbang beban ELB memerlukan penghentian TLS
- Lihat log dan metrik AWS Network Firewall dengan menggunakan Splunk
- Lebih banyak pola

Mengotomatiskan pengaturan peering Antar wilayah dengan AWS Transit Gateway

Ram Kandaswamy, Amazon Web Services

Ringkasan

AWS Transit Gateway menghubungkan cloud pribadi virtual (VPCs) dan jaringan lokal melalui hub pusat. Lalu lintas Transit Gateway selalu berada di tulang punggung Amazon Web Services (AWS) global dan tidak melintasi internet publik, yang mengurangi vektor ancaman, seperti eksploitasi umum dan serangan penolakan layanan terdistribusi. DDo

Jika Anda perlu berkomunikasi antara dua atau lebih Wilayah AWS, Anda dapat menggunakan peering Gateway Transit Antar wilayah untuk membuat koneksi peering antara gateway transit di Wilayah yang berbeda. Namun, mengonfigurasi peering Antar wilayah secara manual dengan Transit Gateway dapat menjadi proses yang memakan waktu yang memiliki beberapa langkah. Pola ini menyediakan proses otomatis untuk menghapus langkah-langkah manual ini dengan menggunakan kode untuk melakukan peering. Anda dapat menggunakan pendekatan ini jika Anda harus berulang kali mengonfigurasi beberapa akun Wilayah dan AWS selama penyiapan organisasi Multi-wilayah.

Pola ini menggunakan CloudFormation tumpukan AWS yang mencakup alur kerja AWS Step Functions, fungsi AWS Lambda, peran AWS Identity and Access Management (IAM), dan grup log di Amazon Logs. CloudWatch Anda kemudian dapat memulai eksekusi Step Functions dan membuat koneksi peering Antar wilayah untuk gateway transit Anda. Untuk mengatur peering Antar wilayah secara manual, lihat Peer di Wilayah AWS VPCs yang berbeda menggunakan AWS Transit Gateway.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Bucket Amazon Simple Storage Service (Amazon S3) yang sudah ada.
- Gateway transit, dibuat dan dikonfigurasi di Wilayah pemohon dan Wilayah akseptor. Wilayah pemohon adalah tempat permintaan peering berasal dan Wilayah akseptor menerima permintaan peering. Untuk informasi selengkapnya tentang hal ini, lihat Membuat dan menerima koneksi peering VPC di dokumentasi VPC Amazon.

- VPCs, diinstal dan dikonfigurasi di Wilayah akseptor dan pemohon. Untuk langkah-langkah membuat VPC, lihat Membuat VPC dari Memulai dengan Amazon VPC di dokumentasi Amazon VPC.
- VPCs Harus menggunakan `addToTransitGateway` tag dan `true` nilai.
- Grup keamanan dan daftar kontrol akses jaringan (ACLs) untuk Anda VPCs, dikonfigurasi sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya tentang ini, lihat Grup keamanan untuk VPC dan Jaringan Anda ACLs di dokumentasi Amazon VPC.

Wilayah dan batasan AWS

- Hanya Wilayah AWS tertentu yang mendukung peering Antar wilayah. Untuk daftar lengkap Wilayah yang mendukung peering Antar wilayah, lihat AWS Transit Gateway. FAQs
- Dalam kode sampel terlampir, Wilayah pemohon diasumsikan`us-east-2`, dan Wilayah akseptor diasumsikan. `us-west-2` Jika Anda ingin mengkonfigurasi Wilayah yang berbeda, Anda harus mengedit nilai-nilai ini di semua file Python. Untuk menerapkan pengaturan yang lebih kompleks yang melibatkan lebih dari dua Wilayah, Anda dapat mengubah Fungsi Langkah untuk meneruskan Wilayah sebagai parameter ke fungsi Lambda dan menjalankan fungsi untuk setiap kombinasi.

Arsitektur

Diagram menunjukkan alur kerja dengan langkah-langkah berikut:

1. Pengguna membuat CloudFormation tumpukan AWS.
2. AWS CloudFormation membuat mesin status Step Functions yang menggunakan fungsi Lambda. Untuk informasi selengkapnya tentang hal ini, lihat Membuat mesin status Step Functions yang menggunakan Lambda dalam dokumentasi AWS Step Functions.
3. Step Functions memanggil fungsi Lambda untuk mengintip.
4. Fungsi Lambda menciptakan koneksi peering antara gateway transit.
5. Step Functions memanggil fungsi Lambda untuk modifikasi tabel rute.
6. Fungsi Lambda memodifikasi tabel rute dengan menambahkan blok Classless Inter-Domain Routing (CIDR) dari. VPCs

Alur kerja Step Functions

Diagram menunjukkan alur kerja Step Functions berikut:

1. Alur kerja Step Functions memanggil fungsi Lambda untuk peering gateway transit.
2. Ada panggilan timer untuk menunggu satu menit.
3. Status peering diambil dan dikirim ke blok kondisi. Blok bertanggung jawab atas perulangan.
4. Jika kondisi keberhasilan tidak terpenuhi, alur kerja dikodekan untuk memasuki tahap timer.
5. Jika kondisi keberhasilan terpenuhi, fungsi Lambda dipanggil untuk memodifikasi tabel rute. Setelah panggilan ini, alur kerja Step Functions berakhir.

Alat

- AWS CloudFormation — AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda.
- Amazon CloudWatch Logs — CloudWatch Log membantu Anda memusatkan log dari semua sistem, aplikasi, dan layanan AWS yang Anda gunakan.
- AWS Identity and Access Management (IAM) — IAM adalah layanan web untuk mengontrol akses ke layanan AWS secara aman.
- AWS Lambda — Lambda menjalankan kode Anda pada infrastruktur komputasi ketersediaan tinggi dan menjalankan semua administrasi sumber daya komputasi.
- AWS Step Functions — Step Functions memudahkan untuk mengoordinasikan komponen aplikasi terdistribusi sebagai serangkaian langkah dalam alur kerja visual.

Epik

Mengotomatiskan mengintip

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file terlampir ke bucket S3 Anda.	Masuk ke AWS Management Console, buka konsol Amazon S3, lalu unggah file, `modify-transit-gateway-routes.zip peer-transit-`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`gateway.zip` , dan `get-transit-gateway-peering-status.zip` file (terlampir) ke bucket S3 Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan AWS.	Jalankan perintah berikut untuk membuat CloudFormation tumpukan AWS menggunakan `transit-gateway-peering.json` file (terlampir): `aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json` CloudFormation Tumpukan AWS membuat alur kerja Step Functions, fungsi Lambda, peran IAM, dan grup log. CloudWatch Pastikan CloudFormation template AWS mengacu pada bucket S3 yang berisi file yang Anda unggah sebelumnya. **Note** Anda juga dapat membuat tumpukan dengan menggunakan CloudFormation konsol AWS. Untuk informasi selengkapnya tentang ini, lihat Membuat tumpukan di CloudFormation konsol AWS dalam	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CloudFormation dokumentasi AWS.	
Mulai eksekusi baru di Step Functions.	Buka konsol Step Functions dan mulai eksekusi baru. Step Functions memanggil fungsi Lambda dan membuat koneksi peering untuk gateway transit. Anda tidak memerlukan file JSON masukan. Verifikasi bahwa lampiran tersedia dan jenis koneksinya adalah Peering. Untuk informasi selengkapnya tentang ini, lihat Memulai eksekusi baru dari Memulai AWS Step Functions dalam dokumentasi AWS Steps Functions.	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi rute dalam tabel rute.	Pengintipan Antar Wilayah didirikan di antara gerbang transit. Tabel rute diperbaru i dengan rentang blok IPv4 CIDR Region VPC peer. Buka konsol VPC Amazon dan pilih tab Asosiasi di tabel rute yang sesuai dengan lampiran gateway transit. Verifikasi rentang blok VPC CIDR dari Wilayah yang diintip. Untuk langkah dan petunjuk mendetail, lihat Mengaitkan tabel rute gateway transit di dokumentasi Amazon VPC.	Administrator jaringan

Sumber daya terkait

- Eksekusi dalam Step Functions
- Lampiran transit gateway peering
- Peer VPCs di Wilayah AWS yang berbeda dengan menggunakan AWS Transit Gateway
- Interkoneksi VPCs di seluruh Wilayah AWS menggunakan AWS Transit Gateway - Demo (video)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Memusatkan konektivitas jaringan menggunakan AWS Transit Gateway

Mydhili Palagummi dan Nikhil Marrapu, Amazon Web Services

Ringkasan

Pola ini menjelaskan konfigurasi paling sederhana di mana AWS Transit Gateway dapat digunakan untuk menghubungkan jaringan lokal ke cloud pribadi virtual (VPCs) di beberapa akun AWS dalam Wilayah AWS. Dengan pengaturan ini, Anda dapat membuat jaringan hybrid yang menghubungkan beberapa jaringan VPC di Wilayah dan jaringan lokal. Ini dilakukan dengan menggunakan gateway transit dan koneksi jaringan pribadi virtual (VPN) ke jaringan lokal.

Prasyarat dan batasan

Prasyarat

- Akun untuk layanan jaringan hosting, dikelola sebagai akun anggota organisasi di AWS Organizations
- VPCs di beberapa akun AWS, tanpa tumpang tindih blok Classless Inter-Domain Routing (CIDR)

Batasan

Pola ini tidak mendukung isolasi lalu lintas antara jaringan tertentu VPCs atau jaringan lokal. Semua jaringan yang terpasang pada gateway transit akan dapat menjangkau satu sama lain. Untuk mengisolasi lalu lintas, Anda perlu menggunakan tabel rute khusus pada gateway transit. Pola ini hanya menghubungkan jaringan VPCs dan lokal dengan menggunakan tabel rute gateway transit default tunggal, yang merupakan konfigurasi paling sederhana.

Arsitektur

Tumpukan teknologi target

- AWS Transit Gateway
- AWS Site-to-Site VPN
- VPC
- Manajer Akses Sumber Daya AWS (AWS RAM)

Arsitektur target

Alat

Layanan AWS

- AWS Resource Access Manager (AWS RAM) membantu Anda berbagi sumber daya dengan aman di seluruh akun AWS, unit organisasi, atau seluruh organisasi Anda dari AWS Organizations.
- AWS Transit Gateway adalah hub pusat yang menghubungkan cloud pribadi virtual (VPCs) dan jaringan lokal.

Epik

Buat gateway transit di akun layanan jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gateway transit.	Di akun AWS tempat Anda ingin meng-host layanan jaringan, buat gateway transit di Wilayah AWS target. Untuk petunjuknya, lihat Membuat gateway transit. Perhatikan hal-hal berikut: • Pilih Asosiasi tabel rute default. • Pilih Perbanyakan tabel rute default.	Administrator jaringan

Connect gateway transit ke jaringan lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan perangkat gateway pelanggan untuk koneksi VPN.	Perangkat gateway pelanggan terpasang di sisi lokal koneksi Site-to-Site VPN	Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	antara gateway transit dan jaringan lokal Anda. Untuk informasi selengkapnya, lihat perangkat gateway pelanggan Anda di dokumentasi AWS Site-to-Site VPN. Identifikasi atau luncurkan perangkat pelanggan lokal yang didukung dan catat alamat IP publiknya. Konfigurasi VPN selesai nanti dalam epik ini.	
Di akun layanan jaringan, buat lampiran VPN ke gateway transit.	Untuk mengatur koneksi, buat lampiran VPN untuk gateway transit. Untuk petunjuk, lihat Lampiran VPN gateway Transit.	Administrator jaringan
Konfigurasikan VPN pada perangkat gateway pelanggan di jaringan lokal Anda.	Unduh file konfigurasi untuk koneksi Site-to-Site VPN yang terkait dengan gateway transit dan konfigurasikan pengaturan VPN pada perangkat gateway pelanggan. Untuk petunjuk, lihat Mengunduh file konfigurasi.	Administrator jaringan

Bagikan gateway transit di akun layanan jaringan ke akun AWS lain atau organisasi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Di akun manajemen AWS Organizations, aktifkan berbagi.	Untuk berbagi gateway transit dengan organisasi Anda atau dengan unit organisas	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	i tertentu, aktifkan berbagi di AWS Organizations. Jika tidak, Anda perlu membagikan gateway transit untuk setiap akun satu per satu. Untuk petunjuknya, lihat Mengaktifkan berbagi sumber daya dalam AWS Organizations.	
Buat berbagi sumber daya gateway transit di akun layanan jaringan.	Untuk mengizinkan akun AWS lain VPCs dalam organisasi Anda terhubung ke gateway transit, di akun layanan jaringan, gunakan konsol AWS RAM untuk berbagi sumber daya gateway transit. Untuk petunjuknya, lihat Membuat berbagi sumber daya.	Administrator sistem AWS

Connect VPCs ke gateway transit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat lampiran VPC di akun individual.	Di akun tempat gateway transit telah dibagikan, buat lampiran VPC gateway transit. Untuk petunjuknya, lihat Membuat lampiran gateway transit ke VPC.	Administrator jaringan
Terima permintaan lampiran VPC.	Di akun layanan jaringan, terima permintaan lampiran VPC gateway transit. Untuk petunjuk, lihat Menerima lampiran bersama.	Administrator jaringan

Konfigurasikan perutean

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan rute di akun individual VPCs.	Di setiap VPC akun individual, tambahkan rute ke jaringan lokal dan ke jaringan VPC lainnya, menggunakan gateway transit sebagai target. Untuk petunjuk, lihat Menambahkan dan menghapus rute dari tabel rute.	Administrator jaringan
Konfigurasikan rute di tabel rute gateway transit.	Rute dari VPCs dan koneksi VPN harus disebarkan dan akan muncul di tabel rute default gateway transit. Jika diperlukan, buat rute statis apa pun (salah satu contohnya adalah rute statis untuk koneksi VPN statis) di tabel rute default gateway transit. Untuk petunjuk, lihat Membuat rute statis.	Administrator jaringan
Tambahkan aturan grup keamanan dan daftar kontrol akses jaringan (ACL).	Untuk EC2 instans dan sumber daya lainnya di VPC, pastikan bahwa aturan grup keamanan dan aturan ACL jaringan mengizinkan lalu lintas VPCs antara serta jaringan lokal. Untuk petunjuknya, lihat Mengontrol lalu lintas ke sumber daya menggunakan grup keamanan	Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan Menambahkan dan menghapus aturan dari ACL.	

Uji konektivitas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji konektivitas antara VPCs.	Pastikan bahwa ACL jaringan dan grup keamanan memungkinkan lalu lintas Internet Control Message Protocol (ICMP), dan kemudian ping dari instance dalam VPC ke VPC lain yang juga terhubung ke gateway transit.	Administrator jaringan
Uji konektivitas antara VPCs dan jaringan lokal.	Pastikan aturan ACL jaringan, aturan grup keamanan, dan firewall apa pun mengizinkan lalu lintas ICMP, lalu ping antara jaringan lokal dan instance di EC2 . VPCs Komunikasi jaringan harus dimulai dari jaringan lokal terlebih dahulu untuk membawa koneksi VPN ke UP status.	Administrator jaringan

Sumber daya terkait

- Membangun Infrastruktur Jaringan AWS multi VPC AWS yang dapat diskalakan dan aman (whitepaper AWS)
- Bekerja dengan sumber daya bersama (dokumentasi AWS RAM)

- Bekerja dengan gateway transit (dokumentasi AWS Transit Gateway)

Konfigurasikan enkripsi HTTPS untuk Oracle JD Edwards EnterpriseOne di Oracle WebLogic dengan menggunakan Application Load Balancer

Thanigaivel Thirumalai, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara mengkonfigurasi enkripsi HTTPS untuk pembongkaran SSL di Oracle JD EnterpriseOne Edwards pada beban kerja Oracle. WebLogic Pendekatan ini mengenkripsi lalu lintas antara browser pengguna dan penyeimbang beban untuk menghilangkan beban enkripsi dari server. EnterpriseOne

Banyak pengguna menskalakan tingkat mesin virtual EnterpriseOne JAVA (JVM) secara horizontal dengan menggunakan AWS Application Load Balancer.Load balancer berfungsi sebagai titik kontak tunggal untuk klien, dan mendistribusikan lalu lintas masuk ke beberapa. JVMs Secara opsional, penyeimbang beban dapat mendistribusikan lalu lintas di beberapa Availability Zone dan meningkatkan ketersediaan. EnterpriseOne

Proses yang dijelaskan dalam pola ini mengonfigurasi enkripsi antara browser dan penyeimbang beban alih-alih mengenkripsi lalu lintas antara penyeimbang beban dan. EnterpriseOne JVMs Pendekatan ini disebut sebagai pembongkaran SSL. Membongkar proses dekripsi SSL dari EnterpriseOne web atau server aplikasi ke Application Load Balancer mengurangi beban di sisi aplikasi. Setelah penghentian SSL di penyeimbang beban, lalu lintas yang tidak terenkripsi dialihkan ke aplikasi di AWS.

Oracle JD Edwards EnterpriseOne adalah solusi perencanaan sumber daya perusahaan (ERP) untuk organisasi yang memproduksi, membangun, mendistribusikan, melayani, atau mengelola produk atau aset fisik. JD Edwards EnterpriseOne mendukung berbagai perangkat keras, sistem operasi, dan platform basis data.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Peran AWS Identity and Access Management (IAM) yang memiliki izin untuk melakukan panggilan layanan AWS dan mengelola sumber daya AWS
- Sertifikat SSL

Versi produk

- Pola ini diuji dengan Oracle WebLogic 12c, tetapi Anda juga dapat menggunakan versi lain.

Arsitektur

Ada beberapa pendekatan untuk melakukan pembongkaran SSL. Pola ini menggunakan Application Load Balancer dan Oracle HTTP Server (OHS), seperti yang diilustrasikan dalam diagram berikut.

Diagram berikut menunjukkan JD Edwards EnterpriseOne, Application Load Balancer, dan Java Application Server (JAS) JVM layout.

Alat

Layanan AWS

- Application Load Balancer mendistribusikan lalu lintas aplikasi yang masuk ke beberapa target, seperti Amazon Elastic Compute Cloud (EC2 instans Amazon), di beberapa Availability Zone.
- AWS Certificate Manager (ACM) membantu Anda membuat, menyimpan, dan memperbarui sertifikat dan kunci SSL/TLS X.509 publik dan pribadi yang melindungi situs web dan aplikasi AWS Anda.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan terukur.

Praktik terbaik

- Untuk praktik terbaik ACM, lihat dokumentasi ACM.

Epik

Mengatur WebLogic dan OHS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan konfigurasikan komponen Oracle.	1. Instal Fusion Middlewar e Infrastructure dengan mengikuti proses instalasi	JDE CNC, administrator WebLogic

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	standar. Program ini membantu Anda menginstal dan mengkonfigurasi WebLogic domain. Untuk instruksi, lihat dokumentasi Oracle. 2. Instal OHS dengan mengikuti proses instalasi standar. Untuk instruksi, lihat dokumentasi Oracle. 3. Ketika instalasi selesai, mulai wizard konfigurasi (`config.sh` file) untuk mengkonfigurasi OHS. • Anda dapat memperbarui domain yang ada atau membuat domain baru. Pola ini mengasumsikan bahwa Anda memperbarui domain yang ada. • Untuk Template yang Tersedia, pilih Oracle Enterprise Manager-Restricted JRF dan Oracle HTTP Server (JRF Terbatas). Memilih opsi Java Required Files (JRF) ini menghilangkan koneksi ke database eksternal. • Untuk Server Terkelola, Cluster, Template Server, Kluster Koherensi, Mesin, Tetapkan Server ke	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Mesin, Target virtual, dan Partisi, terima nilai konfigurasi default dan pilih Berikutnya untuk pindah ke kategori berikutnya. • Lengkapi detail konfigurasi (misalnya, host dan port administrator, alamat dan port dengarkan , nama server) untuk instance OHS (misalnya ,`ohs1`).	
Aktifkan WebLogic plugin di tingkat domain.	WebLogic Plugin diperlukan untuk load balancing. Untuk mengaktifkan plugin: 1. Masuk ke konsol WebLogic administrasi dengan menggunakan tautan: `http://<WeblogicServer>:<Adminport>/console` 2. Pilih Lock & Edit, lalu pilih Configuration, Web Applications. 3. Pilih WebLogic Plugin Diaktifkan (kotak centang atau opsi dropdown). 4. Pilih Simpan dan Aktifkan Perubahan.	JDE CNC, administrator WebLogic

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Edit file konfigurasi.	`mod_wl_ohs.conf` File mengkonfigurasi permintaan proxy dari OHS ke. WebLogic
	1. Edit file ini. Itu terletak di:
	`$ORACLE_HOME/user_projects/domains/`
	Contoh:
	`/home/oracle/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain/config/fmwconfig/components/OHS/instances/ohs1`
	2. Tambahkan nilai WebLogic host (`WebLogicHost`) dan port (`WebLogicPort`) (Pola ini mengasumsikan localhost dan port 8000.)
	3. Tambahkan `WLProxySSL` dan `WLProxySSLPassThrough` nilai sebagai berikut:
	`<VirtualHost *:8000> <Location /jde> WLSRequest On SetHandler weblogic-handler WebLogicHost localhost WebLogicPort 8000`
		JDE CNC, administrator WebLogic

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>WLProxySSL On
	WLProxySSLPassThrough
	 On
	</Location>
	</VirtualHost></pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai K3 dengan menggunakan Enterprise Manager.	1. Masuk ke Enterprise Manager Fusion Middleware dengan menggunakan tautan: `http://<WeblogicServer>:<Adminport>/em/` 2. Di Navigasi Target, di bawah Server HTTP, pilih instance OHS (misalnya ,`ohs1`). 3. Pilih Shut Down dan Start Up untuk memulai ulang instans K3. 4. Ketika pengaturan OHS selesai, Anda dapat terhubung ke klien EnterpriseOne HTML dengan menggunakan nama host server HTTP Anda dengan port 8000 alih-alih nama host EnterpriseOne server. • Tautan lama: `http://<Webserver>:80/jde/owhtml` • Tautan baru: `http://<HTTP server or web server>:8000/jde/owhtml` Jika Anda menggunakan port selain port HTTP Oracle default, edit	JDE CNC, administrator WebLogic

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`httpd.conf` file untuk menambahkan listener untuk port tersebut di dua tempat:
	`#[Listen] OHS_LISTEN_PORT`
	`Listen 8000`
	dan:
	`#`
	`ServerName <Weblogic Server1>:8000`	

Konfigurasikan Application Load Balancer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kelompok sasaran.	1. Buat grup target untuk port server HTTP 8000. 2. Daftarkan target di bawah kelompok sasaran dengan port yang sama. 3. Periksa status target untuk memastikan bahwa mereka sehat. 4. Konfigurasikan pengaturan pemeriksaan kesehatan seperlunya.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk petunjuk terperinci, lihat dokumentasi Elastic Load Balancing.	
Siapkan penyeimbang beban.	1. Buat Application Load Balancer dengan atribut default dan virtual private cloud (VPC) yang diperlukan, grup keamanan, dan subnet. Untuk petunjuk, lihat dokumentasi Elastic Load Balancing. 2. Tambahkan entri listener untuk HTTPS 443 dan teruskan ke grup target yang Anda buat pada langkah sebelumnya. (Untuk petunjuk, lihat dokumentasi Elastic Load Balancing.) Pendengar HTTPS memerlukan sertifikat SSL. Anda dapat memilih sertifikat dari ACM atau mengunggahnya. 3. Untuk kedua pendengar, aktifkan kekakuan dengan mengikuti instruksi dalam dokumentasi Elastic Load Balancing.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan catatan Route 53 (DNS).	(Opsional) Anda dapat menambahkan data DNS Amazon Route 53 untuk subdomain. Catatan ini akan mengarah ke Application Load Balancer Anda. Untuk petunjuk, lihat dokumentasi Route 53.	Administrator AWS

Pemecahan masalah

Isu	Solusi
Server HTTP tidak muncul.	Jika Server HTTP tidak muncul di daftar Target Navigasi di konsol Enterprise Manager, ikuti langkah-langkah berikut: 1. Di bawah WebLogic Domain, Administrasi, pilih Instans K3. 2. Pilih Buat untuk membuat instance K3 baru. 3. Berikan nama instance, lalu pilih OK untuk membuat instance. Ketika instance telah dibuat dan perubahan telah diaktifkan, Anda akan dapat melihat server HTTP di panel Target Navigation.

Sumber daya terkait

Dokumentasi AWS

- Penyeimbang Beban Aplikasi
- Bekerja dengan zona yang dihosting publik

- Bekerja dengan zona yang dihosting pribadi

Dokumentasi Oracle:

- Ikhtisar Plug-In Proxy WebLogic Server Oracle
- Instalasi WebLogic Server menggunakan Infrastructure Installer
- Menginstal dan Mengkonfigurasi Oracle HTTP Server

Connect ke data Application Migration Service dan kontrol pesawat melalui jaringan pribadi

Dipin Jain dan Mike Kuznetsov, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana Anda dapat terhubung ke bidang AWS Application Migration Service data dan bidang kontrol pada jaringan pribadi yang aman dengan menggunakan titik akhir VPC antarmuka.

Layanan Migrasi Aplikasi adalah solusi lift-and-shift (rehost) yang sangat otomatis yang menyederhanakan, mempercepat, dan mengurangi biaya migrasi aplikasi. AWS Ini memungkinkan perusahaan untuk meng-host kembali sejumlah besar server fisik, virtual, atau cloud tanpa masalah kompatibilitas, gangguan kinerja, atau jendela cutover yang panjang. Layanan Migrasi Aplikasi tersedia dari Konsol Manajemen AWS. Ini memungkinkan integrasi tanpa batas dengan yang lain Layanan AWS, seperti AWS CloudTrail, Amazon CloudWatch, dan AWS Identity and Access Management (IAM).

Anda dapat terhubung dari pusat data sumber ke bidang data—yaitu, ke subnet yang berfungsi sebagai area pementasan untuk replikasi data di VPC tujuan—melalui koneksi pribadi dengan menggunakan layanan Site-to-Site VPN , AWS Direct Connect atau mengintip VPC di Layanan Migrasi Aplikasi. Anda juga dapat menggunakan titik akhir VPC antarmuka yang didukung oleh AWS PrivateLink untuk terhubung ke bidang kontrol Layanan Migrasi Aplikasi melalui jaringan pribadi.

Prasyarat dan batasan

Prasyarat

- Subnet area pementasan — Sebelum Anda menyiapkan Layanan Migrasi Aplikasi, buat subnet yang akan digunakan sebagai area pementasan untuk data yang direplikasi dari server sumber Anda ke AWS (yaitu, bidang data). Anda harus menentukan subnet ini di template Pengaturan Replikasi saat pertama kali mengakses konsol Layanan Migrasi Aplikasi. Anda dapat mengganti subnet ini untuk server sumber tertentu di template Pengaturan Replikasi. Meskipun Anda dapat menggunakan subnet yang ada di Anda Akun AWS, kami sarankan Anda membuat subnet khusus baru untuk tujuan ini.
- Persyaratan jaringan — Server replikasi yang diluncurkan oleh Layanan Migrasi Aplikasi di subnet area pementasan Anda harus dapat mengirim data ke titik akhir API Layanan Migrasi Aplikasi

dihttps://mgn.<region>.amazonaws.com/, di mana <region> kode untuk Wilayah AWS Anda replikasi (misalnya,). https://mgn.us-east-1.amazonaws.com Layanan Amazon Simple Storage Service (Amazon S3) diperlukan untuk mengunduh URLs perangkat lunak Layanan Migrasi Aplikasi.

- Installer Agen AWS Replikasi harus memiliki akses ke URL bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) yang Anda gunakan dengan Layanan Wilayah AWS Migrasi Aplikasi.
- Subnet area pementasan harus memiliki akses ke Amazon S3.
- Server sumber tempat Agen AWS Replikasi diinstal harus dapat mengirim data ke server replikasi di subnet area pementasan dan ke titik akhir API Layanan Migrasi Aplikasi di. https://mgn.<region>.amazonaws.com/

Tabel berikut mencantumkan port yang diperlukan.

Sumber	Destinasi	Port	Untuk informasi selengkapnya, silakan lihat
Pusat data sumber	Layanan Amazon S3 URLs	443 (TCP)	Komunikasi melalui port TCP 443
Pusat data sumber	Wilayah AWS-alama t konsol khusus untuk Layanan Migrasi Aplikasi	443 (TCP)	Komunikasi antara server sumber dan Layanan Migrasi Aplikasi melalui port TCP 443
Pusat data sumber	Subnet area pementasan	1500 (TCP)	Komunikasi antara server sumber dan subnet area pementasan melalui port TCP 1500
Subnet area pementasan	Wilayah AWS-alama t konsol khusus untuk Layanan Migrasi Aplikasi	443 (TCP)	Komunikasi antara subnet area pementasan dan Layanan Migrasi

			Aplikasi melalui port TCP 443
Subnet area pementasan	Layanan Amazon S3 URLs	443 (TCP)	Komunikasi melalui port TCP 443
Subnet area pementasan	Titik akhir Amazon Elastic Compute Cloud (Amazon EC2) dari subnet Wilayah AWS	443 (TCP)	Komunikasi melalui port TCP 443

Keterbatasan

Layanan Migrasi Aplikasi saat ini tidak tersedia di semua Wilayah AWS dan sistem operasi.

- Didukung Wilayah AWS
- Sistem operasi yang didukung

Arsitektur

Diagram berikut menggambarkan arsitektur jaringan untuk migrasi tipikal. Untuk informasi selengkapnya tentang arsitektur ini, lihat dokumentasi Layanan Migrasi Aplikasi dan arsitektur layanan Layanan Migrasi Aplikasi dan video arsitektur jaringan.

Tampilan rinci berikut menunjukkan konfigurasi titik akhir VPC antarmuka di area pementasan VPC untuk menghubungkan Amazon S3 dan Layanan Migrasi Aplikasi.

Alat

- AWS Application Migration Servicemenyederhanakan, mempercepat, dan mengurangi biaya rehosting aplikasi. AWS
- Endpoint VPC antarmuka memungkinkan Anda untuk terhubung ke layanan yang didukung oleh AWS PrivateLink tanpa memerlukan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk

berkomunikasi dengan sumber daya di layanan. Lalu lintas antara VPC Anda dan layanan lainnya tidak meninggalkan jaringan Amazon.

Epik

Buat titik akhir untuk Layanan Migrasi Aplikasi, Amazon EC2, dan Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan titik akhir antarmuka untuk Layanan Migrasi Aplikasi.	Pusat data sumber dan area pementasan VPC terhubung secara pribadi ke bidang kontrol Layanan Migrasi Aplikasi melalui titik akhir antarmuka yang Anda buat di area pementasan target VPC. Untuk membuat titik akhir: 1. Buka konsol Amazon Virtual Private Cloud (Amazon VPC). 2. Di panel navigasi, pilih Endpoints, Create Endpoint. 3. Untuk Kategori layanan, pilih Layanan AWS. 4. Untuk Nama Layanan, masukkan `com.amazonaws.<region>.mgn` . Untuk Jenis, pilih Antarmuka. 5. Untuk VPC, pilih area pementasan target VPC untuk membuat titik akhir. 6. Untuk Subnet, pilih subnet untuk membuat antarmuka jaringan endpoint.	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Untuk mengaktifkan DNS pribadi untuk titik akhir antarmuka, di bagian Pengaturan tambahan, pilih Aktifkan Nama DNS. 8. Pilih grup keamanan yang memungkinkan masuknya dari subnet VPC area pementasan melalui TCP 443. 9. Pilih Buat titik akhir. Untuk informasi selengkap nya, lihat Mengakses titik akhir VPC antarmuka Layanan AWS menggunak an antarmuka di dokumentasi VPC Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan titik akhir antarmuka untuk Amazon EC2.	Area pementasan VPC terhubung secara pribadi ke EC2 Amazon API melalui titik akhir antarmuka yang Anda buat di area pementasan target VPC. Untuk membuat titik akhir, ikuti instruksi yang diberikan di cerita sebelumnya. • Untuk nama layanan, masukkan `com.amazonaws.<region>.ec2` . Untuk Jenis, pilih Antarmuka . • Grup keamanan harus mengizinkan lalu lintas HTTPS masuk dari subnet VPC area pementasan melalui port 443. • Di bagian Pengaturan tambahan, pilih Aktifkan nama DNS.	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan titik akhir antarmuka untuk Amazon S3.	Pusat data sumber dan area pementasan VPC terhubung secara pribadi ke Amazon S3 API melalui titik akhir antarmuka yang Anda buat di area pementasan target VPC. Untuk membuat titik akhir, ikuti instruksi yang diberikan di cerita pertama. • Untuk Nama Layanan, masukkan `com.amazonaws.<region>.s3` . Untuk Jenis, pilih Antarmuka . • Grup keamanan VPC harus mengizinkan lalu lintas HTTPS masuk dari subnet VPC area pementasan melalui port 443. • Di bagian Pengaturan tambahan, hapus Aktifkan nama DNS. Titik akhir antarmuka Amazon S3 tidak mendukung nama DNS pribadi. **Note** Anda menggunakan titik akhir antarmuka karena koneksi titik akhir gateway tidak dapat diperpanjang	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dari VPC. (Untuk detailnya, lihat AWS PrivateLink dokumentasi.)	
Konfigurasikan titik akhir Amazon S3 Gateway.	Selama fase konfigurasi, server replikasi harus terhubung ke bucket S3 untuk mengunduh pembaruan perangkat lunak Server AWS Replikasi. Namun, titik akhir antarmuka Amazon S3 tidak mendukung nama DNS pribadi, dan tidak ada cara untuk menyediakan nama DNS titik akhir Amazon S3 ke server replikasi. Untuk mengurangi masalah ini, Anda membuat titik akhir gateway Amazon S3 di VPC yang menjadi milik subnet area pementasan, dan memperbarui tabel rute subnet pementasan dengan rute yang relevan. Untuk informasi selengkapnya, lihat Membuat titik akhir gateway dalam AWS PrivateLink dokumentasi.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan DNS lokal untuk menyelesaikan nama DNS pribadi untuk titik akhir.	Titik akhir antarmuka untuk Layanan Migrasi Aplikasi dan Amazon EC2 memiliki nama DNS pribadi yang dapat diselesaikan di VPC. Namun, Anda juga perlu mengonfigurasi server lokal untuk menyelesaikan nama DNS pribadi untuk titik akhir antarmuka ini. Ada beberapa cara untuk mengkonfigurasi server ini. Dalam pola ini, kami menguji fungsionalitas ini dengan meneruskan kueri DNS lokal ke titik akhir Amazon Route 53 Resolver masuk di area pementasan VPC. Untuk informasi selengkapnya, lihat Menyelesaikan kueri DNS antara VPCs dan jaringan Anda di dokumentasi Route 53.	Insinyur migrasi

Connect ke bidang kontrol Layanan Migrasi Aplikasi melalui tautan pribadi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Agen AWS Replikasi dengan menggunakan AWS PrivateLink.	1. Unduh Agen AWS Replikasi ke bucket S3 pribadi di Wilayah tujuan. 2. Masuk ke server sumber untuk dimigrasi. Installer	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Agen AWS Replikasi memerlukan akses jaringan ke Layanan Migrasi Aplikasi dan titik akhir Amazon S3. Karena jaringan lokal Anda tidak terbuka untuk Layanan Migrasi Aplikasi dan titik akhir publik Amazon S3, Anda harus menginstal Agen dengan bantuan titik akhir antarmuka yang Anda buat di langkah sebelumnya dengan menggunakan. AWS PrivateLink Berikut adalah contoh untuk Linux: 1. Unduh Agen dengan menggunakan perintah: `wget -O ./aws-replication-installer-init.py \ https://aws-application-migration-service-<aws_region>.bucket.<s3-endpoint-DNS-name>/latest/linux/aws-replication-installer-init.py` Misalnya, jika nama DNS dari titik akhir antarmuka	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon S3 `vpce-009c8b07adb052a11-qgf8q50y.s3.us-west-1.vpce.amazonaws.com` adalah dan Wilayah AWS sedang, Anda akan menggunakan perintah: `us-west-1` `wget -O ./aws-replication-installer-init.py \ https://aws-application-migration-service-us-west-1.bucket.vpce-009c8b07adb052a11-qgf8q50y.s3.us-west-1.vpce.amazonaws.com/latest/linux/aws-replication-installer-init.py` **Note** `bucket`adalah kata kunci statis yang harus Anda tambahkan sebelum nama DNS titik akhir antarmuka Amazon S3. Untuk informasi lebih lanjut, lihat Dokumentasi Amazon S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Instal Agen:
	• Jika Anda memilih Aktifkan nama DNS saat membuat titik akhir antarmuka untuk Layanan Migrasi Aplikasi, jalankan perintah:
	`sudo python3 aws-replication-installer-init.py \ --region <aws_region> \ --aws-access-key-id <access-key> \ --aws-secret-access-key <secret-key> \ --no-prompt \ --s3-endpoint <s3-endpoint-DNS-name>`
	• Jika Anda tidak memilih Aktifkan nama DNS saat membuat titik akhir antarmuka untuk Layanan Migrasi Aplikasi, jalankan perintah:
	`sudo python3 aws-replication-installer-init.py \ --region <aws_region> \ --aws-access-key-id <access-key> \`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--aws-secret-access-key <secret-key> \
	--no-prompt \
	--s3-endpoint <s3-endpoint-DNS-name> \
	--endpoint <mgn-endpoint-DNS-name></pre>
	Untuk informasi selengkapnya, lihat petunjuk penginstalan Agen AWS Replikasi di dokumentasi Layanan Migrasi Aplikasi.
	
	Setelah Anda membuat koneksi dengan Layanan Migrasi Aplikasi dan menginstal Agen AWS Replikasi, ikuti petunjuk dalam dokumentasi Layanan Migrasi Aplikasi untuk memigrasikan server sumber Anda ke VPC dan subnet target Anda.	

Sumber daya terkait

Dokumentasi Layanan Migrasi Aplikasi

- Konsep
- Alur kerja migrasi
- Panduan memulai cepat
- FAQ

- Pemecahan Masalah

Sumber daya tambahan

- Menghosting ulang aplikasi Anda dalam arsitektur multi-akun AWS dengan menggunakan titik akhir antarmuka VPC (panduan panduan preskriptif)AWS
- AWS Application Migration Service — Pengantar Teknis (Panduan AWS Pelatihan dan Sertifikasi)
- AWS Application Migration Service arsitektur dan arsitektur jaringan (video)

Informasi tambahan

Pemecahan masalah AWS Instalasi Agen Replikasi di server Linux

Jika Anda mendapatkan kesalahan gcc di server Amazon Linux, konfigurasikan repositori paket dan gunakan perintah berikut:

```
## sudo yum groupinstall "Development Tools"
```

Buat objek Infoblox menggunakan sumber daya CloudFormation khusus AWS dan Amazon SNS

Tim Sutton, Amazon Web Services

Ringkasan

Pemberitahuan: AWS Cloud9 tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS Cloud9 dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

Infoblox Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), dan manajemen alamat IP (Infoblox DDI) memungkinkan Anda untuk memusatkan dan mengontrol lingkungan hybrid yang kompleks secara efisien. Dengan Infoblox DDI, Anda dapat menemukan dan merekam semua aset jaringan dalam satu database manajemen alamat IP otoritatif (IPAM), selain mengelola DNS di tempat dan di Amazon Web Services (AWS) Cloud dengan menggunakan peralatan yang sama.

Pola ini menjelaskan cara menggunakan sumber daya CloudFormation khusus AWS untuk membuat objek Infoblox (misalnya, catatan DNS atau objek IPAM) dengan memanggil Infoblox WAPI API. Untuk informasi selengkapnya tentang WAPI Infoblox, lihat dokumentasi WAPI di dokumentasi Infoblox.

Dengan menggunakan pendekatan pola ini, Anda dapat memperoleh tampilan terpadu dari catatan DNS dan konfigurasi IPAM untuk AWS dan lingkungan lokal, selain menghapus proses manual yang membuat catatan dan menyediakan jaringan Anda. Anda dapat menggunakan pendekatan pola ini untuk kasus penggunaan berikut:

- Menambahkan catatan A setelah membuat instance Amazon Elastic Compute Cloud (Amazon EC2)
- Menambahkan catatan CNAME setelah membuat Application Load Balancer
- Menambahkan objek jaringan setelah membuat virtual private cloud (VPC)
- Menyediakan rentang jaringan berikutnya dan menggunakan rentang itu untuk membuat subnet

Anda juga dapat memperluas pola ini dan menggunakan fitur perangkat Infoblox lainnya seperti menambahkan jenis catatan DNS yang berbeda atau mengonfigurasi Infoblox vDiscovery.

Pola ini menggunakan hub-and-spoke desain di mana hub memerlukan konektivitas ke alat Infoblox di AWS Cloud atau di tempat dan menggunakan AWS Lambda untuk memanggil Infoblox API.

Spoke berada di akun yang sama atau berbeda di organisasi yang sama di AWS Organizations, dan memanggil fungsi Lambda dengan menggunakan sumber daya CloudFormation khusus AWS.

Prasyarat dan batasan

Prasyarat

- Alat atau kisi Infoblox yang ada, diinstal di AWS Cloud, di tempat, atau keduanya, dan dikonfigurasi dengan pengguna admin yang dapat mengelola tindakan IPAM dan DNS. Untuk informasi selengkapnya tentang ini, lihat Tentang akun admin di dokumentasi Infoblox.
- Zona otoritatif DNS yang ada yang ingin Anda tambahkan catatan pada alat Infoblox. Untuk informasi selengkapnya tentang ini, lihat Mengonfigurasi zona otoritatif dalam dokumentasi Infoblox.
- Dua akun AWS aktif di AWS Organizations. Satu akun adalah akun hub dan akun lainnya adalah akun spoke.
- Akun hub dan spoke harus berada di Wilayah AWS yang sama.
- VPC akun hub harus terhubung ke alat Infoblox; misalnya, dengan menggunakan AWS Transit Gateway atau peering VPC.
- AWS Serverless Application Model (AWS SAM), diinstal dan dikonfigurasi secara lokal dengan AWS Cloud9 atau AWS. CloudShell
- `ClientTest.yaml`File `Infoblox-Hub.zip` dan (terlampir), diunduh ke lingkungan lokal yang berisi AWS SAM.

Batasan

- Token layanan sumber daya CloudFormation khusus AWS harus berasal dari Wilayah yang sama tempat tumpukan dibuat. Sebaiknya gunakan akun hub di setiap Wilayah, alih-alih membuat topik Amazon Simple Notification Service (Amazon SNS) di satu Wilayah dan memanggil fungsi Lambda di Wilayah lain.

Versi produk

- Infoblox WAPI versi 2.7

Arsitektur

Diagram berikut menunjukkan alur kerja pola ini.

Diagram menunjukkan komponen-komponen berikut untuk solusi pola ini:

1. Sumber daya CloudFormation khusus AWS memungkinkan Anda menulis logika penyediaan khusus dalam templat yang CloudFormation dijalankan AWS saat Anda membuat, memperbarui, atau menghapus tumpukan. Saat Anda membuat tumpukan, AWS CloudFormation mengirimkan `create` permintaan ke topik SNS yang dipantau oleh aplikasi yang berjalan pada instance EC2 .
2. Pemberitahuan Amazon SNS dari sumber daya CloudFormation khusus AWS dienkripsi melalui kunci AWS Key Management Service (AWS KMS) tertentu dan akses dibatasi ke akun di organisasi Anda di Organizations. Topik SNS memulai sumber daya Lambda yang memanggil API WAPI Infoblox.
3. Amazon SNS memanggil fungsi Lambda berikut yang menggunakan URL WAPI Infoblox, nama pengguna, dan kata sandi AWS Secrets Manager Amazon Resource Names () sebagai variabel lingkungan: ARNs
   - `dnsapi.lambda_handler`— Menerima`DNSName`,`DNSType`, dan `DNSValue` nilai dari sumber daya CloudFormation khusus AWS dan menggunakannya untuk membuat catatan DNS A dan CNAMES.
   - `ipaddr.lambda_handler`— Menerima `VPCCIDRType`,`SubnetPrefix`,, dan `Network Name` nilai dari sumber daya CloudFormation khusus AWS dan menggunakannya untuk menambahkan data jaringan ke dalam database IPAM Infoblox atau menyediakan sumber daya khusus dengan jaringan berikutnya yang tersedia yang dapat digunakan untuk membuat subnet baru.
   - `describeprefixes.lambda_handler`— Memanggil `describe_managed_prefix_lists` AWS API dengan menggunakan `"com.amazonaws."+Region+".s3"` filter untuk mengambil yang diperlukan`prefix ID`.

   > **Important**
   >
   > Fungsi Lambda ini ditulis dengan Python dan mirip satu sama lain tetapi panggilan berbeda. APIs
4. Anda dapat menerapkan grid Infoblox sebagai peralatan jaringan fisik, virtual, atau berbasis cloud. Ini dapat digunakan di tempat atau sebagai alat virtual menggunakan berbagai hypervisor,

termasuk, VMware ESXi Microsoft Hyper-V, Linux KVM, dan Xen. Anda juga dapat menerapkan kisi Infoblox di AWS Cloud dengan Amazon Machine Image (AMI).

5. Diagram menunjukkan solusi hybrid untuk grid Infoblox yang menyediakan DNS dan IPAM ke sumber daya di AWS Cloud dan di tempat.

Tumpukan teknologi

- AWS CloudFormation
- IAM
- AWS KMS
- AWS Lambda
- AWS SAM
- AWS Secrets Manager
- Amazon SNS
- Amazon VPC

Alat

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizations adalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.

- AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka yang membantu Anda membangun aplikasi tanpa server di AWS Cloud.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Kode

Anda dapat menggunakan `ClientTest.yaml` contoh CloudFormation template AWS (terlampir) untuk menguji hub Infoblox. Anda dapat menyesuaikan CloudFormation template AWS untuk menyertakan sumber daya khusus dari tabel berikut.

Buat catatan A menggunakan sumber daya kustom Infoblox spoke	Mengembalikan nilai:
	`infobloxref` — Referensi Infoblox
	Contoh sumber daya:
	`ARECORDCustomResource:`
	`  Type: "Custom::InfobloxAPI"`
	`  Properties:`
	`  ServiceToken: !Sub arn:aws:sns:${AWS::Region}:${HubAccountID}:RunInfobloxDNSFunction`
	`  DNSName: 'arecordtest.company.com'`
	`  DNSType: 'ARecord'`
	`  DNSValue: '10.0.0.1'`

Buat catatan CNAME menggunakan sumber daya kustom Infoblox spoke

Mengembalikan nilai:

`infobloxref` — Referensi Infoblox

Contoh sumber daya:

```
CNAMECustomResource:

  Type: "Custom::InfobloxAPI"

  Properties:

    ServiceToken: !Sub arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfoblox

    DNSFunction

    DNSName: 'cnametest.company.com'

    DNSType: 'cname'

    DNSValue: 'aws.amazon.com'
```

Buat objek jaringan menggunakan sumber daya kustom Infoblox spoke	Mengembalikan nilai:
	`infobloxref` — Referensi Infoblox
	`network`— Jangkauan jaringan (sama sepertiVPCCIDR)
	Contoh sumber daya:
	`VPCCustomResource:`
	`  Type: 'Custom::InfobloxAPI'`
	`  Properties:`
	`    ServiceToken: !Sub  arn:aws:sns:${AWS::Region}:${HubAccountID}:RunInfobloxNextSubnetFunction`
	`    VPCCIDR: !Ref VpcCIDR`
	`    Type: VPC`
	`    NetworkName: My-VPC`

Ambil subnet berikutnya yang tersedia menggunakan sumber daya kustom Infoblox spoke	Mengembalikan nilai:
	`infobloxref` — Referensi Infoblox
	`network` — Jangkauan jaringan subnet
	Contoh sumber daya:
	`Subnet1CustomResource:`
	`  Type: 'Custom::InfobloxAPI'`
	`  DependsOn: VPCCustomResource`
	`  Properties:`
	`  ServiceToken: !Sub arn:aws:sns:${AWS::Region}:${HubAccountID}:RunInfobloxNextSubnetFunction`
	`  VPCCIDR: !Ref VpcCIDR`
	`  Type: Subnet`
	`  SubnetPrefix: !Ref SubnetPrefix`
	`NetworkName: My-Subnet`

Epik

Buat dan konfigurasikan VPC akun hub

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC dengan koneksi ke alat Infoblox.	Masuk ke AWS Management Console untuk akun hub Anda dan buat VPC dengan mengikuti langkah-langkah di Amazon VPC pada penerapan	Administrator jaringan, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	referensi AWS Cloud Quick Start dari AWS Quick Starts. **Important** VPC harus memiliki konektivitas HTTPS ke alat Infoblox dan kami menyarankan Anda menggunakan subnet pribadi untuk koneksi ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Buat titik akhir VPC untuk subnet pribadi.	Endpoint VPC menyediakan konektivitas ke layanan publik untuk subnet pribadi Anda. Titik akhir berikut diperlukan: • Titik akhir gateway untuk Amazon Simple Storage Service (Amazon S3) untuk memungkinkan Lambda berkomunikasi dengan AWS CloudFormation • Endpoint antarmuka untuk Secrets Manager untuk mengaktifkan konektivitas dengan Secrets Manager • Titik akhir antarmuka untuk AWS KMS untuk memungkinkan enkripsi topik SNS dan rahasia Secrets Manager Untuk informasi selengkapnya tentang membuat titik akhir untuk subnet pribadi, lihat titik akhir VPC VPC di dokumenta si Amazon VPC.	Administrator jaringan, Administrator sistem

Menyebarkan hub Infoblox

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bangun template AWS SAM.	1. Jalankan `unzip Infoblox-Hub.zip`	Pengembang, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	perintah di lingkungan yang berisi AWS SAM. 2. Jalankan `cd Hub/` perintah untuk mengubah direktori Anda ke `Hub` direktori. 3. Jalankan `sam build` perintah untuk memproses file template AWS SAM, kode aplikasi, dan file serta dependensi khusus bahasa apa pun. `sam build`Perintah tersebut juga menyalin artefak build dalam format dan lokasi yang diharapkan untuk cerita berikut.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan template AWS SAM.	`sam deploy`Perintah mengambil parameter yang diperlukan dan menyimpannya ke dalam `samconfig.toml` file, menyimpan CloudFormation template AWS dan fungsi Lambda dalam bucket S3, dan kemudian menerapkan template CloudFormation AWS ke akun hub Anda.
	
	Contoh kode berikut menunjukkan cara menerapkan template AWS SAM:
	
	`$ sam deploy --guided`
	
	`Configuring SAM deploy`
	`==========`
	`Looking for config file [samconfig.toml] : Found`
	`Reading default arguments : Success`
	`Setting default arguments for 'sam deploy'`
	`=========`
	`Stack Name [Infoblox-Hub]:`
	`AWS Region [eu-west-1]:`
	`Parameter InfobloxUsername:`
	`Parameter InfobloxPassword:`
		Pengembang, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Parameter InfobloxIPAddress [xxx.xxx.xx.xxx]: Parameter AWSOrganisationID [o-xxxxxxxxx]: Parameter VPCID [vpc-xxxxxxxxx]: Parameter VPCCIDR [xxx.xxx.xxx.xxx/16]: Parameter VPCSubnetID1 [subnet-xxx]: Parameter VPCSubnetID2 [subnet-xxx]: Parameter VPCSubnetID3 [subnet-xxx]: Parameter VPCSubnetID4 []: #Shows you resources changes to be deployed and require a 'Y' to initiate deploy Confirm changes before deploy [Y/n]: y #SAM needs permission to be able to create roles to connect to the resources in your template Allow SAM CLI IAM role creation [Y/n]: n Capabilities [['CAPABILITY_NAMED_IAM']]: Save arguments to configuration file [Y/n]: y	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`SAM configura tion file [samconfi g.toml]: SAM configura tion environment [default]:` **Important** Anda harus menggunakan `--guided` opsi ini setiap kali karena kredenal masuk Infoblox tidak disimpan dalam file. `samconfig.toml`	

Sumber daya terkait

- Memulai dengan WAPIs menggunakan Postman (Infoblox Blog)
- Penyediaan VNIO untuk AWS Menggunakan Model BYOL (dokumentasi Infoblox)
- quickstart-aws-vpc(GitHub repo)
- describe_managed_prefix_lists (AWS SDK untuk dokumentasi Python)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat arsitektur IPAM Multi-wilayah yang hierarkis dengan menggunakan Terraform AWS

Donny Schreiber, Amazon Web Services

Ringkasan

IP Address Management (IPAM) adalah komponen penting dari manajemen jaringan, dan menjadi semakin kompleks ketika organisasi menskalakan infrastruktur cloud mereka. Tanpa IPAM yang tepat, organisasi berisiko konflik alamat IP, ruang alamat yang terbuang, dan pemecahan masalah kompleks yang dapat menyebabkan pemadaman dan downtime aplikasi. Pola ini menunjukkan bagaimana menerapkan solusi IPAM komprehensif untuk lingkungan AWS perusahaan dengan menggunakan HashiCorp Terraform. Ini membantu organisasi untuk membuat arsitektur IPAM multi-wilayah hierarkis yang memfasilitasi manajemen alamat IP terpusat di semua Akun AWS dalam suatu organisasi.AWS

Pola ini membantu Anda menerapkan Manajer Alamat IP VPC Amazon dengan hierarki kumpulan empat tingkat yang canggih: kolam tingkat atas, kolam Regional, kumpulan unit bisnis, dan kumpulan khusus lingkungan. Struktur ini mendukung tata kelola alamat IP yang tepat sambil memungkinkan pendelegasian manajemen IP ke tim yang sesuai dalam organisasi. Solusinya menggunakan AWS Resource Access Manager (AWS RAM) untuk berbagi kumpulan IP Address Manager secara mulus di seluruh organisasi. AWS RAM memusatkan dan menstandarisasi spesifikasi IPAM, yang dapat dibangun oleh tim di semua akun yang dikelola.

Pola ini dapat membantu Anda mencapai hal berikut:

- Mengotomatiskan alokasi alamat IP di seluruh Wilayah AWS, unit bisnis, dan lingkungan.
- Menegakkan kebijakan jaringan organisasi melalui validasi terprogram.
- Skala infrastruktur jaringan secara efisien seiring dengan berkembangnya kebutuhan bisnis.
- Mengurangi overhead operasional melalui manajemen terpusat ruang alamat IP.
- Mempercepat penerapan beban kerja cloud-native dengan alokasi rentang CIDR swalayan.
- Mencegah konflik alamat melalui kontrol dan validasi berbasis kebijakan.

Prasyarat dan batasan

Prasyarat

- Satu atau lebih Akun AWS, dikelola sebagai organisasi di AWS Organizations.

- Hub jaringan atau akun manajemen jaringan yang akan berfungsi sebagai administrator yang didelegasikan oleh IP Address Manager.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi.
- Terraform versi 1.5.0 atau yang lebih baru, diinstal.
- AWS Penyedia untuk Terraform, dikonfigurasi.
- Izin untuk mengelola IP Address Manager, AWS RAM, dan virtual private cloud (VPCs), dikonfigurasi di AWS Identity and Access Management (IAM).

Batasan

- IP Address Manager tunduk pada kuota layanan. Kuota layanan default untuk pool adalah 50 per lingkup. Menjalankan penerapan ini untuk 6 Wilayah, 2 unit bisnis, dan 4 lingkungan akan membuat 67 kumpulan. Oleh karena itu, peningkatan kuota mungkin diperlukan.
- Memodifikasi atau menghapus kumpulan IP Address Manager setelah sumber daya dialokasikan dapat menyebabkan masalah ketergantungan. Anda harus melepaskan alokasi sebelum Anda dapat menghapus kumpulan.
- Di IP Address Manager, pemantauan sumber daya dapat mengalami sedikit keterlambatan dalam mencerminkan perubahan sumber daya. Penundaan ini bisa sekitar 20 menit.
- IP Address Manager tidak dapat secara otomatis menerapkan keunikan alamat IP di berbagai cakupan.
- Tag khusus harus mematuhi praktik terbaik AWS penandaan. Misalnya, setiap kunci harus unik dan tidak dapat dimulai dengan`aws:`.
- Ada pertimbangan dan batasan saat mengintegrasikan IP Address Manager dengan akun di luar organisasi Anda.

Arsitektur

Arsitektur target

Konfigurasi IP Address Manager dan hirarki pool

Diagram berikut menunjukkan konstruksi logis dari arsitektur target. Lingkup adalah wadah tingkat tertinggi di IP Address Manager. Setiap ruang lingkup mewakili ruang alamat IP untuk satu jaringan. Kumpulan adalah kumpulan rentang alamat IP yang berdekatan (atau rentang CIDR) dalam ruang lingkup. Pools membantu Anda mengatur alamat IP Anda sesuai dengan kebutuhan routing dan

keamanan Anda. Diagram ini menunjukkan empat tingkat hierarkis kolam: kolam tingkat atas, kolam Regional, kolam unit bisnis, dan kolam lingkungan.

Solusi ini menetapkan hierarki yang jelas dari kumpulan IP Address Manager:

1. Kumpulan tingkat atas mencakup seluruh ruang alamat IP organisasi, seperti. `10.176.0.0/12`
2. Kolam Regional adalah untuk alokasi khusus Wilayah, seperti untuk. `10.176.0.0/15` `us-east-1`
3. Kumpulan unit bisnis adalah alokasi khusus domain di masing-masing. Wilayah AWS Misalnya, unit bisnis keuangan di `us-east-1` Wilayah mungkin memiliki`10.176.0.0/16`.
4. Kolam lingkungan adalah alokasi khusus tujuan untuk lingkungan yang berbeda. Misalnya, unit bisnis keuangan di `us-east-1` Wilayah mungkin memiliki `10.176.0.0/18` lingkungan produksi.

Topologi penyebaran ini mendistribusikan sumber daya IP Address Manager secara geografis sambil mempertahankan kontrol terpusat. Berikut ini adalah fitur-fiturnya:

- IP Address Manager digunakan dalam satu primer Wilayah AWS.
- Wilayah Tambahan terdaftar sebagai wilayah operasi, di mana Manajer Alamat IP dapat mengelola sumber daya.
- Setiap wilayah operasi menerima kumpulan alamat khusus dari kolam tingkat atas.
- Sumber daya di semua wilayah operasi dikelola secara terpusat melalui IP Address Manager di Wilayah utama.
- Setiap kolam Regional memiliki properti lokal yang terkait dengan Wilayahnya untuk membantu Anda mengalokasikan sumber daya dengan benar.

Validasi rentang CIDR tingkat lanjut

Solusi ini dirancang untuk mencegah penyebaran konfigurasi yang tidak valid. Saat Anda menerapkan kumpulan melalui Terraform, berikut ini divalidasi selama fase rencana Terraform:

- Memvalidasi bahwa semua rentang CIDR lingkungan terkandung dalam rentang CIDR unit bisnis induknya
- Mengonfirmasi bahwa semua rentang CIDR unit bisnis terkandung dalam rentang CIDR regional induknya
- Memverifikasi bahwa semua rentang CIDR Regional terkandung dalam rentang CIDR tingkat atas

- Memeriksa rentang CIDR yang tumpang tindih dalam tingkat hierarki yang sama
- Memvalidasi pemetaan lingkungan yang tepat ke unit bisnis masing-masing

Alokasi rentang CIDR

Diagram berikut menunjukkan contoh bagaimana pengembang atau administrator dapat membuat alamat IP baru VPCs dan mengalokasikan alamat IP dari tingkat pool.

Diagram menunjukkan alur kerja berikut:

1. Melalui Konsol Manajemen AWS, the AWS CLI, atau melalui infrastruktur sebagai kode (IAc), pengembang atau administrator meminta rentang CIDR berikutnya yang tersedia di kolam `AY3` lingkungan.
2. IP Address Manager mengalokasikan rentang CIDR berikutnya yang tersedia di pool tersebut ke VPC. `AY3-4` Rentang CIDR ini tidak dapat lagi digunakan.

Otomatisasi dan skala

Solusi ini dirancang untuk skalabilitas sebagai berikut:

- Ekspansi regional — Tambahkan Wilayah baru dengan memperluas konfigurasi Terraform dengan entri kumpulan Regional tambahan.
- Pertumbuhan unit bisnis — Mendukung unit bisnis baru dengan menambahkannya ke peta konfigurasi BU.
- Fleksibilitas lingkungan — Konfigurasikan jenis lingkungan yang berbeda, seperti pengembangan atau produksi, berdasarkan kebutuhan organisasi.
- Dukungan multi-akun — Bagikan kumpulan di semua akun di organisasi Anda melalui AWS RAM.
- Penyediaan VPC otomatis — Integrasikan dengan alur kerja penyediaan VPC untuk mengotomatiskan alokasi rentang CIDR.

Struktur hierarkis juga memungkinkan untuk skala delegasi dan kontrol yang berbeda, seperti berikut ini:

- Administrator jaringan mungkin mengelola kumpulan tingkat atas dan Regional.
- Tim TI unit bisnis mungkin telah mendelegasikan kontrol dari kumpulan masing-masing.

- Tim aplikasi mungkin menggunakan alamat IP dari kumpulan lingkungan yang ditunjuk.

> **Note**
>
> Anda juga dapat mengintegrasikan solusi ini dengan AWS Control Tower Account Factory for Terraform (AFT). Untuk informasi selengkapnya, lihat Integrasi dengan AFT di bagian Informasi tambahan dari pola ini.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Resource Access Manager (AWS RAM) membantu Anda berbagi sumber daya dengan aman Akun AWS untuk mengurangi overhead operasional dan memberikan visibilitas dan auditabilitas.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWSIP Address Manager adalah fitur Amazon VPC. Ini membantu Anda merencanakan, melacak, dan memantau alamat IP untuk AWS beban kerja Anda.

Alat-alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.

Repositori kode

Kode untuk pola ini tersedia di Contoh Terraform Implementasi untuk Hierarchical IPAM pada repositori di. AWS GitHub Struktur repositori meliputi:

- Modul root - Orkestrasi penerapan dan variabel input.
- Modul IPAM — Implementasi inti dari arsitektur yang dijelaskan dalam pola ini.
- Modul tag - Penandaan standar untuk semua sumber daya.

Praktik terbaik

Pertimbangkan praktik terbaik berikut untuk perencanaan jaringan:

- Rencanakan terlebih dahulu - Rencanakan ruang alamat IP Anda secara menyeluruh sebelum penerapan. Untuk informasi selengkapnya, lihat Merencanakan penyediaan alamat IP.
- Hindari rentang CIDR yang tumpang tindih — Pastikan rentang CIDR di setiap level tidak tumpang tindih.
- Ruang penyangga cadangan — Selalu alokasikan rentang CIDR yang lebih besar dari yang dibutuhkan untuk mengakomodasi pertumbuhan.
- Alokasi alamat IP dokumen — Pertahankan dokumentasi strategi alokasi alamat IP Anda.

Pertimbangkan praktik terbaik penerapan berikut:

- Mulailah dengan non-produksi - Terapkan di lingkungan non-produksi terlebih dahulu.
- Gunakan manajemen status Terraform — Terapkan penyimpanan dan penguncian status jarak jauh. Untuk informasi selengkapnya, lihat Penyimpanan status dan penguncian dalam dokumentasi Terraform.
- Menerapkan kontrol versi - Kontrol versi semua kode Terraform.
- Implementasikan CI/CD integrasi — Gunakan pipeline continuous integration dan continuous delivery (CI/CD) untuk penerapan berulang.

Pertimbangkan praktik terbaik operasional berikut:

- Aktifkan impor otomatis - Konfigurasikan kumpulan Manajer Alamat IP untuk menemukan dan mengimpor sumber daya yang ada secara otomatis. Ikuti petunjuk di Edit kolam IPAM untuk mengaktifkan impor otomatis.
- Memantau penggunaan alamat IP - Mengatur alarm untuk ambang batas penggunaan alamat IP. Untuk informasi selengkapnya, lihat Memantau IPAM dengan Amazon CloudWatch.
- Audit secara teratur - Audit penggunaan dan kepatuhan alamat IP secara berkala. Untuk informasi selengkapnya, lihat Melacak penggunaan alamat IP di IPAM.

- Bersihkan alokasi yang tidak digunakan - Lepaskan alokasi alamat IP saat sumber daya dinonaktifkan. Untuk informasi lebih lanjut, lihat Deprovision CIDRs dari pool.

Pertimbangkan praktik terbaik keamanan berikut:

- Terapkan hak istimewa terkecil - Gunakan peran IAM dengan izin minimum yang diperlukan. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan di IAM dan Manajemen Identitas dan akses di IPAM.
- Gunakan kebijakan kontrol layanan — Menerapkan kebijakan kontrol layanan (SCPs) untuk menerapkan penggunaan IP Address Manager di organisasi Anda. Untuk informasi selengkapnya, lihat Menegakkan penggunaan IPAM untuk pembuatan VPC dengan. SCPs
- Kontrol berbagi sumber daya - Kelola ruang lingkup berbagi sumber daya IP Address Manager dengan hati-hati AWS RAM. Untuk informasi selengkapnya, lihat Berbagi kumpulan IPAM menggunakan AWS RAM.
- Menegakkan penandaan - Menerapkan penandaan wajib untuk semua sumber daya yang terkait dengan Manajer Alamat IP. Untuk informasi selengkapnya, lihat Strategi penandaan di bagian Informasi tambahan.

Epik

Siapkan akun administrator yang didelegasikan untuk IP Address Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan AWS Organizations fitur.	Pastikan semua AWS Organizations fitur diaktifkan. Untuk petunjuk, lihat Mengaktifkan semua fitur untuk organisasi AWS Organizations dalam AWS Organizations dokumentasi.	Administrator AWS
Aktifkan berbagi sumber daya di AWS RAM.	Menggunakan AWS CLI, masukkan perintah berikut untuk mengaktifkan berbagi	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS RAM sumber daya untuk organisasi Anda: `aws ram enable-sharing-with-aws-organization` Untuk informasi selengkap nya, lihat Mengaktifkan berbagi sumber daya AWS Organizations dalam AWS RAM dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan administrator untuk IP Address Manager.	Dari akun manajemen organisasi, menggunakan AWS CLI, masukkan perintah berikut, di `123456789012` mana ID akun yang akan mengelola Manajer Alamat IP: `aws ec2 enable-ipam-organization-admin-account \ --delegated-admin-account-id 123456789012` **Note** Biasanya, akun hub jaringan atau jaringan digunakan sebagai administrator yang didelegasikan untuk IP Address Manager. Untuk informasi selengkap nya, lihat Mengintegrasikan IPAM dengan akun di AWS Organisasi dalam dokumenta si Manajer Alamat IP.	Administrator AWS

Menyebarkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan arsitektur jaringan.	Tentukan dan dokumenta sikan arsitektur jaringan Anda, termasuk rentang CIDR untuk Wilayah, unit bisnis, dan lingkungan. Untuk informasi selengkapnya, lihat Merencanakan penyediaan alamat IP dalam dokumentasi Manajer Alamat IP.	Insinyur jaringan
Kloning repositori.	1. Masukkan perintah berikut untuk mengkloning repositori ke workstation lokal Anda: `git clone https://github.com/aws-samples/sample-amazon-vpc-ipam-terraform.git` 2. Masukkan perintah berikut untuk menavigasi ke `terraform` direktori.	DevOps insinyur
Konfigurasikan variabel.	1. Buka file terraform.tfvars. 2. Konfigurasikan variabel untuk penyebaran IP Address Manager Anda. Untuk contoh file yang sudah selesai, lihat terraform.tfvars.example di repositori kode.	Insinyur jaringan, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Simpan dan tutup file terraform.tfvars.	

Menyebarkan sumber daya Manajer Alamat IP.	1. Masukkan perintah berikut untuk menginisialisasi Terraform:
	`terraform init`
	2. Masukkan perintah berikut untuk menghasilkan rencana eksekusi:
	`terraform plan`
	3. Tinjau rencana, dan validasi sumber daya dan komponen infrastruktur yang akan dibuat.
	4. Masukkan perintah berikut untuk menyebarkan infrastruktur:
	`terraform apply`
	5. Saat diminta, masukkan ya untuk mengonfirmasi penerapan.
	6. Tunggu hingga penerapan selesai.
		Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan.	1. Masuk ke Anda Akun AWS, lalu buka konsol IP Address Manager. 2. Di panel navigasi, pilih Pools. 3. Dalam daftar pool, verifikasi bahwa pool yang Anda konfigurasikan telah dibuat. 4. Buka konsol AWS RAM. 5. Verifikasi bahwa sumber daya Manajer Alamat IP telah dibagikan. Ikuti petunjuk di Melihat pembagian sumber daya yang Anda buat dalam AWS RAM dokumentasi.	AWS Umum, Insinyur jaringan

Membuat VPCs dan mengatur pemantauan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Ikuti langkah-langkah di Buat VPC di dokumentasi Amazon VPC. Saat Anda mencapai langkah untuk memilih rentang CIDR untuk VPC, alokasikan rentang berikutnya yang tersedia dari salah satu Regional, unit bisnis, dan kolam lingkungan Anda.	AWS Umum, Administrator jaringan, Insinyur jaringan
Validasi alokasi rentang CIDR.	1. Buka konsol IP Address Manager.	AWS Umum, Administrator jaringan, Insinyur jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di panel navigasi, pilih Dasbor. 3. Di dasbor, verifikasi bahwa rentang CIDR telah dialokasikan ke VPC yang Anda buat.	
Pantau Manajer Alamat IP.	Konfigurasikan pemantaua n dan alarm yang terkait dengan alokasi sumber daya IP Address Manager. Untuk informasi dan petunjuk selengkapnya, lihat Memantau IPAM dengan Amazon CloudWatch dan Memantau penggunaan CIDR berdasark an sumber daya dalam dokumentasi Manajer Alamat IP.	AWS Umum
Menegakkan penggunaan IP Address Manager.	Buat kebijakan kontrol layanan (SCP) AWS Organizat ions yang mengharuskan anggota di organisasi Anda untuk menggunakan IP Address Manager saat mereka membuat VPC. Untuk petunjuknya, lihat Menerapka n penggunaan IPAM untuk pembuatan VPC dengan dokumentasi SCPs Manajer Alamat IP.	AWS umum, administrator AWS

Pemecahan masalah

Isu	Solusi
Terraform gagal dengan sumber daya Manajer Alamat IP tidak ditemukan	Pastikan akun administrator IP Address Manager didelegasikan dengan benar dan AWS Penyedia Anda diautentikasi ke akun tersebut.
Alokasi rentang CIDR gagal	Periksa apakah rentang CIDR yang diminta sesuai dengan rentang yang tersedia dari kumpulan Manajer Alamat IP dan tidak tumpang tindih dengan alokasi yang ada.
AWS RAM berbagi masalah	Verifikasi bahwa berbagi sumber daya diaktifkan untuk AWS Organisasi Anda. Verifikasi bahwa prinsipal yang benar, organisasi Amazon Resource Name (ARN), digunakan dalam pembagian AWS RAM .
Kesalahan validasi hierarki kumpulan	Pastikan bahwa rentang CIDR kolam anak terkandung dengan benar dalam rentang CIDR kolam induk mereka dan tidak tumpang tindih dengan kolam saudara kandung.
Batas kuota IP Address Manager terlampaui	Minta peningkatan kuota untuk kumpulan IP Address Manager. Untuk informasi selengkapnya, lihat Meminta peningkatan kuota di Panduan Pengguna Service Quotas.

Sumber daya terkait

Layanan AWS dokumentasi

- Dokumentasi Manajer Alamat IP VPC Amazon
- AWS Resource Access Manager dokumentasi
- AWS Organizations dokumentasi

AWS posting blog

- Mengelola kumpulan IP di seluruh VPCs dan Wilayah menggunakan Amazon VPC IP Address Manager
- Manajemen alamat jaringan dan audit dalam skala besar dengan Amazon VPC IP Address Manager

Video dan tutorial

- AWS re:invent 2022: Praktik terbaik untuk desain VPC Amazon dan IPAM (0) NET31
- AWS Re:invent 2022: Desain VPC tingkat lanjut dan kemampuan baru (01) NET4

Informasi tambahan

Integrasi dengan AFT

Anda dapat mengintegrasikan solusi ini dengan AWS Control Tower Account Factory for Terraform (AFT) untuk memastikan bahwa akun yang baru disediakan secara otomatis menerima konfigurasi jaringan yang tepat. Dengan menerapkan solusi IPAM ini di akun hub jaringan Anda, akun baru yang dibuat melalui AFT dapat mereferensikan kumpulan Manajer Alamat IP bersama saat Anda membuat. VPCs

Contoh kode berikut menunjukkan integrasi AFT dalam kustomisasi akun dengan menggunakan AWS Systems Manager Parameter Store:

```
# Get the IP Address Manager pool ID from Parameter Store
data "aws_ssm_parameter" "dev_ipam_pool_id" {
  name = "/org/network/ipam/finance/dev/pool-id"
}

# Create a VPC using the IP Address Manager pool
resource "aws_vpc" "this" {
  ipv4_ipam_pool_id   = data.aws_ssm_parameter.dev_ipam_pool_id.value
  ipv4_netmask_length = 24

  tags = {
    Name = "aft-account-vpc"
  }
}
```

Strategi penandaan

Solusinya menerapkan strategi penandaan yang komprehensif untuk memfasilitasi manajemen sumber daya. Contoh kode berikut menunjukkan bagaimana itu digunakan:

```
# Example tag configuration
module "tags" {
  source = "./modules/tags"

  # Required tags
  product_name  = "enterprise-network"
  feature_name  = "ipam"
  org_id        = "finance"
  business_unit = "network-operations"
  owner         = "network-team"
  environment   = "prod"
  repo          = "https://github.com/myorg/ipam-terraform"
  branch        = "main"
  cost_center   = "123456"
  dr_tier       = "tier1"

  # Optional tags
  optional_tags = {
    "project"    = "network-modernization"
    "stack_role" = "infrastructure"
  }
}
```

Tag ini secara otomatis diterapkan ke semua sumber IP Address Manager. Ini memfasilitasi tata kelola yang konsisten, alokasi biaya, dan manajemen sumber daya.

Kustomisasi CloudWatch peringatan Amazon untuk AWS Network Firewall

Jason Owens, Amazon Web Services

Ringkasan

Pola ini membantu Anda menyesuaikan CloudWatch peringatan Amazon yang dihasilkan oleh AWS Network Firewall. Anda dapat menggunakan aturan yang telah ditentukan sebelumnya atau membuat aturan khusus yang menentukan pesan, metadata, dan tingkat keparahan peringatan. Anda kemudian dapat menindaklanjuti peringatan ini atau mengotomatiskan tanggapan oleh layanan Amazon lainnya, seperti Amazon. EventBridge

Dalam pola ini, Anda menghasilkan aturan firewall yang kompatibel dengan Suricata. Suricata adalah mesin pendeteksi ancaman sumber terbuka. Pertama-tama Anda membuat aturan sederhana dan kemudian mengujinya untuk mengonfirmasi bahwa CloudWatch peringatan dibuat dan dicatat. Setelah Anda berhasil menguji aturan, Anda memodifikasinya untuk menentukan pesan kustom, metadata, dan tingkat keparahan, dan Anda kemudian menguji sekali lagi untuk mengonfirmasi pembaruan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi di workstation Linux, macOS, atau Windows Anda. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui versi terbaru AWS CLI.
- AWS Network Firewall diinstal dan dikonfigurasi untuk menggunakan CloudWatch Log. Untuk informasi selengkapnya, lihat Mencatat lalu lintas jaringan dari AWS Network Firewall.
- Instans Amazon Elastic Compute Cloud (Amazon EC2) di subnet pribadi virtual private cloud (VPC) yang dilindungi oleh Network Firewall.

Versi produk

- Untuk versi 1 AWS CLI, gunakan 1.18.180 atau yang lebih baru. Untuk versi 2 AWS CLI, gunakan 2.1.2 atau yang lebih baru.
- File klasifikasi.config dari Suricata versi 5.0.2. Untuk salinan file konfigurasi ini, lihat bagian Informasi tambahan.

Arsitektur

Diagram arsitektur menunjukkan alur kerja berikut:

1. EC2 Instans Amazon di subnet pribadi membuat permintaan dengan menggunakan curl atau Wget.
2. Network Firewall memproses lalu lintas dan menghasilkan peringatan.
3. Network Firewall mengirimkan peringatan yang dicatat ke CloudWatch Log.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Network Firewalladalah firewall jaringan stateful, dikelola, dan layanan deteksi dan pencegahan intrusi untuk cloud pribadi virtual () VPCs di. AWS Cloud

Alat-alat lainnya

- curl adalah alat baris perintah sumber terbuka dan perpustakaan.
- GNU Wget adalah alat baris perintah gratis.

Epik

Buat aturan firewall dan grup aturan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan.	1. Di editor teks, buat daftar aturan yang ingin Anda	Administrator sistem AWS, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	tambahkan ke firewall. Setiap aturan harus berada pada baris terpisah. Nilai dalam `classtype` parameter berasal dari file konfigurasi klasifikasi Suricata default. Untuk isi file konfigurasi lengkap, lihat bagian Informasi tambahan. Berikut ini adalah dua contoh aturan.
	`alert http any any -> any any (content:"badstuff"; classtype:misc-activity; sid:3; rev:1;)`
	`alert http any any -> any any (content:"morebadstuff"; classtype:bad-unknown; sid:4; rev:1;)`
	2. Simpan aturan dalam file bernama `custom.rules` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat grup aturan.	Di AWS CLI, masukkan perintah berikut. Ini menciptakan grup aturan.
	
	`# aws network-firewall create-rule-group \ --rule-group-name custom --type STATEFUL \ --capacity 10 --rules file://custom.rules \ --tags Key=environment,Value=development`
	
	Berikut ini adalah output contoh. Catat`RuleGroup Arn`, yang Anda butuhkan di langkah selanjutnya.
	
	`{ "UpdateToken": "4f998d72-973c-490a-bed2-fc3460547e23", "RuleGroupResponse": { "RuleGroupArn": "arn:aws:network-firewall:us-east-2:1234567890:stateful-rulegroup/custom", "RuleGroupName": "custom", "RuleGroupId": "238a8259-9eaf-48bb-90af-5e690cf8c48b", "Type": "STATEFUL",`
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Capacity": 10,
	"RuleGrou
	pStatus": "ACTIVE",
	"Tags": [
	{
	"Key":
	"environment",
	"Value":
	"development"
	}
	]
	}</pre>	

Perbarui kebijakan firewall

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Dapatkan ARN dari kebijakan firewall.	Di AWS CLI, masukkan perintah berikut. Ini mengembalikan Nama Sumber Daya Amazon (ARN) dari kebijakan firewall. Rekam ARN untuk digunakan nanti dalam pola ini.
	<pre># aws network-firewall
	describe-firewall \
	--firewall-name
	aws-network-firewall-
	anfw \
	--query 'Firewall
	.FirewallPolicyArn'</pre>
	Berikut ini adalah contoh ARN yang dikembalikan oleh perintah ini.
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`"arn:aws:network-firewall:us-east-2:1234567890:firewall-policy/firewall-policy-anfw"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui kebijakan firewall.	Dalam editor teks, salin tempel kode berikut. Ganti `<RuleGroupArn>` dengan nilai yang Anda rekam di epik sebelumnya. Simpan file sebagai `firewall-policy-anfw.json` . `{ "StatelessDefaultActions": [ "aws:forward_to_sfe" ], "StatelessFragmentDefaultActions": [ "aws:forward_to_sfe" ], "StatefulRuleGroupReferences": [ { "ResourceArn": "<RuleGroupArn>" } ] }` Masukkan perintah berikut di file AWS CLI. Perintah ini memerlukan token pembaruan untuk menambahkan aturan baru. Token digunakan untuk mengonfirmasi bahwa kebijakan tidak berubah sejak terakhir kali Anda mengambil nya.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`UPDATETOKEN=(`aws network-firewall describe-firewall-policy \ --firewall-policy-name firewall-policy-anfw \ --output text --query UpdateToken`)`
	
	`aws network-firewall update-firewall-policy \ --update-token $UPDATETOKEN \ --firewall-policy-name firewall-policy-anfw \ --firewall-policy file://firewall-policy-anfw.json`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan pembaruan kebijakan.	(Opsional) Jika Anda ingin mengonfirmasi aturan ditambahkan dan melihat format kebijakan, masukkan perintah berikut di AWS CLI.
	`# aws network-firewall describe-firewall-policy \ --firewall-policy-name firewall-policy-anfw \ --query FirewallPolicy`
	Berikut ini adalah output contoh.
	`{ "StatelessDefaultActions": [ "aws:forward_to_sfe" ], "StatelessFragmentDefaultActions": [ "aws:forward_to_sfe" ], "StatefulRuleGroupReferences": [ { "ResourceArn": "arn:aws:network-firewall:us-east-2:1234567890:stateful-rulegroup/custom" }`
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>] }</pre>	

Fungsionalitas peringatan uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hasilkan peringatan untuk pengujian.	1. Masuk ke workstation pengujian dalam subnet firewall.
	2. Masukkan perintah yang seharusnya menghasilkan peringatan. Misalnya, Anda dapat menggunakan `wget` atau`curl`.
	<pre>wget -U "badstuff"
	 http://www.amazon.
	com -o /dev/null</pre>
	<pre>curl -A "morebads
	tuff" http://ww
	w.amazon.com -o /
	dev/null</pre>
		Administrator sistem AWS

Validasi bahwa peringatan dicatat.	1. Buka konsol CloudWatch . 2. Arahkan ke grup log dan streaming yang benar. Untuk informasi selengkapnya, lihat Melihat data log yang dikirim ke CloudWatch Log (dokumentasi CloudWatch Log).	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Konfirmasikan peristiwa yang dicatat mirip dengan contoh berikut. Contoh hanya menunjukkan bagian peringatan yang relevan.
	
	Contoh 1
	
	`"alert": { "action": "allowed", "signature_id": 3, "rev": 1, "signature": "", "category": "Misc activity", "severity": 3 }`
	
	Contoh 2
	
	`"alert": { "action": "allowed", "signature_id": 4, "rev": 1, "signature": "", "category": "Potentially Bad Traffic", "severity": 2 }`	

Perbarui aturan firewall dan grup aturan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui aturan firewall.	1. Dalam editor teks, buka `custom.rules` file. 2. Ubah aturan pertama menjadi serupa dengan yang berikut ini. Aturan ini harus dimasukkan pada satu baris dalam file. `alert http any any -> any any (msg:"Watch out - Bad Stuff!!"; content:"badstuff"; classtype:misc-activity; priority:2; sid:3; rev:2; metadata:custom-field-2 Danger!, custom-field More Info;)` Ini membuat perubahan berikut pada aturan: • Menambahkan string msg (situs web Suricata) yang menyediakan informasi teks tentang tanda tangan atau peringatan. Dalam peringatan yang dihasilkan, ini memetakan ke tanda tangan. • Menyesuaikan prioritas default (situs web Suricata) `misc-activity` dari 3 hingga 2.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk nilai default dari berbagai`classtypes` , lihat bagian Informasi tambahan. • Menambahkan metadata khusus (situs web Suricata) ke peringata n. Ini adalah informasi tambahan yang ditambahkan ke tanda tangan. Disarankan agar Anda menggunakan pasangan kunci-nilai. • Mengubah rev (situs web Suricata) dari 1 menjadi 2. Ini mewakili versi tanda tangan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui grup aturan.	Di AWS CLI, jalankan perintah berikut. Gunakan ARN kebijakan firewall Anda. Perintah ini mendapatk an token pembaruan dan memperbarui grup aturan dengan perubahan aturan.
	`# UPDATETOKEN=(`aws network-firewall \ describe-rule-group \ --rule-group-arn arn:aws:network-fi rewall:us-east-2:1 23457890:stateful- rulegroup/custom \ --output text --query UpdateToken`)`
	`# aws network-firewall update-rule-group \ --rule-group-arn arn:aws:network-fi rewall:us-east-2:1 234567890:stateful- rulegroup/custom \ --rules file://cu stom.rules \ --update-token $UPDATETOKEN`
	Berikut ini adalah output contoh.
	`{`
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"UpdateToken": "7536939f-6a1d-414c-96d1-bb28110996ed", "RuleGroupResponse": { "RuleGroupArn": "arn:aws:network-firewall:us-east-2:1234567890:stateful-rulegroup/custom", "RuleGroupName": "custom", "RuleGroupId": "238a8259-9eaf-48bb-90af-5e690cf8c48b", "Type": "STATEFUL", "Capacity": 10, "RuleGroupStatus": "ACTIVE", "Tags": [{ "Key": "environment", "Value": "development" }] } }	

Uji fungsionalitas peringatan yang diperbarui

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan peringatan untuk pengujian.	1. Masuk ke workstation pengujian dalam subnet firewall.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Masukkan perintah yang seharusnya menghasilkan peringatan. Misalnya, Anda dapat menggunakan`curl`. `curl -A "badstuff" http://www.amazon.com -o /dev/null`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi peringatan diubah.	1. Buka konsol CloudWatch . 2. Arahkan ke grup log dan streaming yang benar. 3. Konfirmasikan peristiwa yang dicatat mirip dengan contoh berikut. Contoh hanya menunjukkan bagian peringatan yang relevan. `"alert": { "action": "allowed", "signature_id": 3, "rev": 2, "signature": "Watch out - Bad Stuff!!", "category": "Misc activity", "severity": 2, "metadata": { "custom-field": [ "More Info" ], "custom-field-2": [ "Danger!" ] } }`	Administrator sistem AWS

Sumber daya terkait

Referensi

- Kirim peringatan dari AWS Network Firewall ke saluran Slack (Panduan AWS Preskriptif)
- Meningkatkan pencegahan ancaman AWS dengan Suricata (AWS posting blog)
- Model penyebaran untuk AWS Network Firewall (posting AWS blog)
- Pekerjaan kunci meta Suricata (dokumentasi Suricata)

Tutorial dan video

- AWS Network Firewall lokakarya

Informasi tambahan

Berikut ini adalah file konfigurasi klasifikasi dari Suricata 5.0.2. Klasifikasi ini digunakan saat membuat aturan firewall.

```
# config classification:shortname,short description,priority

config classification: not-suspicious,Not Suspicious Traffic,3
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1
config classification: successful-user,Successful User Privilege Gain,1
config classification: attempted-admin,Attempted Administrator Privilege Gain,1
config classification: successful-admin,Successful Administrator Privilege Gain,1

# NEW CLASSIFICATIONS
config classification: rpc-portmap-decode,Decode of an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3
config classification: suspicious-filename-detect,A suspicious filename was detected,2
config classification: suspicious-login,An attempted login using a suspicious username
 was detected,2
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: trojan-activity,A Network Trojan was detected, 1
```

```
config classification: unusual-client-port-connection,A client was using an unusual
 port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: non-standard-protocol,Detection of a non-standard protocol or
 event,2
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web
 application,2
config classification: web-application-attack,Web Application Attack,1
config classification: misc-activity,Misc activity,3
config classification: misc-attack,Misc Attack,2
config classification: icmp-event,Generic ICMP event,3
config classification: inappropriate-content,Inappropriate Content was Detected,1
config classification: policy-violation,Potential Corporate Privacy Violation,1
config classification: default-login-attempt,Attempt to login by a default username and
 password,2

# Update
config classification: targeted-activity,Targeted Malicious Activity was Detected,1
config classification: exploit-kit,Exploit Kit Activity Detected,1
config classification: external-ip-check,Device Retrieving External IP Address
 Detected,2
config classification: domain-c2,Domain Observed Used for C2 Detected,1
config classification: pup-activity,Possibly Unwanted Program Detected,2
config classification: credential-theft,Successful Credential Theft Detected,1
config classification: social-engineering,Possible Social Engineering Attempted,2
config classification: coin-mining,Crypto Currency Mining Activity Detected,2
config classification: command-and-control,Malware Command and Control Activity
 Detected,1
```

Menerapkan sumber daya di AWS Wavelength Zona dengan menggunakan Terraform

Zahoor Chaudhrey dan Luca Iannario, Amazon Web Services

Ringkasan

AWS Wavelengthmembantu Anda membangun infrastruktur yang dioptimalkan untuk aplikasi Multi-Access Edge Computing (MEC). Wavelength Zones AWS adalah penyebaran infrastruktur yang AWS menanamkan layanan komputasi dan penyimpanan dalam jaringan 5G penyedia layanan komunikasi (CSP). Lalu lintas aplikasi dari perangkat 5G mencapai server aplikasi yang berjalan di Wavelength Zones tanpa meninggalkan jaringan telekomunikasi. Berikut ini memfasilitasi konektivitas jaringan melalui Wavelength:

- Virtual private cloud (VPCs) — VPCs dalam Akun AWS kaleng diperluas untuk menjangkau beberapa Availability Zone, termasuk Wavelength Zones. Instans Amazon Elastic Compute Cloud (Amazon EC2) dan layanan terkait muncul sebagai bagian dari VPC Regional Anda. VPCs dibuat dan dikelola di Amazon Virtual Private Cloud (Amazon VPC).
- Carrier gateway — Gateway operator memungkinkan konektivitas dari subnet di Wavelength Zone ke jaringan CSP, internet, atau melalui jaringan CSP. Wilayah AWS Gerbang pembawa melayani dua tujuan. Ini memungkinkan lalu lintas masuk dari jaringan CSP di lokasi tertentu, dan memungkinkan lalu lintas keluar ke jaringan telekomunikasi dan internet.

Pola ini dan kode Terraform terkait membantu Anda meluncurkan sumber daya, seperti EC2 instans Amazon, volume Amazon Elastic Block Store (Amazon EBS), subnet, dan gateway operator VPCs, di Zona Wavelength.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Lingkungan pengembangan terintegrasi (IDE)
- Ikut serta ke Zona Wavelength target
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Terraform versi 1.8.4 atau yang lebih baru, diinstal (dokumentasi Terraform)
- AWS Penyedia Terraform versi 5.32.1 atau yang lebih baru, dikonfigurasi (dokumentasi Terraform)

- Git, diinstal (GitHub)
- Izin untuk membuat sumber daya Amazon VPC, Wavelength, dan Amazon EC2

Batasan

Tidak semua Wilayah AWS mendukung Wavelength Zones. Untuk informasi selengkapnya, lihat Zona Wavelength yang Tersedia dalam dokumentasi Wavelength.

Arsitektur

Diagram berikut menunjukkan bagaimana Anda dapat membuat subnet dan AWS sumber daya di Wavelength Zone. VPCs yang berisi subnet di Wavelength Zone dapat terhubung ke gateway operator. Gateway operator memungkinkan Anda untuk terhubung ke sumber daya berikut:

- Perangkat 4G/LTE dan 5G di jaringan operator telekomunikasi.
- Akses nirkabel tetap untuk mitra Wavelength Zone tertentu. Untuk informasi selengkapnya, lihat Multi-akses AWS Wavelength.
- Lalu lintas keluar ke sumber daya internet publik.

Alat

Layanan AWS

- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS
- AWS Wavelengthmemperluas AWS Cloud infrastruktur ke jaringan 5G penyedia telekomunikasi. Ini membantu Anda membangun aplikasi yang memberikan latensi sangat rendah ke perangkat seluler dan pengguna akhir.

Alat-alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di GitHub Creating AWS Wavelength Infrastructure menggunakan repositori Terraform. Kode Terraform menyebarkan infrastruktur dan sumber daya berikut:

- VPC
- Zona Wavelength
- Subnet kemaluan di Zona Wavelength
- Gerbang pembawa di Zona Wavelength
- EC2 Contoh Amazon di Zona Wavelength

Praktik terbaik

- Sebelum menerapkan, konfirmasikan bahwa Anda menggunakan versi terbaru Terraform dan. AWS CLI
- Gunakan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) untuk menyebarkan IAc. Untuk informasi selengkapnya, lihat Praktik terbaik untuk mengelola file Status Terraform di AWS CI/CD Pipeline di Blog. AWS

Epik

Penyediaan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Masukkan perintah berikut untuk mengkloning Creating AWS Wavelength Infrastructure menggunakan repositori Terraform ke lingkungan Anda. `git clone git@github.com:aws-samples/terraform-waveleng`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`th-infrastructure.git`	
Perbarui variabel.	1. Arahkan ke repositori kloning. `cd terraform-wavelength-infrastructure` 2. Menggunakan editor teks apa pun, buat file bernama terraform.tfvars di direktori root. 3. Buat variabel berikut dan masukkan nilainya: • `region = <enter Region name>` • `vpc_cidr = <enter CIDR block used by VPC>` • `wavelength_subnet_cidr = <enter CIDR block for the subnet in the Wavelength Zone>` • `availabilityzone_wavelength = <enter Wavelength Zone name>` 4. Simpan file terraform.tfvars.	DevOps insinyur, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi konfigurasi.	Masukkan perintah berikut untuk menginisialisasi direktori kerja. `terraform init`	DevOps insinyur, Terraform
Pratinjau paket Terraform.	Masukkan perintah berikut untuk membandingkan status target dengan keadaan AWS lingkungan Anda saat ini. Perintah ini menghasilkan pratinjau sumber daya yang akan dikonfigurasi. `terraform plan`	DevOps insinyur, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi dan terapkan.	1. Tinjau perubahan konfigurasi dalam paket Terraform dan konfirmasikan bahwa Anda ingin menerapkan perubahan ini. 2. Masukkan perintah berikut untuk menerapkan rencana dan membuat infrastruktur. `terraform apply` 3. Masuk `yes` untuk melanjutkan. Terraform membuat arsitektur yang dideklarasikan dalam file konfigurasi. Untuk informasi selengkapnya tentang arsitektur, lihat bagian Arsitektur target dari pola ini.	DevOps insinyur, Terraform

Validasi dan bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi penyebaran infrastruktur.	1. Jika Anda belum memiliki EC2 instans Amazon di subnet publik di Anda Wilayah AWS, buat satu. Untuk petunjuk, lihat Meluncurkan instance Linux Anda atau Luncurkan instance Windows Anda. Anda akan menggunakan instance ini untuk menguji	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	konektivitas dari Wilayah AWS ke Wavelength Zone. 2. Uji konektivitas dari instance Wilayah AWS ke instance di Wavelength Zone. Untuk petunjuk, lihat Menguji konektivitas dalam dokumentasi Wavelength.	
(Opsional) Bersihkan infrastruktur.	Jika Anda perlu menghapus semua sumber daya yang disediakan oleh Terraform, lakukan hal berikut: 1. Masukkan perintah berikut. `terraform destroy` 2. Masukkan `yes` untuk mengonfirmasi.	DevOps insinyur, Terraform

Pemecahan masalah

Isu	Solusi
Konektivitas ke EC2 instans Amazon di. Wilayah AWS	Lihat Memecahkan masalah saat menghubungkan ke instans Linux Anda atau Memecahkan masalah saat menghubungkan ke instans Windows Anda.
Konektivitas ke EC2 instans Amazon di Wavelength Zone.	Lihat Memecahkan masalah konektivitas SSH atau RDP ke EC2 instans saya yang diluncurkan di Wavelength Zone.

Isu	Solusi
Kapasitas di Zona Wavelength.	Lihat Kuota dan pertimbangan untuk Zona Wavelength.
Konektivitas seluler atau operator dari jaringan operator ke Wilayah AWS.	1. Verifikasi bahwa gateway operator beroperasi. Lakukan hal-hal berikut: a. Buka konsol Amazon VPC. b. Di panel navigasi, pilih Your VPCs. c. Pilih VPC yang berisi Wavelength Zone. d. Pada panel Detail, untuk gateway Operator, konfirmasikan bahwa nilainya dilampirkan. 2. Verifikasi bahwa setiap alamat IP elastis yang dilampirkan ke instance di Wavelength Zone beroperasi. Lakukan hal-hal berikut: a. Buka EC2 konsol Amazon. b. Di panel navigasi, pilih Instans. c. Pilih instance di Wavelength Zone. d. Pilih tab Jaringan. e. Konfirmasikan bahwa elastic network interface memiliki alamat IP Elastis yang terpasang. 3. Hubungi tim dukungan jaringan operator.

Sumber daya terkait

- Apa itu AWS Wavelength?
- Bagaimana cara AWS Wavelength kerja
- Ketahanan di AWS Wavelength

Migrasikan data DNS secara massal ke zona host pribadi Amazon Route 53

Ram Kandaswamy, Amazon Web Services

Ringkasan

Insinyur jaringan dan administrator cloud memerlukan cara yang efisien dan sederhana untuk menambahkan catatan Domain Name System (DNS) ke zona yang dihosting pribadi di Amazon Route 53. Menggunakan pendekatan manual untuk menyalin entri dari lembar kerja Microsoft Excel ke lokasi yang sesuai di konsol Route 53 membosankan dan rawan kesalahan. Pola ini menjelaskan pendekatan otomatis yang mengurangi waktu dan upaya yang diperlukan untuk menambahkan beberapa catatan. Ini juga menyediakan serangkaian langkah yang dapat diulang untuk beberapa pembuatan zona yang dihosting.

Pola ini menggunakan Amazon Simple Storage Service (Amazon S3) untuk menyimpan catatan. Untuk bekerja dengan data secara efisien, pola menggunakan format JSON karena kesederhanaan dan kemampuannya untuk mendukung kamus Python `dict` (tipe data).

> **Note**
>
> Jika Anda dapat membuat file zona dari sistem Anda, pertimbangkan untuk menggunakan fitur impor Route 53 sebagai gantinya.

Prasyarat dan batasan

Prasyarat

- Lembar kerja Excel yang berisi catatan zona yang dihosting pribadi
- Keakraban dengan berbagai jenis catatan DNS seperti catatan A, catatan Name Authority Pointer (NAPTR), dan catatan SRV (lihat Jenis rekaman DNS yang didukung)
- Keakraban dengan bahasa Python dan perpustakaannya

Batasan

- Pola tidak memberikan cakupan yang luas untuk semua skenario kasus penggunaan. Misalnya, panggilan change_resource_record_sets tidak menggunakan semua properti API yang tersedia.

- Di lembar kerja Excel, nilai di setiap baris diasumsikan unik. Beberapa nilai untuk setiap nama domain yang memenuhi syarat (FQDN) diharapkan muncul di baris yang sama. Jika itu tidak benar, Anda harus memodifikasi kode yang disediakan dalam pola ini untuk melakukan penggabungan yang diperlukan.
- Pola ini menggunakan AWS SDK for Python (Boto3) untuk memanggil layanan Route 53 secara langsung. Anda dapat menyempurnakan kode untuk menggunakan CloudFormation pembungkus AWS untuk `update_stack` perintah `create_stack` dan, dan menggunakan nilai JSON untuk mengisi sumber daya template.

Arsitektur

Tumpukan teknologi

- Route 53 zona host pribadi untuk merutekan lalu lintas
- Amazon S3 untuk menyimpan file JSON output

Alur kerja terdiri dari langkah-langkah ini, seperti yang diilustrasikan dalam diagram sebelumnya dan dibahas di bagian Epik:

1. Unggah lembar kerja Excel yang memiliki informasi kumpulan rekaman ke bucket S3.
2. Buat dan jalankan skrip Python yang mengonversi data Excel ke format JSON.
3. Baca catatan dari bucket S3 dan bersihkan datanya.
4. Buat kumpulan rekaman di zona host pribadi Anda.

Alat

- Route 53 - Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan yang menangani pendaftaran domain, perutean DNS, dan pemeriksaan kesehatan.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web.

Epik

Siapkan data untuk otomatisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file Excel untuk catatan Anda.	Gunakan catatan yang Anda ekspor dari sistem Anda saat ini untuk membuat lembar kerja Excel yang memiliki kolom yang diperlukan untuk catatan, seperti nama domain yang sepenuhnya memenuhi syarat (FQDN), jenis rekaman, Waktu untuk Hidup (TTL), dan nilai. Untuk catatan NAPTR dan SRV, nilainya adalah kombinasi dari beberapa properti, jadi gunakan `concat` metode Excel untuk menggabungkan properti ini. FqdnN Recor e Nilai TTL somet .exam org A 1.1.1.1 900	Insinyur data, keterampilan Excel
Verifikasi lingkungan kerja.	Di IDE Anda, buat file Python untuk mengonversi lembar kerja input Excel ke format JSON. (Alih-alih IDE, Anda juga dapat menggunakan SageMaker notebook Amazon	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	untuk bekerja dengan kode Python.)
	
	Verifikasi bahwa versi Python yang Anda gunakan adalah versi 3.7 atau yang lebih baru.
	
	`python3 --version`
	
	Instal paket panda.
	
	`pip3 install pandas --user`	

Ubah data lembar kerja Excel ke JSON.	Buat file Python yang berisi kode berikut untuk dikonversi dari Excel ke JSON.
	
	`import pandas as pd`
	`data=pd.read_excel('./Book1.xls')`
	`data.to_json(path_or_buf='my.json', orient='records')`
	
	di mana `Book1` adalah nama lembar kerja Excel dan `my.json` merupakan nama file JSON keluaran.
		Insinyur data, keterampilan Python

Unggah file JSON ke bucket S3.	Unggah `my.json` file ke bucket S3. Untuk informasi selengkapnya, lihat Membuat bucket di dokumentasi Amazon S3.	Pengembang aplikasi

Sisipkan catatan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat zona host pribadi.	Gunakan create_hosted_zone API dan kode contoh Python berikut untuk membuat zona host pribadi. Ganti parameter`hostedZoneName`, `vpcRegion`, dan `vpcId` dengan nilai Anda sendiri.
	`import boto3`
	`import random`
	`hostedZoneName ="xxx"`
	`vpcRegion = "us-east-1"`
	`vpcId="vpc-xxxx"`
	`route53_client = boto3.client('route53')`
	`response = route53_client.create_hosted_zone(`
	`Name= hostedZoneName,`
	`VPC={`
	`'VPCRegion: vpcRegion,`
	`'VPCId': vpcId`
	`},`
	`CallerReference=str(random.random()*100000),`
	`HostedZoneConfig={`
	`'Comment': "private hosted zone`
		Arsitek cloud, administrator Jaringan, keterampilan Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>created by automatio
	n",
	'PrivateZ
	one': True
	}
	)
	print(response)</pre>
	Anda juga dapat menggunakan alat infrastruktur sebagai kode (IAc) seperti AWS CloudFormation untuk mengganti langkah-langkah ini dengan templat yang membuat tumpukan dengan sumber daya dan properti yang sesuai.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ambil detail sebagai kamus dari Amazon S3.	Gunakan kode berikut untuk membaca dari bucket S3 dan untuk mendapatkan nilai JSON sebagai kamus Python. `fileobj = s3_client .get_object( Bucket=bu cket_name, Key='my.json' ) filedata = fileobj[' Body'].read() contents = filedata. decode('utf-8') json_content=json. loads(contents) print(json_content )` dimana `json_content` berisi kamus Python.	Pengembang aplikasi, keterampilan Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bersihkan nilai data untuk spasi dan karakter Unicode.	Sebagai langkah pengamanan untuk memastikan kebenaran data, gunakan kode berikut untuk melakukan operasi strip pada nilai di`json_content` . Kode ini menghapus karakter spasi di bagian depan dan akhir setiap string. Hal ini juga menggunakan `replace` metode untuk menghapus hard (non-breaking) spasi (`\xa0`karakter).
	`for item in json_content:`
	`  fqn_name = unicodedata.normalize("NFKD",item["FqdnName"].replace("u'", "'").replace('\xa0', '').strip())`
	`  rec_type = item["RecordType"].replace('\xa0', '').strip()`
	`  res_rec = {`
	`  'Value': item["Value"].replace('\xa0', '').strip()`
	`  }`
		Pengembang aplikasi, keterampilan Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sisipkan catatan.	Gunakan kode berikut sebagai bagian dari `for` loop sebelumnya. `change_response = route53_client.change_resource_record_sets( HostedZoneId="xxxxxxxx", ChangeBatch={ 'Comment': 'Created by automation', 'Changes': [ { 'Action': 'UPSERT', 'ResourceRecordSet': { 'Name': fqn_name, 'Type': rec_type, 'TTL': item["TTL"], 'ResourceRecords': res_rec } } ] } )`	Pengembang aplikasi, keterampilan Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Di `xxxxxxx` mana ID zona yang dihosting dari langkah pertama epik ini.	

Sumber daya terkait

Referensi

- Membuat catatan dengan mengimpor file zona (dokumentasi Amazon Route 53)
- metode create_hosted_zone (dokumentasi Boto3)
- metode change_resource_record_sets (dokumentasi Boto3)

Tutorial dan video

- Tutorial Python (dokumentasi Python)
- Desain DNS menggunakan Amazon Route 53 (YouTube video, AWS Online Tech Talks)

Ubah header HTTP saat Anda bermigrasi dari F5 ke Application Load Balancer di AWS

Sachin Trivedi, Amazon Web Services

Ringkasan

Saat Anda memigrasikan aplikasi yang menggunakan penyeimbang Load F5 ke Amazon Web Services (AWS) dan ingin menggunakan Application Load Balancer di AWS, memigrasikan aturan F5 untuk modifikasi header adalah masalah umum. Application Load Balancer tidak mendukung modifikasi header, tetapi Anda dapat menggunakan Amazon CloudFront sebagai jaringan pengiriman konten (CDN) dan Lambda @Edge untuk memodifikasi header.

Pola ini menjelaskan integrasi yang diperlukan dan menyediakan kode contoh untuk modifikasi header dengan menggunakan AWS CloudFront dan Lambda @Edge.

Prasyarat dan batasan

Prasyarat

- Aplikasi lokal yang menggunakan penyeimbang beban F5 dengan konfigurasi yang menggantikan nilai header HTTP dengan menggunakan. `if, else` Untuk informasi selengkapnya tentang konfigurasi ini, lihat HTTP: :header dalam dokumentasi produk F5.

Batasan

- Pola ini berlaku untuk kustomisasi header load balancer F5. Untuk penyeimbang beban pihak ketiga lainnya, silakan periksa dokumentasi penyeimbang beban untuk informasi dukungan.
- Fungsi Lambda yang Anda gunakan untuk Lambda @Edge harus berada di Wilayah AS Timur (Virginia N.).

Arsitektur

Diagram berikut menunjukkan arsitektur di AWS, termasuk aliran integrasi antara CDN dan komponen AWS lainnya.

Alat

Layanan AWS

- Application Load Balancer – Application Load Balancer adalah layanan load balancing AWS yang dikelola sepenuhnya yang berfungsi pada lapisan ketujuh model Open Systems Interconnection (OSI). Ini menyeimbangkan lalu lintas di beberapa target dan mendukung permintaan perutean lanjutan berdasarkan header dan metode HTTP, string kueri, dan perutean berbasis host atau berbasis jalur.
- Amazon CloudFront - Amazon CloudFront adalah layanan web yang mempercepat distribusi konten web statis dan dinamis Anda, seperti.html, .css, .js, dan file gambar, kepada pengguna Anda. CloudFront mengirimkan konten Anda melalui jaringan pusat data di seluruh dunia yang disebut lokasi tepi untuk latensi yang lebih rendah dan peningkatan kinerja.
- Lambda @Edge – Lambda @Edge adalah ekstensi AWS Lambda yang memungkinkan Anda menjalankan fungsi untuk menyesuaikan konten yang dikirimkan. CloudFront Anda dapat membuat fungsi di Wilayah AS Timur (Virginia N.), dan kemudian mengaitkan fungsi tersebut dengan CloudFront distribusi untuk secara otomatis mereplikasi kode Anda di seluruh dunia, tanpa menyediakan atau mengelola server. Ini mengurangi latensi dan meningkatkan pengalaman pengguna.

Kode

Kode contoh berikut menyediakan cetak biru untuk memodifikasi CloudFront header respons. Ikuti instruksi di bagian Epik untuk menyebarkan kode.

```
exports.handler = async (event, context) => {
    const response = event.Records[0].cf.response;
    const headers = response.headers;


    const headerNameSrc = 'content-security-policy';
    const headerNameValue = '*.xyz.com';


    if (headers[headerNameSrc.toLowerCase()]) {
        headers[headerNameSrc.toLowerCase()] = [{
            key: headerNameSrc,
            value: headerNameValue,
        }];
```

```
        console.log(`Response header "${headerNameSrc}" was set to ` +
                    `"${headers[headerNameSrc.toLowerCase()][0].value}"`);
    }
    else {
            headers[headerNameSrc.toLowerCase()] = [{
            key: headerNameSrc,
            value: headerNameValue,
            }];
    }
    return response;
};
```

Epik

Buat distribusi CDN

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat distribusi CloudFront web.	Pada langkah ini, Anda membuat CloudFront distribusi untuk memberi tahu dari CloudFront mana Anda ingin konten dikirim, dan detail tentang cara melacak dan mengelola pengiriman konten. Untuk membuat distribusi menggunakan konsol, masuk ke AWS Management Console, buka CloudFront konsol, lalu ikuti langkah-langkah dalam CloudFront dokumentasi.	Administrator awan

Buat dan terapkan fungsi Lambda @Edge

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan terapkan fungsi Lambda @Edge.	Anda dapat membuat fungsi Lambda @Edge dengan menggunakan cetak biru untuk memodifikasi header respons. CloudFront (Cetak Biru lainnya tersedia untuk kasus penggunaan yang berbeda; untuk informasi selengkapnya, lihat Lambda @Edge contoh fungsi dalam dokumentasi.) CloudFront Untuk membuat fungsi Lambda @Edge: 1. Masuk ke AWS Management Console dan buka konsol AWS Lambda di. https://console.aws.amazon.com/lambda/ 2. Pastikan Anda berada di Wilayah AS Timur (Virginia N.). CloudFront cetak biru hanya tersedia di Wilayah ini. 3. Pilih Buat fungsi. 4. Pilih Gunakan cetak biru, lalu masukkan cloudfront di kolom Pencarian cetak biru.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Pilih cloudfront-modify-response-headercetak biru, lalu pilih Konfigurasi. 6. Pada halaman Informasi dasar, masukkan informasi berikut: a. Masukkan nama fungsi. b. Untuk peran Eksekusi, pilih Buat peran baru dari templat kebijakan AWS. c. Kaitkan nama peran AWS Identity and Access Management (IAM) yang diperlukan. 7. Pilih Buat fungsi. 8. Di bagian Desainer halaman, pilih nama fungsi Anda. 9. Di bagian Kode fungsi, ganti kode template dengan kode sampel yang disediakan sebelumnya dalam pola ini, di bagian Kode. 10. Dalam kode contoh, ganti `xyz.com` dengan nama domain Anda. 11. Pilih Simpan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan fungsi Lambda @Edge.	Ikuti petunjuk di langkah 4 dari Tutorial: Membuat fungsi Lambda @Edge sederhana dalam CloudFront dokumentasi Amazon untuk mengonfigurasi CloudFront pemicu dan menyebarkan fungsi.	Administrator AWS

Sumber daya terkait

CloudFront dokumentasi

- Perilaku permintaan dan respons untuk asal kustom
- Bekerja dengan distribusi
- Lambda @Edge contoh fungsi
- Menyesuaikan di tepi dengan Lambda @Edge
- Tutorial: Membuat fungsi Lambda @Edge sederhana

Buat laporan temuan Network Access Analyzer untuk akses internet masuk dalam beberapa Akun AWS

Mike Virgilio, Amazon Web Services

Ringkasan

Akses internet masuk yang tidak disengaja ke AWS sumber daya dapat menimbulkan risiko bagi perimeter data organisasi. Network Access Analyzer adalah fitur Amazon Virtual Private Cloud (Amazon VPC) yang membantu Anda mengidentifikasi akses jaringan yang tidak diinginkan ke sumber daya Anda di Amazon Web Services ().AWS Anda dapat menggunakan Network Access Analyzer untuk menentukan persyaratan akses jaringan Anda dan untuk mengidentifikasi jalur jaringan potensial yang tidak memenuhi persyaratan yang Anda tentukan. Anda dapat menggunakan Network Access Analyzer untuk melakukan hal berikut:

1. Identifikasi AWS sumber daya yang dapat diakses oleh internet melalui gateway internet.
2. Validasi bahwa virtual private cloud (VPCs) Anda tersegmentasi dengan tepat, seperti mengisolasi lingkungan produksi dan pengembangan dan memisahkan beban kerja transaksional.

Network Access Analyzer menganalisis kondisi jangkauan end-to-end jaringan dan bukan hanya satu komponen. Untuk menentukan apakah sumber daya dapat diakses internet, Network Access Analyzer mengevaluasi gateway internet, tabel rute VPC, daftar kontrol akses jaringan (ACLs), alamat IP publik pada antarmuka jaringan elastis, dan grup keamanan. Jika salah satu komponen ini mencegah akses internet, Network Access Analyzer tidak menghasilkan temuan. Misalnya, jika instans Amazon Elastic Compute Cloud (Amazon EC2) memiliki grup keamanan terbuka yang memungkinkan lalu lintas dari `0/0` tetapi instance berada dalam subnet pribadi yang tidak dapat dirutekan dari gateway internet mana pun, maka Network Access Analyzer tidak akan menghasilkan temuan. Ini memberikan hasil kesetiaan tinggi sehingga Anda dapat mengidentifikasi sumber daya yang benar-benar dapat diakses dari internet.

Ketika Anda menjalankan Network Access Analyzer, Anda menggunakan Network Access Scopes untuk menentukan persyaratan akses jaringan Anda. Solusi ini mengidentifikasi jalur jaringan antara gateway internet dan elastic network interface. Dalam pola ini, Anda menyebarkan solusi secara terpusat Akun AWS di organisasi Anda, dikelola oleh AWS Organizations, dan menganalisis semua akun, di mana pun Wilayah AWS, di organisasi.

Solusi ini dirancang dengan mempertimbangkan hal-hal berikut:

- AWS CloudFormation Template mengurangi upaya yang diperlukan untuk menyebarkan AWS sumber daya dalam pola ini.
- Anda dapat menyesuaikan parameter dalam CloudFormation template dan skrip naa-script.sh pada saat penerapan untuk menyesuaikannya untuk lingkungan Anda.
- Bash scripting secara otomatis menyediakan dan menganalisis Network Access Scopes untuk beberapa akun, secara paralel.
- Skrip Python memproses temuan, mengekstrak data, dan kemudian mengkonsolidasikan hasilnya. Anda dapat memilih untuk meninjau laporan konsolidasi temuan Network Access Analyzer dalam format CSV atau dalam format. AWS Security Hub CSPM Contoh laporan CSV tersedia di bagian Informasi tambahan dari pola ini.
- Anda dapat memulihkan temuan, atau Anda dapat mengecualikannya dari analisis future dengan menambahkannya ke file naa-exclusions.csv.

Prasyarat dan batasan

Prasyarat

- Sebuah Akun AWS untuk layanan dan alat keamanan hosting, dikelola sebagai akun anggota organisasi di AWS Organizations. Dalam pola ini, akun ini disebut sebagai akun keamanan.
- Di akun keamanan, Anda harus memiliki subnet pribadi dengan akses internet keluar. Untuk petunjuknya, lihat Membuat subnet dalam dokumentasi Amazon VPC. Anda dapat membuat akses internet dengan menggunakan gateway NAT atau titik akhir VPC antarmuka.
- Akses ke akun AWS Organizations manajemen atau akun yang telah mendelegasikan izin administrator untuk. CloudFormation Untuk petunjuk, lihat Mendaftarkan administrator yang didelegasikan dalam CloudFormation dokumentasi.
- Aktifkan akses tepercaya antara AWS Organizations dan CloudFormation. Untuk petunjuk, lihat Mengaktifkan akses tepercaya dengan AWS Organizations dalam CloudFormation dokumentasi.
- Jika Anda mengunggah temuan ke Security Hub CSPM, CSPM Security Hub harus diaktifkan di akun dan tempat instans Amazon Wilayah AWS disediakan. EC2 Untuk informasi selengkapnya, lihat Menyiapkan AWS Security Hub CSPM.

Batasan

- Jalur jaringan lintas akun saat ini tidak dianalisis karena keterbatasan fitur Network Access Analyzer.

- Target Akun AWS harus dikelola sebagai organisasi di AWS Organizations. Jika Anda tidak menggunakan AWS Organizations, Anda dapat memperbarui CloudFormation template naa-execrole.yaml dan skrip naa-script.sh untuk lingkungan Anda. Sebagai gantinya, Anda memberikan daftar Akun AWS IDs dan Wilayah tempat Anda ingin menjalankan skrip.
- CloudFormation Template ini dirancang untuk menyebarkan EC2 instance Amazon di subnet pribadi yang memiliki akses internet keluar. AWS Systems Manager Agen (Agen SSM) memerlukan akses keluar untuk mencapai titik akhir layanan Systems Manager, dan Anda memerlukan akses keluar untuk mengkloning repositori kode dan menginstal dependensi. Jika Anda ingin menggunakan subnet publik, Anda harus memodifikasi template naa-resources.yaml untuk mengaitkan alamat IP Elastis dengan instance Amazon. EC2

Arsitektur

Arsitektur target

Opsi 1: Akses temuan di ember Amazon S3

Diagram menunjukkan proses berikut:

1. Jika Anda menjalankan solusi secara manual, pengguna mengautentikasi ke EC2 instance Amazon dengan menggunakan Session Manager dan kemudian menjalankan skrip naa-script.sh. Skrip shell ini melakukan langkah 2—7.

   Jika Anda menjalankan solusi secara otomatis, skrip naa-script.sh dimulai secara otomatis pada jadwal yang Anda tentukan dalam ekspresi cron. Skrip shell ini melakukan langkah 2—7. Untuk informasi selengkapnya, lihat Otomatisasi dan skala di akhir bagian ini.
2. EC2 Instans Amazon mengunduh file naa-exception.csv terbaru dari bucket Amazon S3. File ini digunakan nanti dalam proses ketika skrip Python memproses pengecualian.
3. EC2 Instans Amazon mengasumsikan peran `NAAEC2Role` AWS Identity and Access Management (IAM), yang memberikan izin untuk mengakses bucket Amazon S3 dan untuk mengambil peran `NAAExecRole` IAM di akun lain di organisasi.
4. EC2 Instans Amazon mengasumsikan peran `NAAExecRole` IAM dalam akun manajemen organisasi dan menghasilkan daftar akun dalam organisasi.
5. EC2 Instans Amazon mengasumsikan peran `NAAExecRole` IAM dalam akun anggota organisasi (disebut akun beban kerja dalam diagram arsitektur) dan melakukan penilaian keamanan di setiap akun. Temuan disimpan sebagai file JSON di EC2 instance Amazon.

6. EC2 Instans Amazon menggunakan skrip Python untuk memproses file JSON, mengekstrak bidang data, dan membuat laporan CSV.
7. EC2 Instans Amazon mengunggah file CSV ke bucket Amazon S3.
8. EventBridge Aturan Amazon mendeteksi unggahan file dan menggunakan topik Amazon SNS untuk mengirim email yang memberi tahu pengguna bahwa laporan sudah lengkap.
9. Pengguna mengunduh file CSV dari bucket Amazon S3. Pengguna mengimpor hasil ke template Excel dan meninjau hasilnya.

Opsi 2: Akses temuan di AWS Security Hub CSPM

Diagram menunjukkan proses berikut:

1. Jika Anda menjalankan solusi secara manual, pengguna mengautentikasi ke EC2 instance Amazon dengan menggunakan Session Manager dan kemudian menjalankan skrip naa-script.sh. Skrip shell ini melakukan langkah 2—7.

   Jika Anda menjalankan solusi secara otomatis, skrip naa-script.sh dimulai secara otomatis pada jadwal yang Anda tentukan dalam ekspresi cron. Skrip shell ini melakukan langkah 2—7. Untuk informasi selengkapnya, lihat Otomatisasi dan skala di akhir bagian ini.
2. EC2 Instans Amazon mengunduh file naa-exception.csv terbaru dari bucket Amazon S3. File ini digunakan nanti dalam proses ketika skrip Python memproses pengecualian.
3. EC2 Instans Amazon mengasumsikan peran `NAAEC2Role` IAM, yang memberikan izin untuk mengakses bucket Amazon S3 dan untuk mengambil peran `NAAExecRole` IAM di akun lain di organisasi.
4. EC2 Instans Amazon mengasumsikan peran `NAAExecRole` IAM dalam akun manajemen organisasi dan menghasilkan daftar akun dalam organisasi.
5. EC2 Instans Amazon mengasumsikan peran `NAAExecRole` IAM dalam akun anggota organisasi (disebut akun beban kerja dalam diagram arsitektur) dan melakukan penilaian keamanan di setiap akun. Temuan disimpan sebagai file JSON di EC2 instance Amazon.
6. EC2 Instans Amazon menggunakan skrip Python untuk memproses file JSON dan mengekstrak bidang data untuk diimpor ke Security Hub CSPM.
7. EC2 Instans Amazon mengimpor temuan Network Access Analyzer ke Security Hub CSPM.
8. EventBridge Aturan Amazon mendeteksi impor dan menggunakan topik Amazon SNS untuk mengirim email yang memberi tahu pengguna bahwa prosesnya selesai.

9. Pengguna melihat temuan di Security Hub CSPM.

Otomatisasi dan skala

Anda dapat menjadwalkan solusi ini untuk menjalankan skrip naa-script.sh secara otomatis pada jadwal khusus. Untuk mengatur jadwal kustom, dalam template CloudFormation naa-resources.yaml, ubah parameter. `CronScheduleExpression` Misalnya, nilai default `0 0 * * 0` menjalankan solusi pada tengah malam setiap hari Minggu. Nilai `0 0 * 1-12 0` akan menjalankan solusi pada tengah malam pada hari Minggu pertama setiap bulan. Untuk informasi selengkapnya tentang penggunaan ekspresi cron, lihat Cron dan ekspresi tingkat dalam dokumentasi Systems Manager.

Jika Anda ingin sesuaikan jadwal setelah `NAA-Resources` tumpukan digunakan, Anda dapat mengedit jadwal cron secara manual. `/etc/cron.d/naa-schedule`

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Security Hub CSPMmemberikan pandangan komprehensif tentang keadaan keamanan Anda di AWS. Ini juga membantu Anda memeriksa AWS lingkungan Anda terhadap standar industri keamanan dan praktik terbaik.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar. Pola ini menggunakan Session Manager, kemampuan Systems Manager.

Repositori kode

Kode untuk pola ini tersedia di repositori Analisis Multi-Akun GitHub Network Access Analyzer. Repositori kode berisi file-file berikut:

- naa-script.sh - Skrip bash ini digunakan untuk memulai analisis Network Access Analyzer dari beberapa Akun AWS, secara paralel. Seperti yang didefinisikan dalam CloudFormation template naa-resources.yaml, skrip ini secara otomatis diterapkan ke folder pada instance Amazon. `/usr/local/naa` EC2
- naa-resources.yaml - Anda menggunakan CloudFormation template ini untuk membuat tumpukan di akun keamanan dalam organisasi. Template ini menyebarkan semua sumber daya yang diperlukan untuk akun ini untuk mendukung solusinya. Tumpukan ini harus digunakan sebelum template naa-execrole.yaml.

> **Note**
>
> Jika tumpukan ini dihapus dan digunakan kembali, Anda harus membangun kembali kumpulan `NAAExecRole` tumpukan untuk membangun kembali dependensi lintas akun di antara peran IAM.

- naa-execrole.yaml - Anda menggunakan CloudFormation templat ini untuk membuat kumpulan tumpukan yang menerapkan peran `NAAExecRole` IAM di semua akun di organisasi, termasuk akun manajemen.
- naa-processfindings.py - Skrip naa-script.sh secara otomatis memanggil skrip Python ini untuk memproses output JSON Network Access Analyzer, mengecualikan sumber daya yang diketahui baik dalam file naa-exclusions.csv, dan kemudian menghasilkan file CSV dari hasil konsolidasi atau mengimpor hasilnya ke Security Hub CSPM.

Epik

Mempersiapkan penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode.	1. Di antarmuka baris perintah, ubah direktori kerja Anda ke lokasi tempat Anda ingin menyimpan file sampel. 2. Masukkan perintah berikut: `git clone https://github.com/aws-samples/network-access-analyzer-multi-account-analysis.git`	AWS DevOps
Tinjau template.	1. Di repositori kloning, buka file naa-resources.yaml dan naa-execrole.yaml. 2. Tinjau sumber daya yang dibuat oleh templat ini dan sesuaikan templat sesuai kebutuhan untuk lingkungan Anda. Untuk informasi selengkapnya, lihat Bekerja dengan templat dalam CloudFormation dokumentasi. 3. Simpan dan tutup file naa-resources.yaml dan naa-execrole.yaml.	AWS DevOps

Buat CloudFormation tumpukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan sumber daya di akun keamanan.	Menggunakan template naa-resources.yaml, Anda membuat CloudFormation tumpukan yang menyebarkan semua sumber daya yang diperlukan di akun keamanan. Untuk petunjuk, lihat Membuat tumpukan dalam CloudFormation dokumentasi. Perhatikan hal berikut saat menerapkan template ini: 1. Pada halaman Tentukan template, pilih Template siap, lalu unggah file naa-resources.yaml. 2. Pada halaman Tentukan detail tumpukan, di kotak Nama tumpukan, masukkan`NAA-Resources`. 3. Di bagian Parameter, masukkan yang berikut ini: • `VPCId`— Pilih VPC di akun. • `SubnetId`— Pilih subnet pribadi yang memiliki akses internet. **Note** Jika Anda memilih subnet	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	publik, EC2 instans Amazon mungkin tidak diberi alamat IP publik karena CloudFormation templat, secara default, tidak menyediakan dan melampirkan alamat IP Elastis. • `InstanceType` — Tinggalkan tipe instance default. • `InstanceImageId` — Tinggalkan default. • `KeyPairName` — Jika Anda menggunakan SSH untuk akses, tentukan nama key pair yang ada. • `PermittedSSHInbound` — Jika Anda menggunakan SSH untuk akses, tentukan blok CIDR yang diizinkan. Jika Anda tidak menggunakan SSH, pertahankan nilai default. `127.0.0.1` • `BucketName` — Nilai defaultnya adalah`naa-<accountID>-<region>` . Anda dapat memodifikasi ini	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sesuai kebutuhan. Jika Anda menentukan nilai kustom, ID akun dan Wilayah secara otomatis ditambahkan ke nilai yang ditentukan. • `EmailAddress` — Tentukan alamat email untuk pemberitahuan Amazon SNS saat analisis selesai. **Note** Konfigurasi langganan Amazon SNS harus dikonfirm asi sebelum menyelesaikan analisis, atau pemberitahuan tidak akan dikirim. • `NAAEC2Role` — Pertahankan default kecuali konvensi penamaan Anda memerlukan nama yang berbeda untuk peran IAM ini. • `NAAExecRole` — Pertahankan default kecuali nama lain akan digunakan saat	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menerapkan naa-execr ole.yaml • `Parallelism` — Tentukan jumlah penilaian paralel yang akan dilakukan. • `Regions`— Tentukan yang ingin Wilayah AWS Anda analisis. • `ScopeNameValue` — Tentukan tag yang akan ditetapkan ke ruang lingkup. Tag ini digunakan untuk menentukan Network Access Scope. • `ExclusionFile` — Tentukan nama file pengecualian. Entri dalam file ini akan dikecualikan dari temuan. • `FindingsToCSV` — Tentukan apakah temuan harus output ke CSV. Nilai yang diterima adalah `true` dan`false`. • `FindingsToSecurityHub` — Tentukan apakah temuan harus diimpor ke Security Hub CSPM. Nilai yang diterima adalah `true` dan`false`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `EmailNotifications ForSecurityHub` — Tentukan apakah mengimpor temuan ke Security Hub CSPM harus menghasilkan pemberitahuan email. Nilai yang diterima adalah `true` dan`false`. • `Scheduled Analysis` — Jika Anda ingin solusi berjalan secara otomatis sesuai jadwal, masukkan`true`, lalu sesuaikan jadwal dalam `CronSched uleExpression` parameter. Jika Anda tidak ingin menjalankan solusi secara otomatis, masukkan`false`. • `CronScheduleExpres sion` — Jika Anda menjalankan solusi secara otomatis, masukkan ekspresi cron untuk menentukan jadwal. Untuk informasi selengkapnya, lihat Otomatisasi dan skala di bagian Arsitektur dari pola ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Pada halaman Tinjauan, pilih Sumber daya berikut memerlukan kemampuan: [AWS::IAM::Role], lalu pilih Buat Tumpukan. 2. Setelah tumpukan berhasil dibuat, di CloudFormation konsol, pada tab Output, salin Nama Sumber Daya `NAAEC2Role` Amazon (ARN). Anda menggunakan ARN ini nanti saat menerapkan file naa-execrole.yaml.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan peran IAM di akun anggota.	Di akun AWS Organizat ions manajemen atau akun dengan izin administrator yang didelegasikan CloudForm ation, gunakan templat naa-execrole.yaml untuk membuat kumpulan tumpukan. CloudFormation Kumpulan tumpukan menyebarkan peran `NAAExecRole` IAM di semua akun anggota di organisas i. Untuk petunjuknya, lihat Membuat kumpulan tumpukan dengan izin yang dikelola layanan dalam dokumentasi. CloudFormation Perhatikan hal berikut saat menerapkan template ini: 1. Di bawah Siapkan template, pilih Template siap, lalu unggah file naa-execr ole.yaml. 2. Pada halaman Tentukan StackSet detail, beri nama kumpulan tumpukan`NAA-ExecRole` . 3. Di bagian Parameter, masukkan yang berikut ini: • `AuthorizedARN` — Masukkan `NAAEC2Rol e` ARN, yang Anda salin saat Anda membuat	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tumpukan. `NAA-Resources` • `NAARoleName` — Pertahankan nilai default `NAAExecRole` kecuali nama lain digunakan saat menerapkan file naa-resources.yaml. 4. Dalam Izin, pilih Izin yang dikelola layanan. 5. Pada halaman Setel opsi penerapan, di bawah target Deployment, pilih Deploy to organization dan terima semua default. **Note** Jika Anda ingin tumpukan disebarkan ke semua akun anggota secara bersamaan, setel akun bersamaan maksimum dan toleransi kegagalan ke nilai tinggi, seperti. `100` 6. Di bawah Wilayah Deployment, pilih Wilayah tempat EC2 instans Amazon untuk Network Access Analyzer digunakan. Karena sumber daya IAM	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bersifat global dan bukan Regional, ini menyebark an peran IAM di semua Wilayah aktif. 7. Pada halaman Tinjauan, pilih Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM dengan nama khusus, lalu pilih Buat. StackSet 8. Pantau tab Stack instance (untuk status akun individua l) dan tab Operasi (untuk status keseluruhan) untuk menentukan kapan penerapan selesai.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan peran IAM dalam akun manajemen.	Menggunakan template naa-execrole.yaml, Anda membuat CloudFormation tumpukan yang menerapkan peran `NAAExecRole` IAM di akun manajemen organisasi. Kumpulan tumpukan yang Anda buat sebelumnya tidak menerapkan peran IAM di akun manajemen. Untuk petunjuk, lihat Membuat tumpukan dalam CloudFormation dokumentasi. Perhatikan hal berikut saat menerapkan template ini: 1. Pada halaman Tentukan template, pilih Template siap, lalu unggah file naa-execrole.yaml. 2. Pada halaman Tentukan detail tumpukan, di kotak Nama tumpukan, masukkan`NAA-ExecRole` . 3. Di bagian Parameter, masukkan yang berikut ini: • `AuthorizedARN` — Masukkan `NAAEC2Role` ARN, yang Anda salin saat Anda membuat tumpukan. `NAA-Resources`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `NAARoleName` — Pertahankan nilai default `NAAExecRole` kecuali nama lain digunakan saat menerapkan file naa-resources.yaml. 4. Pada halaman Tinjauan, pilih Sumber daya berikut memerlukan kemampuan: [AWS::IAM::Role], lalu pilih Buat Tumpukan.	

Lakukan analisis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan skrip shell.	1. Masuk ke akun keamanan di organisasi. 2. Menggunakan Session Manager, sambungkan ke EC2 instans Amazon untuk Network Access Analyzer yang sebelumnya Anda sediakan. Untuk petunjuk, lihat Connect ke instance Linux Anda menggunakan Session Manager. Jika Anda tidak dapat terhubung, lihat bagian Pemecahan Masalah dari pola ini. 3. Masukkan perintah berikut untuk membuka file naa-script.sh untuk diedit:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`sudo -i`
	`cd /usr/local/naa`
	`vi naa-script.sh`
	4. Tinjau dan modifikasi parameter dan variabel yang dapat disesuaikan dalam skrip ini sesuai kebutuhan untuk lingkungan Anda. Untuk informasi selengkapnya tentang opsi penyesuaian, lihat komentar di awal skrip.
	Misalnya, alih-alih mendapatkan daftar semua akun anggota di organisasi dari akun manajemen, Anda dapat memodifikasi skrip untuk menentukan Akun AWS IDs atau Wilayah AWS yang ingin Anda pindai, atau Anda dapat mereferensikan file eksternal yang berisi parameter ini.
	5. Simpan dan tutup file naa-script.sh.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Analisis akun target.	1. Masukkan perintah berikut. Ini menjalankan skrip naa-script.sh:
	`sudo -i`
	`cd /usr/local/naa`
	`screen`
	`./naa-script.sh`
	Perhatikan hal-hal berikut:
	• `screen`Perintah memungkinkan skrip untuk terus berjalan jika waktu koneksi habis atau Anda kehilangan akses konsol.
	• Setelah pemindaian dimulai, Anda dapat memaksa layar terlepas dengan menekan Ctrl +A D. Layar terlepas, dan Anda dapat menutup koneksi instance saat analisis berlangsung.
	• Untuk melanjutk an sesi terpisah, sambungkan ke instance, masukkan `sudo -i` lalu masukkan`screen -r`.
	2. Pantau output untuk kesalahan apa pun untuk memastikan bahwa skrip berfungsi dengan baik. Untuk keluaran sampel,
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lihat bagian Informasi tambahan dari pola ini. 3. Tunggu analisis selesai. Jika Anda mengonfigurasi notifikasi email, Anda akan menerima email saat hasilnya telah diunggah ke bucket Amazon S3 atau diimpor ke CSPM Security Hub.	
Opsi 1 - Ambil hasil dari bucket Amazon S3.	1. Unduh file CSV dari `naa-<accountID>-<region>` ember. Untuk petunjuk, lihat Mengunduh objek di dokumentasi Amazon S3. 2. Hapus file CSV dari bucket Amazon S3. Ini adalah praktik terbaik untuk optimasi biaya. Untuk petunjuknya, lihat Menghapus objek dalam dokumentasi Amazon S3.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 2 — Tinjau hasilnya di Security Hub CSPM.	1. Buka konsol CSPM Security Hub. 2. Pilih Temuan dari panel navigasi. 3. Tinjau temuan Network Access Analyzer. Untuk petunjuk, lihat Melihat daftar pencarian dan detail dalam dokumentasi CSPM Security Hub. **Note** Anda dapat mencari temuan dengan menambahkan Judul dimulai dengan filter dan masuk`Network Access Analyzer`.	AWS DevOps

Memperbaiki dan mengecualikan temuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memperbaiki temuan.	Perbaiki temuan apa pun yang ingin Anda atasi. Untuk informasi selengkapnya dan praktik terbaik tentang cara membuat perimeter di sekitar AWS identitas, sumber daya, dan jaringan Anda, lihat	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Membangun perimeter data di AWS (AWS Whitepaper).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kecualikan sumber daya dengan jalur jaringan yang diketahui baik.	Jika Network Access Analyzer menghasilkan temuan untuk sumber daya yang seharusnya dapat diakses dari internet, maka Anda dapat menambahkan sumber daya ini ke daftar pengecualian. Lain kali Network Access Analyzer berjalan, itu tidak akan menghasilkan temuan untuk sumber daya itu. 1. Arahkan ke`/usr/local/naa` , lalu buka skrip naa-script.sh. Catat nilai `S3_EXCLUSION_FILE` variabel. 2. Jika nilai `S3_EXCLUSION_FILE` variabeln `yatrue`, unduh file naa-exclusions.csv dari `naa-<accountID>-<region>` bucket. Untuk petunjuk, lihat Mengunduh objek di dokumentasi Amazon S3. Jika nilai `S3_EXCLUSION_FILE` variabeln `yafalse`, navigasikan ke `/usr/local/naa` dan kemudian buka file naa-exclusions.csv.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Note
	Jika nilai `S3_EXCLUSION_FILE` variabelnya `false`, skrip menggunakan versi lokal dari file pengecualian. Jika nanti Anda mengubah nilainya `true`, skrip akan menimpa versi lokal dengan file di bucket Amazon S3.
	3. Dalam file naa-exclusions.csv, masukkan sumber daya yang ingin Anda kecualikan. Masukkan satu sumber daya di setiap baris, dan gunakan format berikut.
	`<resource_id>,<secgroup_id>,<sgrule_cidr>,<sgrule_portrange>,<sgrule_protocol>`
	Berikut ini adalah sumber daya contoh.
	`eni-1111aaaaa2222bbbb,sg-3333ccccc44`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`44dddd,0.0.0.0/0,80 to 80,tcp` 4. Simpan dan tutup file naa-exclusions.csv. 5. Jika Anda mengunduh file naa-exclusions.csv dari bucket Amazon S3, unggah versi baru. Untuk petunjuk, lihat Mengunggah objek dalam dokumentasi Amazon S3.	

(Opsional) Perbarui skrip naa-script.sh

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui skrip naa-script.sh.	Jika Anda ingin memperbarui skrip naa-script.sh ke versi terbaru di repo, lakukan hal berikut: 1. Connect ke EC2 instans Amazon menggunakan Session Manager. Untuk petunjuk, lihat Connect ke instance Linux Anda menggunakan Session Manager. 2. Masukkan perintah berikut: `sudo -i` 3. Arahkan ke direktori skrip naa-script.sh:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cd /usr/local/naa`
	4. Masukkan perintah berikut untuk menyimpan skrip lokal sehingga Anda dapat menggabungkan perubahan kustom ke versi terbaru:
	`git stash`
	5. Masukkan perintah berikut untuk mendapatkan skrip versi terbaru:
	`git pull`
	6. Masukkan perintah berikut untuk menggabungkan skrip kustom dengan skrip versi terbaru:
	`git stash pop`	

(Opsional) Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus semua sumber daya yang digunakan.	Anda dapat meninggalkan sumber daya yang digunakan di akun. Jika Anda ingin menghentikan semua sumber daya, lakukan hal berikut:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Hapus `NAA-ExecRole` tumpukan yang disediakan di akun manajemen. Untuk petunjuk, lihat Menghapus tumpukan dalam CloudFormation dokumentasi. 2. Hapus kumpulan `NAA-ExecRole` tumpukan yang disediakan di akun manajemen organisasi atau di akun administrator yang didelegasikan. Untuk petunjuk, lihat Menghapus kumpulan tumpukan dalam CloudFormation dokumentasi. 3. Hapus semua objek di bucket `naa-<accountID>-<region>` Amazon S3. Untuk petunjuknya, lihat Menghapus objek dalam dokumentasi Amazon S3. 4. Hapus `NAA-Resources` tumpukan yang disediakan di akun keamanan. Untuk petunjuk, lihat Menghapus tumpukan dalam CloudFormation dokumentasi.	

Pemecahan masalah

Isu	Solusi
Tidak dapat terhubung ke EC2 instans Amazon dengan menggunakan Pengelola Sesi.	Agen SSM harus dapat berkomunikasi dengan endpoint Systems Manager. Lakukan hal-hal berikut: 1. Validasi subnet tempat EC2 instans Amazon digunakan memiliki akses internet. 2. Reboot EC2 instance Amazon.
Saat menerapkan set tumpukan, CloudFormation konsol meminta Anda untuk melakukannya. `Enable trusted access with AWS Organizations to use service-managed permissions`	Ini menunjukkan bahwa akses tepercaya belum diaktifkan antara AWS Organizations dan CloudFormation. Akses tepercaya diperlukan untuk menerapkan set tumpukan yang dikelola layanan. Pilih tombol untuk mengaktifkan akses tepercaya. Untuk informasi selengkapnya, lihat Mengaktifkan akses tepercaya dalam CloudFormation dokumentasi.

Sumber daya terkait

- Baru - Penganalisis Akses Jaringan Amazon VPC (AWS posting blog)
- AWS re:Inforce 2022 - Validasi kontrol akses jaringan yang efektif pada AWS (NIS202) (video)
- Demo - Analisis Jalur Data Ingress Internet di Seluruh Organisasi Menggunakan Penganalisis Akses Jaringan (video)

Informasi tambahan

Contoh keluaran konsol

Contoh berikut menunjukkan output dari menghasilkan daftar akun target dan menganalisis akun target.

```
[root@ip-10-10-43-82 naa]# ./naa-script.sh
download: s3://naa-<account ID>-us-east-1/naa-exclusions.csv to ./naa-exclusions.csv
```

```
AWS Management Account: <Management account ID>

AWS Accounts being processed...
<Account ID 1> <Account ID 2> <Account ID 3>

Assessing AWS Account: <Account ID 1>, using Role: NAAExecRole
Assessing AWS Account: <Account ID 2>, using Role: NAAExecRole
Assessing AWS Account: <Account ID 3>, using Role: NAAExecRole
Processing account: <Account ID 1> / Region: us-east-1
Account: <Account ID 1> / Region: us-east-1 - Detecting Network Analyzer scope...
Processing account: <Account ID 2> / Region: us-east-1
Account: <Account ID 2> / Region: us-east-1 - Detecting Network Analyzer scope...
Processing account: <Account ID 3> / Region: us-east-1
Account: <Account ID 3> / Region: us-east-1 - Detecting Network Analyzer scope...
Account: <Account ID 1> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 1> / Region: us-east-1 - Continuing analyses with Scope ID.
 Accounts with many resources may take up to one hour
Account: <Account ID 2> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 2> / Region: us-east-1 - Continuing analyses with Scope ID.
 Accounts with many resources may take up to one hour
Account: <Account ID 3> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 3> / Region: us-east-1 - Continuing analyses with Scope ID.
 Accounts with many resources may take up to one hour
```

Contoh laporan CSV

Gambar berikut adalah contoh dari output CSV.

Mengatur resolusi DNS untuk jaringan hybrid di lingkungan multi-akun AWS

Anvesh Kokganti, Amazon Web Services

Ringkasan

Pola ini memberikan solusi komprehensif untuk menyiapkan resolusi DNS di lingkungan jaringan hybrid yang mencakup beberapa akun Amazon Web Services (AWS). Ini memungkinkan resolusi DNS dua arah antara jaringan lokal dan lingkungan melalui titik akhir. AWS Amazon Route 53 Resolver Pola ini menyajikan dua solusi untuk mengaktifkan resolusi DNS dalam arsitektur multi-akun dan terpusat:

- Pengaturan dasar tidak menggunakan Profil Route 53. Ini membantu mengoptimalkan biaya untuk penyebaran kecil hingga menengah dengan kompleksitas yang lebih rendah.
- Penyiapan yang disempurnakan menggunakan Route 53 Profiles untuk menyederhanakan operasi. Cara terbaik untuk penyebaran DNS yang lebih besar atau lebih kompleks.

> **Note**
>
> Tinjau bagian Batasan untuk batasan layanan dan kuota sebelum implementasi. Pertimbangkan faktor-faktor seperti overhead manajemen, biaya, kompleksitas operasional, dan keahlian tim ketika Anda membuat keputusan.

Prasyarat dan batasan

Prasyarat

- Lingkungan AWS multi-akun dengan Amazon Virtual Private Cloud (Amazon VPC) diterapkan di seluruh Layanan Bersama dan akun beban kerja (sebaiknya disiapkan melalui AWS Control Tower dengan AWS mengikuti praktik terbaik untuk struktur akun).
- Konektivitas hybrid yang ada (AWS Direct Connect atau AWS Site-to-Site VPN) antara jaringan lokal dan AWS lingkungan.
- Amazon VPC peering, AWS Transit Gateway, atau AWS Cloud WAN untuk konektivitas jaringan Layer 3 antara. VPCs (Konektivitas ini diperlukan untuk lalu lintas aplikasi. Resolusi DNS tidak diperlukan untuk bekerja. Resolusi DNS beroperasi secara independen dari konektivitas jaringan antara VPCs.)

- Server DNS berjalan di lingkungan lokal.

Batasan

- Titik akhir, aturan, dan Profil Resolver Route 53 adalah konstruksi Regional dan mungkin memerlukan replikasi dalam beberapa untuk organisasi global. Wilayah AWS
- Untuk daftar lengkap kuota layanan untuk Resolver Route 53, zona host pribadi, dan Profil, lihat Kuota dalam dokumentasi Route 53.

Arsitektur

Tumpukan teknologi target

- Rute 53 titik akhir keluar dan masuk
- Aturan Route 53 Resolver untuk penerusan bersyarat
- AWS Resource Access Manager (AWS RAM)
- Route 53 zona host pribadi

Arsitektur target

Titik akhir keluar dan masuk

Diagram berikut menunjukkan aliran resolusi DNS dari AWS ke tempat. Ini adalah pengaturan konektivitas untuk resolusi keluar tempat domain di-host di tempat. Berikut adalah ikhtisar tingkat tinggi dari proses yang terlibat dalam pengaturan ini. Untuk detailnya, lihat bagian Epik.

1. Terapkan titik akhir Resolver Route 53 keluar di VPC Layanan Bersama.
2. Buat aturan Resolver Route 53 (aturan penerusan) di akun Layanan Bersama untuk domain yang dihosting di tempat.
3. Bagikan dan kaitkan aturan dengan VPCs akun lain yang menghosting sumber daya yang diperlukan untuk menyelesaikan domain yang dihosting lokal. Ini dapat dilakukan dengan cara yang berbeda tergantung pada kasus penggunaan Anda, seperti yang dijelaskan nanti di bagian ini.

Setelah Anda mengatur konektivitas, langkah-langkah yang terlibat dalam resolusi keluar adalah sebagai berikut:

1. Instans Amazon Elastic Compute Cloud (Amazon EC2) mengirimkan permintaan resolusi DNS `db.onprem.example.com` ke Resolver Route 53 VPC di alamat VPC+2.
2. Route 53 Resolver memeriksa aturan Resolver dan meneruskan permintaan ke server DNS lokal dengan menggunakan titik akhir keluar. IPs
3. Titik akhir keluar meneruskan permintaan ke DNS lokal. IPs Lalu lintas melewati konektivitas jaringan hybrid yang mapan antara VPC Layanan Bersama dan pusat data lokal.
4. Server DNS lokal merespons kembali ke titik akhir keluar, yang kemudian meneruskan respons kembali ke Resolver Route 53 VPC. Resolver mengembalikan respons terhadap instance. EC2

Diagram berikutnya menunjukkan aliran resolusi DNS dari lingkungan lokal ke. AWS Ini adalah pengaturan konektivitas untuk resolusi masuk tempat domain di-host. AWS Berikut adalah ikhtisar tingkat tinggi dari proses yang terlibat dalam pengaturan ini. Untuk detailnya, lihat bagian Epik.

1. Menerapkan titik akhir Resolver masuk di VPC Layanan Bersama.
2. Buat zona host pribadi di akun Layanan Bersama (pendekatan terpusat).
3. Kaitkan zona yang dihosting pribadi dengan VPC Layanan Bersama. Bagikan dan kaitkan zona ini dengan akun silang VPCs untuk resolusi VPC-to-VPC DNS. Ini dapat dilakukan dengan cara yang berbeda tergantung pada kasus penggunaan Anda, seperti yang dijelaskan nanti di bagian ini.

Setelah Anda mengatur konektivitas, langkah-langkah yang terlibat dalam resolusi masuk adalah sebagai berikut:

1. Sumber daya lokal mengirimkan permintaan resolusi DNS `ec2.prod.aws.example.com` ke server DNS lokal.
2. Server DNS lokal meneruskan permintaan ke titik akhir Resolver masuk di VPC Layanan Bersama melalui koneksi jaringan hybrid.
3. Titik akhir Resolver masuk mencari permintaan di zona host pribadi terkait dengan bantuan VPC Route 53 Resolver dan mendapatkan alamat IP yang sesuai.
4. Alamat IP ini dikirim kembali ke server DNS lokal, yang mengembalikan respons ke sumber daya lokal.

Konfigurasi ini memungkinkan sumber daya lokal untuk menyelesaikan nama domain AWS pribadi dengan merutekan kueri melalui titik akhir masuk ke zona host pribadi yang sesuai. Dalam arsitektur ini, zona host pribadi dipusatkan dalam VPC Layanan Bersama, yang memungkinkan manajemen DNS pusat oleh satu tim. Zona ini dapat dikaitkan dengan banyak orang VPCs untuk mengatasi kasus penggunaan resolusi VPC-to-VPC DNS. Atau, Anda mungkin ingin mendelegasikan kepemilikan dan pengelolaan domain DNS ke masing-masing domain. Akun AWS Dalam hal ini, setiap akun mengelola zona host pribadinya sendiri dan mengaitkan setiap zona dengan VPC Layanan Bersama pusat untuk resolusi terpadu dengan lingkungan lokal. Pendekatan terdesentralisasi ini berada di luar cakupan pola ini. Untuk informasi selengkapnya, lihat Menskalakan manajemen DNS di beberapa akun dan VPCs di whitepaper opsi DNS Cloud Hybrid untuk Amazon VPC.

Saat Anda menetapkan alur resolusi DNS dasar dengan menggunakan titik akhir Resolver, Anda perlu menentukan cara mengelola pembagian dan asosiasi aturan Resolver dan zona host pribadi di seluruh area Anda. Akun AWS Anda dapat melakukan pendekatan ini dengan dua cara: melalui berbagi yang dikelola sendiri dengan menggunakan AWS RAM untuk berbagi aturan Resolver dan mengarahkan asosiasi zona host pribadi, seperti yang dirinci di bagian Pengaturan dasar, atau melalui Profil Route 53, seperti yang dibahas di bagian Penyiapan yang ditingkatkan. Pilihannya tergantung pada preferensi manajemen DNS organisasi Anda dan persyaratan operasional. Diagram arsitektur berikut menggambarkan lingkungan berskala yang mencakup beberapa akun yang berbeda, VPCs yang mewakili penerapan perusahaan yang khas.

Pengaturan dasar

Dalam penyiapan dasar, implementasi untuk resolusi DNS hibrid di AWS lingkungan multi-akun digunakan AWS RAM untuk berbagi aturan penerusan Resolver dan asosiasi zona host pribadi untuk mengelola kueri DNS antara lokal dan sumber daya. AWS Metode ini menggunakan titik akhir Resolver Route 53 terpusat dalam VPC Layanan Bersama yang terhubung ke jaringan lokal Anda untuk menangani resolusi DNS masuk dan keluar secara efisien.

- Untuk resolusi keluar, aturan penerusan Resolver dibuat di akun Layanan Bersama dan kemudian dibagikan dengan orang lain dengan menggunakan. Akun AWS AWS RAM Berbagi ini terbatas pada akun dalam Wilayah yang sama. Akun target kemudian dapat mengaitkan aturan ini dengan aturan mereka VPCs dan memungkinkan sumber daya di akun tersebut VPCs untuk menyelesaikan nama domain lokal.
- Untuk resolusi masuk, zona yang dihosting pribadi dibuat di akun Layanan Bersama dan dikaitkan dengan VPC Layanan Bersama. Zona ini kemudian dapat dikaitkan dengan VPCs akun lain dengan menggunakan API Route 53 AWS SDKs,, atau AWS Command Line Interface (AWS CLI).

Sumber daya yang terkait kemudian VPCs dapat menyelesaikan catatan DNS yang ditentukan di zona host pribadi, yang menciptakan tampilan DNS terpadu di seluruh lingkungan Anda. AWS

Diagram berikut menunjukkan aliran resolusi DNS dalam pengaturan dasar ini.

Pengaturan ini berfungsi dengan baik ketika Anda bekerja dengan infrastruktur DNS dalam skala terbatas. Namun, itu bisa menjadi tantangan untuk dikelola saat lingkungan Anda tumbuh. Overhead operasional untuk mengelola bagaimana zona host pribadi dan aturan Resolver dibagikan dan dikaitkan dengan VPCs individu meningkat secara signifikan dengan skala. Selain itu, kuota layanan seperti batas asosiasi VPC 300 per zona host pribadi dapat menjadi faktor pembatas dalam penerapan skala besar. Penyiapan yang disempurnakan mengatasi tantangan ini.

Pengaturan yang disempurnakan

Route 53 Profiles menawarkan solusi efisien untuk mengelola resolusi DNS di jaringan hybrid di beberapa jaringan. Akun AWS Alih-alih mengelola zona yang dihosting pribadi dan aturan Resolver secara individual, Anda dapat mengelompokkan konfigurasi DNS ke dalam satu wadah yang dapat dengan mudah dibagikan dan diterapkan di beberapa akun VPCs dan akun di Wilayah. Pengaturan ini mempertahankan arsitektur titik akhir Resolver terpusat dalam VPC Layanan Bersama sambil secara signifikan menyederhanakan pengelolaan konfigurasi DNS.

Diagram berikut menunjukkan aliran resolusi DNS dalam pengaturan yang disempurnakan.

Profil Route 53 memungkinkan Anda mengemas asosiasi zona host pribadi, aturan penerusan Resolver, dan aturan firewall DNS ke dalam satu unit yang dapat dibagikan. Anda dapat membuat Profil di akun Layanan Bersama dan membagikannya dengan akun anggota dengan menggunakan AWS RAM. Ketika profil dibagikan dan diterapkan ke target VPCs, semua asosiasi dan konfigurasi yang diperlukan ditangani secara otomatis oleh layanan. Ini secara signifikan mengurangi overhead operasional manajemen DNS dan memberikan skalabilitas yang sangat baik untuk lingkungan yang berkembang.

Otomatisasi dan skala

Gunakan alat infrastruktur sebagai kode (IAc) seperti CloudFormation atau Terraform untuk secara otomatis menyediakan dan mengelola titik akhir Route 53 Resolver, aturan, zona yang dihosting pribadi, dan Profil. Integrasikan konfigurasi DNS dengan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) untuk konsistensi, pengulangan, dan pembaruan cepat.

Alat

Layanan AWS

- AWS Resource Access Manager (AWS RAM) membantu Anda berbagi sumber daya dengan aman Akun AWS untuk mengurangi overhead operasional dan memberikan visibilitas dan auditabilitas.
- Amazon Route 53 Resolvermerespons secara rekursif kueri DNS dari AWS sumber daya dan tersedia secara default di semua. VPCs Anda dapat membuat titik akhir Resolver dan aturan penerusan bersyarat untuk menyelesaikan ruang nama DNS antara pusat data lokal dan pusat data lokal Anda. VPCs
- Amazon Route 53 private hosted zone adalah wadah yang menyimpan informasi tentang bagaimana Anda ingin Route 53 merespons kueri DNS untuk domain dan subdomainnya.
- Profil Amazon Route 53 memungkinkan Anda menerapkan dan mengelola konfigurasi Route 53 terkait DNS di banyak VPCs dan berbeda dengan Akun AWS cara yang disederhanakan.

Praktik terbaik

Bagian ini menyediakan beberapa praktik terbaik untuk mengoptimalkan Resolver Route 53. Ini mewakili bagian dari praktik terbaik Route 53. Untuk daftar lengkap, lihat Praktik terbaik untuk Amazon Route 53.

Hindari konfigurasi loop dengan titik akhir Resolver

- Rancang arsitektur DNS Anda untuk mencegah perutean rekursif dengan merencanakan asosiasi VPC dengan cermat. Saat VPC menghosting titik akhir masuk, hindari mengaitkannya dengan aturan Resolver yang dapat membuat referensi melingkar.
- Gunakan AWS RAM secara strategis saat Anda berbagi sumber daya DNS di seluruh akun untuk mempertahankan jalur perutean yang bersih.

Untuk informasi selengkapnya, lihat Menghindari konfigurasi loop dengan titik akhir Resolver dalam dokumentasi Route 53.

Titik akhir Resolver Skala

- Untuk lingkungan yang membutuhkan jumlah query per detik (QPS) yang tinggi, ketahuilah bahwa ada batas 10.000 QPS per ENI di titik akhir. Lebih banyak ENIs dapat ditambahkan ke titik akhir untuk menskalakan DNS QPS.

- Amazon CloudWatch menyediakan `InboundQueryVolume` dan `OutboundQueryVolume` metrik (lihat CloudWatch dokumentasi). Kami menyarankan Anda mengatur aturan pemantauan yang mengingatkan Anda jika ambang batas melebihi nilai tertentu (misalnya, 80 persen dari 10.000 QPS).
- Konfigurasikan aturan grup keamanan stateful untuk titik akhir Resolver untuk mencegah batas pelacakan koneksi menyebabkan pembatasan kueri DNS selama lalu lintas volume tinggi. Untuk mempelajari selengkapnya tentang cara kerja pelacakan koneksi di grup keamanan, lihat Pelacakan sambungan grup EC2 keamanan Amazon di EC2 dokumentasi Amazon.

Untuk informasi selengkapnya, lihat Penskalaan titik akhir Resolver dalam dokumentasi Route 53.

Menyediakan ketersediaan tinggi untuk titik akhir Resolver

- Buat titik akhir masuk dengan alamat IP di setidaknya dua Availability Zone untuk redundansi.
- Menyediakan antarmuka jaringan tambahan untuk memastikan ketersediaan selama pemeliharaan atau lonjakan lalu lintas.

Untuk informasi selengkapnya, lihat Ketersediaan tinggi untuk titik akhir Resolver dalam dokumentasi Route 53.

Epik

Menyebarkan titik akhir Resolver Route 53

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan titik akhir masuk.	Route 53 Resolver menggunakan titik akhir masuk untuk menerima kueri DNS dari resolver DNS lokal. Untuk petunjuk, lihat Meneruskan kueri DNS masuk ke dokumentasi Route 53 Anda VPCs . Catat alamat IP titik akhir masuk.	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan titik akhir keluar.	Route 53 Resolver menggunakan titik akhir keluar untuk mengirim kueri DNS ke resolver DNS lokal. Untuk petunjuk, lihat Meneruskan kueri DNS keluar ke jaringan Anda di dokumentasi Route 53. Catat ID titik akhir keluaran.	Administrator AWS, Administrator cloud

Konfigurasikan dan bagikan Route 53 zona yang dihosting pribadi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat zona host pribadi untuk domain yang di-host AWS.	Zona ini menyimpan catatan DNS untuk sumber daya dalam domain AWS-host (misalnya,`prod.aws.example.com`) yang harus diselesaikan dari lingkungan lokal. Untuk petunjuk, lihat Membuat zona host pribadi dalam dokumentasi Route 53. Saat membuat zona host pribadi, Anda harus mengaitkan VPC dengan zona host yang dimiliki oleh akun yang sama. Pilih VPC Layanan Bersama untuk tujuan ini.	Administrator AWS, Administrator cloud
Pengaturan dasar: Kaitkan zona yang dihosting pribadi VPCs dengan akun lain.	Jika Anda menggunakan pengaturan dasar (lihat bagian Arsitektur):	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk mengaktifkan sumber daya di akun anggota VPCs untuk menyelesaikan catatan DNS di zona host pribadi ini, Anda harus mengaitkannya VPCs dengan zona yang dihosting. Anda harus mengotorisasi asosiasi dan kemudian membuat asosiasi secara terprogram. Untuk petunjuknya, lihat Mengaitkan VPC Amazon dan zona host pribadi yang Anda buat dengan Akun AWS berbeda dalam dokumentasi Route 53.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Penyiapan yang disempurn akan: Konfigurasikan dan bagikan Profil Route 53.	Jika Anda menggunakan penyiapan yang disempurn akan (lihat bagian Arsitektur): 1. Buat Profil Route 53 dan kaitkan zona host pribadi yang relevan dengannya . Untuk petunjuknya, lihat Membuat Profil Route 53 dalam dokumentasi Route 53. 2. Gunakan AWS RAM untuk berbagi profil dengan akun anggota dan kemudian mengaitkan profil bersama dengan target VPCs. Untuk petunjuk, lihat Berbagi Profil Route 53 dan Mengaitkan Profil Route 53 ke VPCs dalam dokumentasi Route 53. **Note** Bergantung pada struktur organisasi dan persyaratan DNS, Anda mungkin perlu membuat dan mengelola beberapa Profil untuk akun atau beban kerja yang berbeda.	Administrator AWS, Administrator cloud

Konfigurasikan dan bagikan aturan penerusan Resolver Route 53

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan penerusan untuk domain yang di-host di tempat.	Aturan ini akan menginstruksikan Resolver Route 53 untuk meneruskan kueri DNS apa pun untuk domain lokal (seperti) ke resolver DNS lokal. `onprem.example.com` Untuk membuat aturan ini, Anda memerlukan alamat IP resolver DNS lokal dan ID titik akhir keluar. Untuk petunjuknya, lihat Membuat aturan penerusan dalam dokumentasi Route 53.	Administrator AWS, Administrator cloud
Pengaturan dasar: Bagikan dan kaitkan aturan penerusan dengan akun Anda VPCs di akun lain.	Jika Anda menggunakan pengaturan dasar: Agar aturan penerusan berlaku, Anda harus membagikan dan mengaitkan aturan tersebut dengan akun Anda VPCs di akun lain. Route 53 Resolver kemudian mempertimbangkan aturan saat menyelesaikan domain. Untuk petunjuknya, lihat Berbagi aturan Resolver dengan aturan lain Akun AWS dan menggunakan aturan bersama serta Mengaitkan aturan penerusan dengan	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	VPC dalam dokumentasi Route 53.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Penyiapan yang disempurn akan: Konfigurasikan dan bagikan Profil Route 53.	Jika Anda menggunakan pengaturan yang disempurn akan: 1. Jika Anda telah membuat Profil Route 53 di langkah sebelumnya, Anda dapat menggunakan Profil yang sama. Jika tidak, buat Profil Route 53 dan kaitkan aturan penerusan Resolver yang relevan dengannya . Untuk petunjuknya, lihat Membuat Profil Route 53 dalam dokumentasi Route 53. 2. Gunakan AWS RAM untuk berbagi profil dengan akun anggota, lalu kaitkan profil bersama dengan target VPCs. Untuk petunjuk, lihat Berbagi Profil Route 53 dan Mengaitkan Profil Route 53 ke VPCs dalam dokumenta si Route 53. **Note** Bergantung pada struktur organisas i dan persyaratan DNS, Anda mungkin perlu membuat dan mengelola beberapa	Administrator AWS, Administr ator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Profil untuk akun atau beban kerja yang berbeda.	

Konfigurasikan resolver DNS lokal untuk integrasi AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan penerusan bersyarat di resolver DNS lokal.	Agar kueri DNS dikirim AWS dari lingkungan lokal untuk resolusi, Anda harus mengonfigurasi penerusan bersyarat di resolver DNS lokal untuk menunjuk ke alamat IP titik akhir masuk. Ini menginstr uksikan resolver DNS untuk meneruskan semua kueri DNS untuk domain AWS-host (misalnya, untuk`prod.aws.example.com`) ke alamat IP titik akhir masuk untuk resolusi oleh Route 53 Resolver.	Administrator jaringan

Verifikasi resolusi end-to-end DNS di lingkungan hybrid

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji resolusi DNS dari AWS ke lingkungan lokal.	Dari instance di VPC yang memiliki aturan penerusan yang terkait dengannya, lakukan kueri DNS untuk	Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	domain host lokal (misalnya , untuk). `db.onprem.example.com`	
Uji resolusi DNS dari lingkungan lokal ke. AWS	Dari server lokal, lakukan resolusi DNS untuk domain AWS-host (misalnya, untuk). `ec2.prod.aws.example.com`	Administrator jaringan

Sumber daya terkait

- Opsi DNS Cloud Hybrid untuk Amazon VPCAWS (whitepaper)
- Bekerja dengan zona host pribadi (dokumentasi Route 53)
- Memulai dengan Resolver Route 53 (Dokumentasi Route 53)
- Sederhanakan manajemen DNS di lingkungan multi-akun dengan Route 53 Resolver (posting blog)AWS
- Menyatukan manajemen DNS menggunakan Profil Amazon Route 53 dengan beberapa VPCs dan Akun AWS (AWS posting blog)
- Memigrasi lingkungan DNS multi-akun Anda ke Profil Amazon Route 53 (posting blog)AWS
- Menggunakan Profil Amazon Route 53 untuk AWS lingkungan multi-akun yang dapat diskalakan (AWS posting blog)

Verifikasi bahwa penyeimbang beban ELB memerlukan penghentian TLS

Priyanka Chaudhary, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, Elastic Load Balancing (ELB) secara otomatis mendistribusikan lalu lintas aplikasi yang masuk ke beberapa target, seperti instans Amazon Elastic Compute Cloud (Amazon EC2), container, alamat IP, dan fungsi AWS Lambda. Load balancer menggunakan pendengar untuk menentukan port dan protokol yang digunakan penyeimbang beban untuk menerima lalu lintas dari pengguna. Application Load Balancers membuat keputusan routing di lapisan aplikasi dan menggunakan protokol. HTTP/HTTPS Classic Load Balancer membuat keputusan routing baik pada layer transport, dengan menggunakan protokol TCP atau Secure Sockets Layer (SSL), atau pada layer aplikasi, dengan menggunakan HTTP/HTTPS.

Pola ini menyediakan kontrol keamanan yang memeriksa beberapa jenis peristiwa untuk Application Load Balancers dan Classic Load Balancers. Saat fungsi dipanggil, AWS Lambda memeriksa peristiwa dan memastikan bahwa penyeimbang beban sesuai.

Fungsi ini memulai CloudWatch peristiwa Amazon Events pada panggilan API berikut: CreateLoadBalancer, CreateLoadBalancerListeners, DeleteLoadBalancerListeners, CreateLoadBalancerPolicy, SetLoadBalancerPoliciesOfListener, CreateListenerDeleteListener, dan ModifyListener. Ketika acara mendeteksi salah satunya APIs, ia memanggil AWS Lambda, yang menjalankan skrip Python. Skrip Python mengevaluasi untuk melihat apakah listener berisi sertifikat SSL, dan apakah kebijakan yang diterapkan menggunakan Transport Layer Security (TLS). Jika kebijakan SSL ditentukan sebagai apa pun selain TLS, fungsi akan mengirimkan notifikasi Amazon Simple Notification Service (Amazon SNS) kepada pengguna dengan informasi yang relevan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif

Batasan

- Kontrol keamanan ini tidak memeriksa penyeimbang beban yang ada, kecuali pembaruan dilakukan untuk pendengar penyeimbang beban.

- Kontrol keamanan ini bersifat regional. Anda harus menerapkannya di setiap Wilayah AWS yang ingin Anda pantau.

Arsitektur

Arsitektur target

Otomatisasi dan skala

- Jika Anda menggunakan AWS Organizations, Anda dapat menggunakan AWS Cloudformation StackSets untuk menerapkan template ini di beberapa akun yang ingin Anda pantau.

Alat

Layanan AWS

- AWS CloudFormation — AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual.
- Amazon CloudWatch Events — Amazon CloudWatch Events menghadirkan aliran peristiwa sistem yang mendekati real-time yang menjelaskan perubahan sumber daya AWS.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel yang dapat digunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan danau data.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) mengoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien, termasuk server web dan alamat email. Pelanggan menerima semua pesan yang dipublikasikan ke topik tempat mereka berlangganan, dan semua pelanggan topik menerima pesan yang sama.

Kode

Pola ini mencakup lampiran berikut:

- `ELBRequirestlstermination.zip`— Kode Lambda untuk kontrol keamanan.
- `ELBRequirestlstermination.yml`— CloudFormation Template yang mengatur acara dan fungsi Lambda.

Epik

Siapkan ember S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan bucket S3.	Di konsol Amazon S3, pilih atau buat bucket S3 untuk meng-host file kode Lambda .zip. Bucket S3 ini harus berada di Wilayah AWS yang sama dengan penyeimbang beban yang ingin Anda evaluasi. Nama bucket S3 unik secara global, dan namespace dibagikan oleh semua akun AWS. Nama bucket S3 tidak dapat menyertakan garis miring terdepan.	Arsitek awan
Unggah kode Lambda.	Unggah kode Lambda (`ELBRequirestlstermination.zip` file) yang disediakan di bagian Lampiran ke bucket S3.	Arsitek awan

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan CloudFormation template AWS.	Buka CloudFormation konsol AWS di Wilayah AWS yang sama dengan bucket S3 Anda dan terapkan template terlampir. `ELBRequirestlstermination.yml` Untuk informasi selengkapnya tentang penerapan CloudFormation templat AWS, lihat Membuat tumpukan di CloudFormation konsol AWS dalam CloudFormation dokumentasi.	Arsitek awan
Lengkapi parameter dalam template.	Ketika Anda meluncurkan template, Anda akan diminta untuk informasi berikut: • Bucket S3: Tentukan bucket yang Anda buat atau pilih di epik pertama. Di sinilah Anda mengunggah kode Lambda terlampir `ELBRequirestlstermination.zip` (file). • Kunci S3: Tentukan lokasi file.zip Lambda di bucket S3 Anda (misalnya, atau). `ELBRequirestlstermination.zip` `controls/ELBRequirestlstermination.z`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ip` Jangan sertakan garis miring terdepan. • Email pemberitahuan: Berikan alamat email aktif tempat Anda ingin menerima notifikasi Amazon SNS. • Level logging Lambda: Tentukan level logging dan frekuensi untuk fungsi Lambda. Gunakan Info untuk mencatat pesan informasi terperinci tentang kemajuan, Kesalahan untuk peristiwa kesalahan yang masih memungkinkan penerapan dilanjutkan, dan Peringatan untuk situasi yang berpotensi berbahaya.	

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Ketika CloudFormation template berhasil digunakan , ia mengirimkan email berlangganan ke alamat email yang Anda berikan. Anda harus mengonfirmasi langganan email ini untuk mulai menerima pemberita huan pelanggaran.	Arsitek awan

Sumber daya terkait

- Membuat tumpukan di CloudFormation konsol AWS (CloudFormation dokumentasi AWS)
- Apa itu AWS Lambda? (Dokumentasi AWS Lambda)
- Apa itu Classic Load Balancer? (Dokumentasi ELB)
- Apa itu Penyeimbang Beban Aplikasi? (Dokumentasi ELB)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Lihat log dan metrik AWS Network Firewall dengan menggunakan Splunk

Ivo Pinto, Amazon Web Services

Ringkasan

Banyak organisasi menggunakan Splunk Enterprise sebagai alat agregasi dan visualisasi terpusat untuk log dan metrik dari berbagai sumber. Pola ini membantu Anda mengonfigurasi Splunk untuk mengambil log dan metrik AWS Network Firewall dari Amazon CloudWatch Logs dengan menggunakan Splunk Add-On untuk AWS.

Untuk mencapai hal ini, Anda membuat peran AWS Identity and Access Management (IAM) read-only. Splunk Add-On untuk AWS menggunakan peran ini untuk mengakses. CloudWatch Anda mengonfigurasi Add-On Splunk untuk AWS untuk mengambil metrik dan log. CloudWatch Terakhir, Anda membuat visualisasi di Splunk dari data log dan metrik yang diambil.

Prasyarat dan batasan

Prasyarat

- Akun Splunk
- Instans Splunk Enterprise, versi 8.2.2 atau yang lebih baru
- Akun AWS yang aktif
- Network Firewall, atur dan konfigurasi untuk mengirim log ke CloudWatch Log

Batasan

- Splunk Enterprise harus digunakan sebagai klaster instans Amazon Elastic Compute Cloud (Amazon EC2) di AWS Cloud.
- Mengumpulkan data dengan menggunakan peran IAM yang ditemukan secara otomatis untuk Amazon EC2 tidak didukung di Wilayah AWS China.

Arsitektur

Diagram ini menggambarkan hal sebagai berikut:

1. Network Firewall menerbitkan log ke CloudWatch Log.

2. Splunk Enterprise mengambil metrik dan log dari. CloudWatch

Untuk mengisi contoh metrik dan log dalam arsitektur ini, beban kerja menghasilkan lalu lintas yang melewati titik akhir Network Firewall untuk masuk ke internet. Ini dicapai dengan menggunakan tabel rute. Meskipun pola ini menggunakan satu instans Amazon EC2 sebagai beban kerja, pola ini dapat diterapkan ke arsitektur apa pun selama Network Firewall dikonfigurasi untuk mengirim log ke Log. CloudWatch

Arsitektur ini juga menggunakan instance Splunk Enterprise di virtual private cloud (VPC) lainnya. Namun, instance Splunk dapat berada di lokasi lain, seperti di VPC yang sama dengan beban kerja, selama dapat mencapai. CloudWatch APIs

Alat

Layanan AWS

- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, dan layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Network Firewall adalah firewall jaringan yang stateful, dikelola, dan layanan deteksi dan pencegahan intrusi untuk AWS Cloud VPCs .

Alat lainnya

- Splunk membantu Anda memantau, memvisualisasikan, dan menganalisis data log.

Epik

Membuat peran IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM.	Ikuti petunjuk dalam Membuat kebijakan menggunakan editor JSON untuk	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membuat kebijakan IAM yang memberikan akses hanya-baca ke data dan metrik Log. CloudWatch CloudWatch Tempelkan kebijakan berikut ke editor JSON. `{ "Statement": [ { "Action": [ "cloudwatch:List*", "cloudwatch:Get*", "network-firewall: List*", "logs:Describe*", "logs:Get*", "logs:List*", "logs:StartQuery", "logs:StopQuery", "logs:TestMetricFi lter", "logs:FilterLogEve nts", "network-firewall: Describe*" ], "Effect": "Allow",`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Resource": "*"
	}
	],
	"Version": "2012-10-17"
	}</pre>	

Buat peran IAM baru.	Ikuti petunjuk dalam Membuat peran untuk mendelegasikan izin ke layanan AWS guna membuat peran IAM yang digunakan oleh Add-On Splunk untuk AWS. CloudWatch Untuk kebijakan Izin, pilih kebijakan yang Anda buat sebelumnya.	Administrator AWS
Tetapkan peran IAM ke instans EC2 di cluster Splunk.	1. Buka konsol Amazon EC2 di. https://console.aws.amazon.com/ec2/ 2. Di panel navigasi, pilih Instans. 3. Pilih instans EC2 di cluster Splunk. 4. Pilih Tindakan, Keamanan, dan kemudian Ubah peran IAM. 5. Pilih peran IAM yang Anda buat sebelumnya, lalu pilih Simpan.	Administrator AWS

Instal Add-On Splunk untuk AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal add-on.	1. Di dasbor Splunk, arahkan ke Aplikasi Splunk. 2. Cari Add-on Splunk untuk Amazon Web Services. 3. Pilih Instal. 4. Berikan kredensi Splunk Anda.	Administrator splunk
Konfigurasikan kredenal AWS.	1. Di dasbor Splunk, navigasikan ke Splunk Add-on untuk AWS. 2. Pilih Konfigurasi. 3. Di kolom Peran IAM yang Ditemukan Otomatis, pilih peran IAM yang Anda buat sebelumnya. Untuk informasi selengkapnya, lihat Menemukan peran IAM dalam instans platform Splunk Anda dalam dokumentasi Splunk.	Administrator splunk

Konfigurasikan akses Splunk ke CloudWatch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengambilan log Network Firewall dari CloudWatch Log.	1. Di dasbor Splunk, navigasikan ke Splunk Add-on untuk AWS. 2. Pilih Input.	Administrator splunk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pilih Buat Input Baru. 4. Dalam daftar, pilih Jenis Data Kustom, lalu pilih CloudWatch Log. 5. Berikan Nama, Akun AWS, Wilayah AWS, dan Grup Log untuk log Firewall Jaringan Anda. 6. Pilih Simpan. Secara default, Splunk mengambil data log setiap 10 menit. Ini adalah parameter yang dapat dikonfigurasi di bawah Pengaturan Lanjutan. Untuk informasi selengkap nya, lihat Mengkonfigurasi input CloudWatch Log menggunakan Splunk Web dalam dokumentasi Splunk.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengambilan metrik Network Firewall dari. CloudWatch	1. Di dasbor Splunk, navigasikan ke Splunk Add-on untuk AWS. 2. Pilih Input. 3. Pilih Buat Input Baru. 4. Dalam daftar, pilih CloudWatch. 5. Berikan Nama, Akun AWS, dan Wilayah AWS untuk metrik Network Firewall Anda. 6. Di samping Konfigurasi Metrik, pilih Edit dalam mode lanjutan. 7. (Opsional) Hapus semua ruang nama yang telah dikonfigurasi sebelumnya. 8. Pilih Tambahkan Namespace, lalu beri nama AWS/. NetworkFirewall 9. Dalam Nilai Dimensi, tambahkan yang berikut ini. `[{"AvailabilityZone":[".*"],"Engine":[".*"],"FirewallName":[".*"]}]` 10. Untuk Metrik, pilih Semua. 11. Untuk Statistik Metrik, pilih Jumlah. 12. Pilih OK. 13. Pilih Simpan.	Administrator splunk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Secara default, Splunk mengambil data metrik setiap 5 menit. Ini adalah parameter yang dapat dikonfigurasi di bawah Pengaturan Lanjutan. Untuk informasi selengkap nya, lihat Mengkonfigurasi CloudWatch input menggunak an Splunk Web dalam dokumentasi Splunk.	

Buat visualisasi Splunk dengan menggunakan kueri

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat alamat IP sumber teratas.	1. Di dasbor Splunk, navigasik an ke Pencarian & Pelaporan. 2. Di kotak enter search here, masukkan yang berikut ini. `sourcetype="aws:cl oudwatchlogs" | top event.src_ip` Kueri ini menampilkan tabel alamat IP sumber dengan lalu lintas terbanyak, dalam urutan menurun. 3. Untuk representasi grafis, pilih Visualisasi.	Administrator splunk
Lihat statistik paket.	1. Di dasbor Splunk, navigasik an ke Pencarian & Pelaporan.	Administrator splunk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di kotak enter search here, masukkan yang berikut ini. `sourcetype="aws:cloudwatch"| timechart sum(Sum) by metric_name` Kueri ini menampilkan tabel metrik `DroppedPackets` ,`PassedPackets` , dan `ReceivedPackets` per menit. 3. Untuk representasi grafis, pilih Visualisasi.	
Lihat port sumber yang paling banyak digunakan.	1. Di dasbor Splunk, navigasikan ke Pencarian & Pelaporan. 2. Di kotak enter search here, masukkan yang berikut ini. `sourcetype="aws:cloudwatchlogs" | top event.dest_port` Kueri ini menampilkan tabel port sumber dengan lalu lintas terbanyak, dalam urutan menurun. 3. Untuk representasi grafis, pilih Visualisasi.	Administrator splunk

Sumber daya terkait

Dokumentasi AWS

- Membuat peran untuk mendelegasikan izin ke layanan AWS (dokumentasi IAM)
- Membuat kebijakan IAM (dokumentasi IAM)
- Pencatatan dan pemantauan di AWS Network Firewall (dokumentasi Network Firewall)
- Konfigurasi tabel rute untuk AWS Network Firewall (dokumentasi Network Firewall)

Posting blog AWS

- Model penerapan AWS Network Firewall

AWS Marketplace

- Gambar Mesin Amazon Splunk Enterprise (AMI)

Lebih banyak pola

- Akses host bastion menggunakan Session Manager dan Amazon EC2 Instance Connect
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS Fargate, PrivateLink AWS, dan Network Load Balancer
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS PrivateLink dan Network Load Balancer
- Memusatkan resolusi DNS dengan menggunakan Microsoft Active AWS Managed Microsoft AD Directory dan lokal
- Buat portal untuk frontend mikro dengan menggunakan AWS Amplify, Angular, dan Federasi Modul
- Menerapkan API Amazon API Gateway di situs web internal menggunakan endpoint pribadi dan Application Load Balancer
- Menyebarkan kontrol akses berbasis atribut detektif untuk subnet publik dengan menggunakan AWS Config
- Menyebarkan kontrol akses berbasis atribut preventif untuk subnet publik
- Aktifkan koneksi terenkripsi untuk instans PostgreSQL DB di Amazon RDS
- Memperluas VRFs ke AWS dengan menggunakan AWS Transit Gateway Connect
- Migrasikan beban kerja F5 BIG-IP ke F5 BIG-IP VE di AWS Cloud
- Migrasikan Pengontrol Ingress NGINX saat mengaktifkan Mode Otomatis Amazon EKS
- Pertahankan ruang IP yang dapat dirutekan dalam desain VPC multi-akun untuk subnet non-beban kerja
- Mencegah akses internet di tingkat akun dengan menggunakan kebijakan kontrol layanan
- Kirim peringatan dari AWS Network Firewall ke saluran Slack
- Sajikan konten statis dalam bucket Amazon S3 melalui VPC dengan menggunakan Amazon CloudFront
- Siapkan pemulihan bencana untuk Oracle JD Edwards dengan EnterpriseOne AWS Elastic Disaster Recovery
- Gunakan kueri BMC Discovery untuk mengekstrak data migrasi untuk perencanaan migrasi
- Gunakan Network Firewall untuk menangkap nama domain DNS dari Indikasi Nama Server untuk lalu lintas keluar

Pengiriman konten

Topik

- Kirim AWS WAF log ke Splunk dengan menggunakan AWS Firewall Manager dan Amazon Data Firehose
- Sajikan konten statis dalam bucket Amazon S3 melalui VPC dengan menggunakan Amazon CloudFront
- Lebih banyak pola

Kirim AWS WAF log ke Splunk dengan menggunakan AWS Firewall Manager dan Amazon Data Firehose

Michael Friedenthal, Aman Kaur Gandhi, dan JJ Johnson, Amazon Web Services

Ringkasan

Secara historis, ada dua cara untuk memindahkan data ke Splunk: arsitektur push atau pull. Arsitektur pull menawarkan jaminan data pengiriman melalui percobaan ulang, tetapi membutuhkan sumber daya khusus di Splunk data jajak pendapat tersebut. Arsitektur pull biasanya tidak real time karena polling. Arsitektur push biasanya memiliki latensi yang lebih rendah, lebih terukur, dan mengurangi kompleksitas dan biaya operasional. Namun, itu tidak menjamin pengiriman dan biasanya membutuhkan agen.

Integrasi splunk dengan Amazon Data Firehose memberikan data streaming real-time ke Splunk melalui HTTP event collector (HEC). Integrasi ini memberikan keuntungan dari arsitektur push dan pull — ini menjamin pengiriman data melalui percobaan ulang, mendekati waktu nyata, dan latensi rendah dan kompleksitas rendah. HEC dengan cepat dan efisien mengirimkan data melalui HTTP atau HTTPS langsung ke Splunk. HECs berbasis token, yang menghilangkan kebutuhan untuk hardcode kredensil dalam aplikasi atau dalam file pendukung.

Dalam AWS Firewall Manager kebijakan, Anda dapat mengonfigurasi pencatatan untuk semua lalu lintas ACL AWS WAF web di semua akun, dan kemudian Anda dapat menggunakan aliran pengiriman Firehose untuk mengirim data log tersebut ke Splunk untuk pemantauan, visualisasi, dan analisis. Solusi ini memberikan manfaat sebagai berikut:

- Manajemen pusat dan pencatatan untuk lalu lintas ACL AWS WAF web di semua akun Anda
- Integrasi splunk dengan satu Akun AWS
- Skalabilitas
- Mendekati pengiriman data log secara real-time
- Optimalisasi biaya melalui penggunaan solusi tanpa server, sehingga Anda tidak perlu membayar sumber daya yang tidak digunakan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS yang merupakan bagian dari organisasi di AWS Organizations.
- Anda harus memiliki izin berikut untuk mengaktifkan logging dengan Firehose:
  - `iam:CreateServiceLinkedRole`
  - `firehose:ListDeliveryStreams`
  - `wafv2:PutLoggingConfiguration`
- AWS WAF dan webnya ACLs harus dikonfigurasi. Untuk petunjuk, lihat Memulai dengan AWS WAF.
- AWS Firewall Manager harus setup. Untuk instruksi, lihat AWS Firewall Manager prasyarat.
- Kebijakan keamanan Firewall Manager untuk AWS WAF harus dikonfigurasi. Untuk petunjuk, lihat Memulai AWS Firewall ManagerAWS WAF kebijakan.
- Splunk harus diatur dengan titik akhir HTTP publik yang dapat dijangkau oleh Firehose.

Batasan

- Akun AWS Harus dikelola dalam satu organisasi di AWS Organizations.
- ACL web harus berada di Wilayah yang sama dengan aliran pengiriman. Jika Anda menangkap log untuk Amazon CloudFront, buat aliran pengiriman Firehose di Wilayah AS Timur (Virginia N.),. `us-east-1`
- Add-on Splunk untuk Firehose tersedia untuk penerapan Splunk Cloud berbayar, penerapan Splunk Enterprise terdistribusi, dan penerapan Splunk Enterprise satu instance. Add-on ini tidak didukung untuk penerapan Splunk Cloud uji coba gratis.

Arsitektur

Tumpukan teknologi target

- Firewall Manager
- Firehouse
- Amazon Simple Storage Service (Amazon S3)
- AWS WAF
- Splunk

Arsitektur target

Gambar berikut menunjukkan bagaimana Anda dapat menggunakan Firewall Manager untuk mencatat semua AWS WAF data secara terpusat dan mengirimkannya ke Splunk melalui Firehose.

1. AWS WAF Web ACLs mengirim data log firewall ke Firewall Manager.
2. Firewall Manager mengirimkan data log ke Firehose.
3. Aliran pengiriman Firehose meneruskan data log ke Splunk dan ke bucket S3. Bucket S3 bertindak sebagai cadangan jika terjadi kesalahan dengan aliran pengiriman Firehose.

Otomatisasi dan skala

Solusi ini dirancang untuk skala dan mengakomodasi semua AWS WAF web ALCs dalam organisasi. Anda dapat mengonfigurasi semua web ACLs untuk menggunakan instance Firehose yang sama. Namun, jika Anda ingin mengatur dan menggunakan beberapa instance Firehose, Anda bisa.

Alat

Layanan AWS

- AWS Firewall Manageradalah layanan manajemen keamanan yang membantu Anda mengonfigurasi dan mengelola aturan firewall secara terpusat di seluruh akun dan aplikasi Anda. AWS Organizations
- Amazon Data Firehose membantu Anda mengirimkan data streaming real-time ke titik akhir HTTP kustom lainnya Layanan AWS, dan titik akhir HTTP yang dimiliki oleh penyedia layanan pihak ketiga yang didukung, seperti Splunk.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS WAFadalah firewall aplikasi web yang membantu Anda memantau permintaan HTTP dan HTTPS yang diteruskan ke sumber daya aplikasi web Anda yang dilindungi.

Alat-alat lainnya

- Splunk membantu Anda memantau, memvisualisasikan, dan menganalisis data log.

Epik

Konfigurasikan Splunk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Aplikasi Splunk untuk AWS.	1. Masuk ke forwarder berat Splunk Anda. URL defaultnya adalah`http://<IP address>:8000` . 2. Di navigasi kiri, di sebelah Aplikasi, pilih tombol roda gigi. 3. Pilih Jelajahi lebih banyak aplikasi. 4. Cari AWS. 5. Untuk Aplikasi Splunk AWS, pilih Instal. 6. Masukkan kredensi masuk Splunk.com Anda, setujui syarat dan ketentuan, lalu pilih Login dan Instal. 7. Pilih Selesai.	Administrator keamanan, administrator Splunk
Instal add-on untuk AWS WAF.	Ulangi instruksi sebelumny a untuk menginstal Add-on Firewall Aplikasi AWS Web untuk Splunk.	Administrator keamanan, administrator Splunk
Instal dan konfigurasikan add-on Splunk untuk Firehose.	1. Instal dan konfigura sikan add-on Splunk untuk Firehose. Sebagai bagian dari instalasi dan konfigurasi, jika perlu untuk platform Splunk Anda, Anda menyiapkan HTTP Event	Administrator keamanan, administrator Splunk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Collector dan menyiapkan infrastruktur untuk mengirim data log ke pengindeks Anda. Lihat petunjuk yang sesuai dengan penerapan Splunk Anda: • Penyebaran Splunk Cloud (Dokumentasi Splunk) • Penyebaran Perusahaan Splunk Terdistribusi (Dokumentasi Splunk) • Penerapan Splunk Enterprise satu instance (dokumentasi Splunk) **Important** Hentikan prosedur ini setelah Anda menginstal dan mengkonfigurasi add-on Splunk. Jangan lanjutkan dengan instruksi untuk mengkonfigurasi Firehose untuk mengirim data ke platform Splunk. 2. Catat token kolektor acara HTTP dan titik akhir HTTP. Anda memerlukan nilai ini	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nanti, saat Anda mengonfig urasi aliran pengiriman.	

Buat aliran pengiriman Firehose

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan akses Firehose ke tujuan Splunk.	Konfigurasikan kebijakan akses yang mengizinkan Firehose mengakses tujuan Splunk dan mencadangkan data log ke bucket S3. Untuk informasi selengkapnya, lihat Grant Firehose akses ke tujuan Splunk.	Administrator keamanan
Buat aliran pengiriman Firehose.	Di akun yang sama tempat Anda mengelola web AWS WAF, ACLs buat aliran pengiriman di Firehose. Anda harus memiliki IAM role saat membuat aliran pengirima n. Firehose mengasumsikan bahwa IAM berperan dan mendapatkan akses ke bucket S3 yang ditentukan. Untuk petunjuknya, lihat Membuat aliran pengiriman. Perhatikan hal-hal berikut: • Nama aliran pengiriman harus dimulai dengan `aws-waf-logs-` . • Untuk sumbernya, pilih Direct PUT.	Administrator keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk mode cadangan S3, pilih Backup semua acara, lalu pilih bucket yang sudah ada atau buat yang baru. • Untuk tujuan, ikuti petunjuk di Pilih Splunk untuk tujuan Anda di dokumentasi Firehose. Untuk informasi tentang nilai untuk titik akhir dan tipe titik akhir Splunk, lihat Mengonfig urasi Amazon Data Firehose dalam dokumentasi Splunk. Ulangi proses ini untuk setiap token yang Anda konfigura sikan di kolektor acara HTTP.	
Uji aliran pengiriman.	Uji aliran pengiriman untuk memvalidasi bahwa itu dikonfigurasi dengan benar. Untuk petunjuk, lihat Menguji menggunakan Splunk sebagai tujuan dalam dokumentasi Firehose.	Administrator keamanan

Konfigurasikan Firewall Manager untuk mencatat data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kebijakan Firewall Manager.	Kebijakan Firewall Manager harus dikonfigurasi untuk mengaktifkan logging dan meneruskan log ke aliran	Administrator keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengiriman Firehose yang benar. Untuk informasi dan petunjuk selengkapnya, lihat Mengonfigurasi pencatatan untuk AWS WAF kebijakan.	

Sumber daya terkait

AWS sumber daya

- Logging lalu lintas ACL web (AWS WAF dokumentasi)
- Mengkonfigurasi pencatatan untuk AWS WAF kebijakan (AWS WAF dokumentasi)
- Tutorial: Mengirim Log Aliran VPC ke Splunk Menggunakan Amazon Data Firehose (dokumentasi Firehose)
- Bagaimana cara mendorong log aliran VPC ke Splunk menggunakan Amazon Data Firehose? (Pusat AWS Pengetahuan)
- Daya konsumsi data ke Splunk menggunakan Amazon Data Firehose (posting blog)AWS

Dokumentasi splunk

- Pengaya Splunk untuk Amazon Data Firehose

Sajikan konten statis dalam bucket Amazon S3 melalui VPC dengan menggunakan Amazon CloudFront

Malaikat Emmanuel Hernandez Cebrian, Amazon Web Services

Ringkasan

Saat Anda menayangkan konten statis yang di-host di Amazon Web Services (AWS), pendekatan yang disarankan adalah menggunakan bucket Amazon Simple Storage Service (S3) Amazon Simple Storage Service (S3) sebagai asal dan menggunakan Amazon CloudFront untuk mendistribusikan konten. Solusi ini memiliki dua manfaat utama: kenyamanan caching konten statis di lokasi tepi, dan kemampuan untuk menentukan daftar kontrol akses web (web ACLs) untuk CloudFront distribusi, yang membantu Anda mengamankan permintaan ke konten dengan konfigurasi minimal dan overhead administratif.

Namun, ada batasan arsitektur umum untuk standar, pendekatan yang direkomendasikan. Di beberapa lingkungan, Anda ingin peralatan firewall virtual digunakan di cloud pribadi virtual (VPC) untuk memeriksa semua konten, termasuk konten statis. Pendekatan standar tidak merutekan lalu lintas melalui VPC untuk diperiksa. Pola ini memberikan solusi arsitektur alternatif. Anda masih menggunakan CloudFront distribusi untuk menyajikan konten statis dalam bucket S3, tetapi lalu lintas dirutekan melalui VPC dengan menggunakan Application Load Balancer. Fungsi AWS Lambda kemudian mengambil dan mengembalikan konten dari bucket S3.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Konten situs web statis yang dihosting di bucket S3.

Batasan

- Sumber daya dalam pola ini harus dalam satu Wilayah AWS, tetapi dapat disediakan di akun AWS yang berbeda.
- Batas berlaku untuk ukuran permintaan dan respons maksimum yang dapat diterima dan dikirim oleh fungsi Lambda. Untuk informasi selengkapnya, lihat Batas dalam fungsi Lambda sebagai target (dokumentasi Elastic Load Balancing).

- Penting untuk menemukan keseimbangan yang baik antara kinerja, skalabilitas, keamanan, dan efektivitas biaya saat menggunakan pendekatan ini. Meskipun skalabilitas Lambda tinggi, jika jumlah pemanggilan Lambda bersamaan melebihi kuota maksimum, beberapa permintaan dibatasi. Untuk informasi selengkapnya, lihat Kuota Lambda (dokumentasi Lambda). Anda juga perlu mempertimbangkan harga saat menggunakan Lambda. Untuk meminimalkan pemanggilan Lambda, pastikan Anda menentukan cache untuk distribusi dengan benar. CloudFront Untuk informasi selengkapnya, lihat Mengoptimalkan caching dan ketersediaan (CloudFront dokumentasi).

Arsitektur

Tumpukan teknologi target

- CloudFront
- Amazon Virtual Private Cloud (Amazon VPC)
- Penyeimbang Beban Aplikasi
- Lambda
- Amazon S3

Arsitektur target

Gambar berikut menunjukkan arsitektur yang disarankan saat Anda perlu menggunakan CloudFront untuk menyajikan konten statis dari bucket S3 melalui VPC.

1. Klien meminta URL CloudFront distribusi untuk mendapatkan file situs web tertentu di bucket S3.
2. CloudFront mengirimkan permintaan ke AWS WAF. AWS WAF memfilter permintaan dengan menggunakan web yang ACLs diterapkan pada distribusi. CloudFront Jika permintaan ditentukan valid, aliran berlanjut. Jika permintaan ditentukan tidak valid, klien menerima kesalahan 403.
3. CloudFront memeriksa cache internalnya. Jika ada kunci yang valid yang cocok dengan permintaan yang masuk, nilai terkait dikirim kembali ke klien sebagai respons. Jika tidak, aliran terus berlanjut.
4. CloudFront meneruskan permintaan ke URL Application Load Balancer yang ditentukan.
5. Application Load Balancer memiliki pendengar yang terkait dengan grup target berdasarkan fungsi Lambda. Application Load Balancer memanggil fungsi Lambda.

6. Fungsi Lambda terhubung ke bucket S3, melakukan `GetObject` operasi di atasnya, dan mengembalikan konten sebagai respons.

Otomatisasi dan skala

Untuk mengotomatiskan penyebaran konten statis menggunakan pendekatan ini, buat CI/CD pipeline untuk memperbarui bucket Amazon S3 yang meng-host situs web.

Fungsi Lambda menskalakan secara otomatis untuk menangani permintaan bersamaan, dalam kuota dan batasan layanan. Untuk informasi selengkapnya, lihat Penskalaan fungsi Lambda dan kuota Lambda (dokumentasi Lambda). Untuk layanan dan fitur AWS lainnya, seperti CloudFront dan Application Load Balancer, AWS menskalakan ini secara otomatis.

Alat

- Amazon CloudFront mempercepat distribusi konten web Anda dengan mengirimkannya melalui jaringan pusat data di seluruh dunia, yang menurunkan latensi dan meningkatkan kinerja.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Dalam pola ini, Anda menggunakan Application Load Balancer yang disediakan melalui Elastic Load Balancing untuk mengarahkan lalu lintas ke fungsi Lambda.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Epik

Gunakan CloudFront untuk menyajikan konten statis dari Amazon S3 melalui VPC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Buat VPC untuk hosting sumber daya yang digunakan dalam pola ini, seperti Application Load Balancer dan fungsi Lambda. Untuk petunjuk, lihat Membuat VPC (dokumentasi Amazon VPC).	Arsitek awan
Buat AWS WAF web ACL.	Buat AWS WAF web ACL. Kemudian dalam pola ini, Anda menerapkan ACL web ini ke CloudFront distribusi. Untuk petunjuk, lihat Membuat ACL web (dokumentasi AWS WAF).	Arsitek awan
Buat fungsi Lambda.	Buat fungsi Lambda yang menyajikan konten statis yang dihosting di bucket S3 sebagai situs web. Gunakan kode yang disediakan di bagian Informasi tambahan dari pola ini. Sesuaikan kode untuk mengidentifikasi bucket S3 target Anda.	AWS Umum
Unggah fungsi Lambda.	Masukkan perintah berikut untuk mengunggah kode fungsi Lambda ke arsip file.zip di Lambda.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws lambda update-function-code \ --function-name \ --zip-file fileb://lambda-alb-s3-website.zip`	
Buat Application Load Balancer.	Buat Application Load Balancer yang menghadap ke internet yang menunjuk ke fungsi Lambda. Untuk petunjuk, lihat Membuat grup target untuk fungsi Lambda (dokumentasi Elastic Load Balancing). Untuk konfigurasi ketersediaan tinggi, buat Application Load Balancer dan pasang ke subnet pribadi di Availability Zone yang berbeda.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFront distribusi.	Buat CloudFront distribusi yang menunjuk ke Application Load Balancer yang Anda buat. 1. Masuk ke AWS Management Console dan buka CloudFront konsol di https://console.aws.amazon.com/cloudfront/v3/home. 2. Pilih Buat Distribusi. 3. Di halaman pertama Buat Wizard Distribusi, di Web bagian, pilih Memulai. 4. Tentukan pengaturan untuk distribusi Anda. Untuk informasi selengkapnya, lihat Nilai yang Anda tentukan saat membuat atau memperbarui distribusi. Perhatikan hal-hal berikut: a. Atur Application Load Balancer sebagai asal. b. Dalam pengaturan Distribusi, pilih web ACLs yang ada yang ingin Anda terapkan melalui AWS WAF. Untuk informasi selengkapnya, lihat AWS WAF web ACL. 5. Simpan perubahan Anda.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Setelah CloudFront membuat distribusi Anda, nilai kolom Status untuk distribusi Anda berubah dari InProgressDeployed. Jika Anda memilih untuk mengaktifkan distribus i, maka akan siap untuk memproses permintaan setelah status beralih ke Diterapkan.	

Sumber daya terkait

Dokumentasi AWS

- Mengoptimalkan caching dan ketersediaan (dokumentasi) CloudFront
- Lambda berfungsi sebagai target (dokumentasi Elastic Load Balancing)
- Kuota Lambda (dokumentasi Lambda)

Situs web layanan AWS

- Penyeimbang Beban Aplikasi
- Lambda
- CloudFront
- Amazon S3
- AWS WAF
- Amazon VPC

Informasi tambahan

Kode

Berikut contoh fungsi Lambda ditulis dalam Node.js. Fungsi Lambda ini bertindak sebagai server web yang melakukan `GetObject` operasi ke bucket S3 yang berisi sumber daya situs web.

```
/**

 * This is an AWS Lambda function created for demonstration purposes.

 * It retrieves static assets from a defined Amazon S3 bucket.

 * To make the content available through a URL, use an Application Load Balancer with a
 Lambda integration.
 *
 * Set the S3_BUCKET environment variable in the Lambda function definition.
 */

var AWS = require('aws-sdk');

exports.handler = function(event, context, callback) {

    var bucket = process.env.S3_BUCKET;
    var key = event.path.replace('/', '');

    if (key == '') {
        key = 'index.html';
    }

    // Fetch from S3
    var s3 = new AWS.S3();
    return s3.getObject({Bucket: bucket, Key: key},
       function(err, data) {

            if (err) {
                return err;
            }

            var isBase64Encoded = false;
            var encoding = 'utf8';

            if (data.ContentType.indexOf('image/') > -1) {
                isBase64Encoded = true;
                encoding = 'base64'
            }
```

```
            var resp = {
                statusCode: 200,
                headers: {
                    'Content-Type': data.ContentType,
                },
                body: new Buffer(data.Body).toString(encoding),
                isBase64Encoded: isBase64Encoded
            };

            callback(null, resp);
        }
    );
};
```

Lebih banyak pola

- Periksa CloudFront distribusi Amazon untuk log akses, HTTPS, dan versi TLS
- Menerapkan aplikasi berbasis GRPC di cluster Amazon EKS dan mengaksesnya dengan Application Load Balancer
- Menyebarkan kontrol akses berbasis atribut preventif untuk subnet publik
- Menerapkan sumber daya di AWS Wavelength Zona dengan menggunakan Terraform
- Terapkan Otomatisasi Keamanan untuk AWS WAF solusi dengan menggunakan Terraform
- Siapkan router sel tanpa server untuk arsitektur berbasis sel
- Gunakan Amazon Q Developer sebagai asisten pengkodean untuk meningkatkan produktivitas Anda
- Lihat log dan metrik AWS Network Firewall dengan menggunakan Splunk

Database & penyimpanan

Topik

- Basis Data
- Penyimpanan & cadangan

Basis Data

Topik

- Mengakses tabel Microsoft SQL Server lokal dari Microsoft SQL Server di Amazon EC2 menggunakan server tertaut
- Tambahkan HA ke Oracle PeopleSoft di Amazon RDS Custom dengan menggunakan replika baca
- Menganalisis dependensi objek untuk migrasi database sebagian dari Oracle ke PostgreSQL
- Menilai kinerja kueri untuk memigrasi database SQL Server ke MongoDB Atlas di AWS
- Mengotomatiskan blue/green penyebaran database global Amazon Aurora dengan menggunakan prinsip IAc
- Otomatiskan replikasi instans Amazon RDS di seluruh Akun AWS
- Mengotomatiskan tugas database dalam edisi SQL Server Express yang berjalan di Amazon EC2 dengan menggunakan AWS Lambda dan Penjadwal Tugas
- Otomatiskan failover dan failback lintas wilayah dengan menggunakan DR Orchestrator Framework
- Secara otomatis mencadangkan database SAP HANA menggunakan Systems Manager dan EventBridge
- Secara otomatis menghasilkan model PynamoDB dan fungsi CRUD untuk Amazon DynamoDB dengan menggunakan aplikasi Python
- Blokir akses publik ke Amazon RDS dengan menggunakan Cloud Custodian
- Konfigurasikan otentikasi Windows untuk Amazon RDS untuk Microsoft SQL Server menggunakan AWS Managed Microsoft AD
- Mengonfigurasikan akses lintas akun ke Amazon DynamoDB
- Konfigurasikan perutean hanya-baca dalam grup ketersediaan Selalu Aktif di SQL Server di AWS
- Connect dengan menggunakan terowongan SSH di pgAdmin
- Mengkonversi kueri JSON Oracle ke database PostgreSQL SQL
- Salin tabel Amazon DynamoDB di seluruh akun menggunakan AWS Backup
- Salin tabel Amazon DynamoDB di seluruh akun menggunakan implementasi kustom
- Buat laporan biaya dan penggunaan terperinci untuk Amazon RDS dan Amazon Aurora
- Menerapkan instance cluster failover SQL Server di Amazon EC2 dan Amazon FSx dengan menggunakan Terraform
- Meniru array PL/SQL asosiatif Oracle di Amazon Aurora PostgreSQL dan Amazon RDS untuk PostgreSQL

- Aktifkan koneksi terenkripsi untuk instans PostgreSQL DB di Amazon RDS
- Enkripsi Amazon RDS yang ada untuk instans DB Amazon RDS for PostgreSQL
- Menerapkan penandaan otomatis database Amazon RDS saat diluncurkan
- Perkirakan biaya tabel DynamoDB untuk kapasitas sesuai permintaan
- Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB
- Perkirakan ukuran mesin Amazon RDS untuk database Oracle dengan menggunakan laporan AWR
- Ekspor Amazon RDS untuk tabel SQL Server ke bucket S3 dengan menggunakan AWS DMS
- Menangani blok anonim dalam pernyataan SQL Dinamis di Aurora PostgreSQL
- Membantu menegakkan penandaan DynamoDB
- Menerapkan pemulihan bencana lintas wilayah dengan AWS DMS dan Amazon Aurora
- Migrasikan fungsi dan prosedur Oracle yang memiliki lebih dari 100 argumen ke PostgreSQL
- Migrasikan beban kerja Redis ke Redis Enterprise Cloud di AWS
- Migrasikan SAP HANA ke AWS menggunakan SAP HSR dengan nama host yang sama
- Migrasi grup ketersediaan Microsoft SQL Server Always On menggunakan AWS Application Migration Service
- Migrasikan SQL Server ke AWS menggunakan grup ketersediaan terdistribusi
- Migrasi database relasional ke MongoDB Atlas di AWS
- Migrasikan lingkungan MongoDB yang dihosting sendiri ke MongoDB Atlas di AWS
- Migrasikan database Oracle ke Amazon DynamoDB menggunakan AWS DMS
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Oracle dan AWS DMS SharePlex
- Memigrasikan database MySQL lokal ke Amazon EC2
- Pantau Amazon Aurora untuk instance tanpa enkripsi
- Memantau GoldenGate log Oracle dengan menggunakan Amazon CloudWatch
- Replatform Oracle Database Enterprise Edition ke Edisi Standar 2 di Amazon RDS for Oracle
- Replikasi database mainframe ke AWS dengan menggunakan Exactly Connect
- Jadwalkan pekerjaan untuk Amazon RDS untuk PostgreSQL dan Aurora PostgreSQL dengan menggunakan Lambda dan Secrets Manager
- Mengirim pemberitahuan untuk instans database Amazon RDS for SQL Server dengan menggunakan server SMTP lokal dan Database Mail

- Siapkan pemulihan bencana untuk SAP di IBM Db2 di AWS
- Siapkan CI/CD pipeline untuk migrasi database dengan menggunakan Terraform
- Mengatur cluster failover Microsoft SQL Server di Amazon EC2 menggunakan Windows FSx File Server
- Menyiapkan HA/DR arsitektur untuk Oracle E-Business Suite di Amazon RDS Custom dengan database siaga aktif
- Streaming data dari IBM Db2, SAP, Sybase, dan database lainnya ke MongoDB Atlas AWS
- Peran transisi untuk PeopleSoft aplikasi Oracle di Amazon RDS Custom for Oracle
- Membongkar data dari klaster Amazon Redshift di seluruh akun ke Amazon S3
- Pola migrasi database berdasarkan beban kerja
- Lebih banyak pola

Mengakses tabel Microsoft SQL Server lokal dari Microsoft SQL Server di Amazon EC2 menggunakan server tertaut

Tirumala Dasari dan Eduardo Valentim, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara mengakses tabel database Microsoft SQL Server lokal yang berjalan di Microsoft Windows, dari database Microsoft SQL Server yang berjalan atau dihosting di instans Amazon Elastic Compute Cloud (EC2Amazon) Windows atau Linux dengan menggunakan server tertaut.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Amazon EC2 dengan Microsoft SQL Server berjalan di Amazon Linux AMI (Gambar Mesin Amazon)
- AWS Direct Connect antara server Microsoft SQL Server (Windows) lokal dan instans Windows atau Linux EC2

Versi produk

- SQL Server 2016 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server lokal yang berjalan di Windows
- Amazon EC2 dengan Microsoft SQL Server berjalan pada Windows AMI atau Linux AMI

Tumpukan teknologi target

- Amazon EC2 dengan Microsoft SQL Server berjalan di Amazon Linux AMI
- Amazon EC2 dengan Microsoft SQL Server berjalan pada Windows AMI

Arsitektur basis data sumber dan target

Alat

- Microsoft SQL Server Management Studio (SSMS) adalah lingkungan terintegrasi untuk mengelola infrastruktur SQL Server. Ini menyediakan antarmuka pengguna dan sekelompok alat dengan editor skrip kaya yang berinteraksi dengan SQL Server.

Epik

Ubah mode otentikasi ke Windows untuk SQL Server di Windows SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke Windows SQL Server melalui SSMS.		DBA
Ubah mode otentikasi ke Windows di SQL Server dari menu konteks (klik kanan) untuk instance Windows SQL Server.		DBA

Mulai ulang layanan Windows MSSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai ulang layanan SQL.	1. Di SSMS Object Explorer, pilih instance SQL Server. 2. Buka menu konteks (klik kanan). 3. Pilih Mulai Ulang.	DBA

Buat login baru dan pilih database untuk diakses di Windows SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Di tab Keamanan, buka menu konteks (klik kanan) untuk Login dan pilih login baru.		DBA
Di tab Umum, pilih otentikasi SQL Server, masukkan nama pengguna, masukkan kata sandi, lalu konfirmasikan kata sandi dan hapus opsi untuk mengubah kata sandi pada login berikutnya.		DBA
Di tab Peran Server, pilih Publik.		DBA
Di tab Pemetaan Pengguna, pilih database dan skema yang ingin Anda akses, lalu sorot database untuk memilih peran database.	Pilih public dan db_datareader untuk mengakses data dari tabel database.	DBA
Pilih OK untuk membuat pengguna.		DBA

Tambahkan IP Windows SQL Server ke file host Linux SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke kotak Linux SQL Server melalui jendela terminal.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka file /etc/hosts dan tambahkan alamat IP mesin Windows dengan SQL Server.		DBA
Simpan file host.		DBA

Buat server tertaut di Linux SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server tertaut dengan menggunakan prosedur tersimpan master.sys.sp_addlinkedserver dan master.dbo.sp_addlinkedsrvlogin.	Untuk informasi selengkapnya tentang menggunakan prosedur tersimpan ini, lihat bagian Informasi tambahan.	DBA, Pengembang

Verifikasi server dan database tertaut yang dibuat di SSMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Di Linux SQL Server di SSMS, buka Server Tertaut dan segarkan.		DBA
Perluas server dan katalog tertaut yang dibuat di panel kiri.	Anda akan melihat database SQL Server yang dipilih dengan tabel dan tampilan.	DBA

Verifikasi bahwa Anda dapat mengakses tabel database Windows SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Di jendela kueri SSMS, jalankan kueri: “pilih 3 teratas * dari [sqllin] .dms_samp le_win.dbo.mlb_data”.	Perhatikan bahwa klausa FROM menggunakan sintaks empat bagian: computer.database. schema.table (misalnya , PILIH nama "database ” FROM [sqllin] .master.s ys.databases). SQL2 Dalam contoh kami, kami membuat alias untuk SQL2 dalam file host, jadi Anda tidak perlu memasukkan nama NetBIOS yang sebenarnya di antara tanda kurung siku. Jika Anda menggunakan nama NetBIOS yang sebenarnya, perhatikan bahwa AWS default ke nama NetBIOS seperti Win-XXXX, dan SQL Server memerlukan tanda kurung siku untuk nama dengan tanda hubung.	DBA, Pengembang

Sumber daya terkait

- Catatan rilis untuk SQL Server di Linux

Informasi tambahan

Menggunakan prosedur tersimpan untuk membuat server tertaut

SSMS tidak mendukung pembuatan server tertaut untuk Linux SQL Server, jadi Anda harus menggunakan prosedur tersimpan ini untuk membuatnya:

```
EXEC master.sys.sp_addlinkedserver @server= N'SQLLIN' , @srvproduct= N'SQL Server'
EXEC master.dbo.sp_addlinkedsrvlogin
 @rmtsrvname=N'SQLLIN',@useself=N'False',@locallogin=NULL,@rmtuser=N'username',@rmtpassword='Te
```

Catatan 1: Masukkan kredenal masuk yang Anda buat sebelumnya di Windows SQL Server dalam prosedur tersimpan. `master.dbo.sp_addlinkedsrvlogin`

Catatan 2: `@server` nama `SQLLIN` dan nama entri file host `172.12.12.4 SQLLIN` harus sama.

Anda dapat menggunakan proses ini untuk membuat server tertaut untuk skenario berikut:

- Linux SQL Server ke Windows SQL Server melalui server tertaut (seperti yang ditentukan dalam pola ini)
- Windows SQL Server ke Linux SQL Server melalui server tertaut
- Linux SQL Server ke Linux SQL Server lain melalui server yang terhubung

Tambahkan HA ke Oracle PeopleSoft di Amazon RDS Custom dengan menggunakan replika baca

sampath kathirvel, Amazon Web Services

Ringkasan

Untuk menjalankan solusi Oracle PeopleSoft Enterprise Resource Planning (ERP) di Amazon Web Services (AWS), Anda dapat menggunakan Amazon Relational Database Service (Amazon RDS) atau Amazon RDS Custom for Oracle, yang mendukung aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi dan lingkungan database yang mendasarinya. Untuk faktor kunci yang perlu dipertimbangkan saat merencanakan migrasi, lihat Strategi migrasi database Oracle di AWS Prescriptive Guidance.

Pada tulisan ini, RDS Custom for Oracle tidak mendukung opsi Multi-AZ, yang tersedia untuk Amazon RDS for Oracle sebagai solusi HA menggunakan replikasi penyimpanan. Sebaliknya, pola ini mencapai HA dengan menggunakan database siaga yang membuat dan memelihara salinan fisik database utama. Pola ini berfokus pada langkah-langkah untuk menjalankan database PeopleSoft aplikasi di Amazon RDS Custom dengan HA dengan menggunakan Oracle Data Guard untuk menyiapkan replika baca.

Pola ini juga mengubah replika baca menjadi mode hanya-baca. Memiliki replika baca Anda dalam mode hanya-baca memberikan manfaat tambahan:

- Memunggah beban kerja hanya-baca dari database utama
- Mengaktifkan perbaikan otomatis blok rusak dengan mengambil blok sehat dari database siaga menggunakan fitur Oracle Active Data Guard
- Menggunakan kemampuan Far Sync untuk menjaga database siaga jarak jauh tetap sinkron tanpa overhead kinerja yang terkait dengan transmisi log redo jarak jauh.

Menggunakan replika dalam mode read-only memerlukan opsi Oracle Active Data Guard, yang datang dengan biaya tambahan karena merupakan fitur berlisensi terpisah dari Oracle Database Enterprise Edition.

Prasyarat dan batasan

Prasyarat

- PeopleSoft Aplikasi yang ada di Amazon RDS Custom. Jika Anda tidak memiliki aplikasi, lihat polanya Migrasikan Oracle ke PeopleSoft Amazon RDS Custom.
- Tingkat PeopleSoft aplikasi tunggal. Namun, Anda dapat menyesuaikan pola ini untuk bekerja dengan beberapa tingkatan aplikasi.
- Amazon RDS Kustom dikonfigurasi dengan setidaknya 8 GB ruang swap.
- Lisensi database Oracle Active Data Guard untuk mengubah replika baca menjadi mode hanya-baca dan menggunakannya untuk membongkar tugas pelaporan ke siaga. Untuk informasi selengkapnya, lihat Daftar Harga Komersil Teknologi Oracle.

Keterbatasan

- Batasan umum dan konfigurasi yang tidak didukung untuk RDS Custom for Oracle
- Keterbatasan yang terkait dengan Amazon RDS Custom untuk replika baca Oracle

Versi produk

- Untuk versi Oracle Database yang didukung oleh Amazon RDS Custom, lihat RDS Custom for Oracle.
- Untuk kelas instans Oracle Database yang didukung oleh Amazon RDS Custom, lihat dukungan kelas instans DB untuk RDS Custom for Oracle.

Arsitektur

Tumpukan teknologi target

- Amazon RDS Custom for Oracle
- AWS Secrets Manager
- Penjaga Data Aktif Oracle
- Aplikasi Oracle PeopleSoft

Arsitektur target

Diagram berikut menunjukkan instans Amazon RDS Custom DB dan replika baca Amazon RDS Custom. Replika baca menggunakan Oracle Active Data Guard untuk mereplikasi ke Availability Zone

lain. Anda juga dapat menggunakan replika baca untuk membongkar lalu lintas baca di database utama dan untuk tujuan pelaporan.

Untuk arsitektur representatif yang menggunakan Oracle PeopleSoft di AWS, lihat Menyiapkan PeopleSoft arsitektur yang sangat tersedia di AWS.

Alat

Layanan AWS

- Amazon RDS Custom for Oracle adalah layanan database terkelola untuk aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi dan lingkungan database yang mendasarinya.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Dalam pola ini, Anda mengambil kata sandi pengguna database dari Secrets Manager `RDS_DATAGUARD` dengan nama `do-not-delete-rds-custom-+<<RDS Resource ID>>+-dg` rahasia.

Alat Lainnya

- Oracle Data Guard membantu Anda membuat, memelihara, mengelola, dan memantau database siaga.

Praktik terbaik

Untuk bekerja menuju tujuan zero data loss (RPO = 0), gunakan mode perlindungan `MaxAvailability` Data Guard, dengan `SYNC+NOAFFIRM` pengaturan redo transport untuk kinerja yang lebih baik. Untuk informasi selengkapnya tentang memilih mode perlindungan basis data, lihat bagian Informasi tambahan.

Epik

Buat replika baca

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat replika baca.	Untuk membuat replika baca instans Amazon RDS Custom DB, ikuti petunjuk dalam dokumentasi Amazon RDS dan gunakan instans Amazon RDS Custom DB yang Anda buat (lihat bagian Prasyarat) sebagai database sumber. Secara default, replika baca Kustom Amazon RDS dibuat sebagai siaga fisik dan dalam keadaan terpasang. Ini disengaja untuk memastikan kepatuhan dengan lisensi Oracle Active Data Guard. Pola ini mencakup kode untuk menyiapkan database kontainer multitenant (CDB) atau instance non-CDB.	DBA

Ubah mode perlindungan Oracle Data Guard menjadi MaxAvailability

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses konfigurasi broker Data Guard pada database utama.	Dalam contoh ini, replika baca Kustom Amazon RDS adalah `RDS_CUSTOM_ORCL_D` untuk instans non-CDB dan `RDS_CUSTOM_RDSCDB_`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	B untuk instance CDB. Database untuk non-CDB adalah `orcl_a` (primer) dan `orcl_d` (siaga). Nama database untuk CDB adalah `rdscdb_a` (primer) dan `rdscdb_b` (siaga). Anda dapat terhubung ke replika baca Kustom RDS secara langsung atau melalui database utama. Anda dapat menemukan nama layanan bersih untuk database Anda di `tnsnames.ora` file yang terletak di `$ORACLE_HOME/network/admin` direktori . RDS Custom for Oracle secara otomatis mengisi entri ini untuk database utama Anda dan replika baca Anda. Kata sandi untuk `RDS_DATAGUARD` pengguna disimpan di AWS Secrets Manager, dengan nama rahasia`do-not-delete-rds-custom-+<<RDS Resource ID>>+-dg`. Untuk informasi selengkapnya tentang cara menyambung ke instans Kustom RDS menggunakan kunci SSH (Secure Shell) yang diambil dari Secrets Manager, lihat	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Menyambung ke instans RDS Custom DB menggunakan SSH.
	
	Untuk mengakses konfigurasi broker Oracle Data Guard melalui baris perintah Data Guard (`dgmgrl`), gunakan kode berikut.
	
	Non-CDB
	
	`$ dgmgrl RDS_DATAGUARD@RDS_CUSTOM_ORCL_D`
	`DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Sep 30 22:44:49 2022`
	`Version 19.10.0.0.0`
	`Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.`
	`Welcome to DGMGRL, type "help" for information.`
	`Password:`
	`Connected to "ORCL_D"`
	`Connected as SYSDG.`
	`DGMGRL>`
	`DGMGRL> show database orcl_d`
	`Database - orcl_d`
	`Role: PHYSICAL STANDBY`
	`Intended State: APPLY-ON`
	`Transport Lag: 0 seconds (computed 0 seconds ago)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 11.00 KByte/s Instance(s): ORCL SUCCESS DGMGRL> CDB -bash-4.2$ dgmgrl C##RDS_DATAGUARD@RDS_CUSTOM_RDSCDB_B DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 11 20:24:11 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for information. Password: Connected to "RDSCDB_B" Connected as SYSDG. DGMGRL> DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> Transport Lag:
	0 seconds (computed 1
	second ago)
	  Apply Lag:
	0 seconds (computed 1
	second ago)
	  Average Apply Rate:
	2.00 KByte/s
	  Real Time Query:
	OFF
	  Instance(s):
	    RDSCDB
	Database Status:
	SUCCESS
	DGMGRL></pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah pengaturan transport log dengan menghubungkan ke DGMGRL dari node utama.	Ubah mode transportasi log menjadi`FastSync`, sesuai dengan pengaturan `SYNC +NOAFFIRM` redo transport . Untuk memastikan bahwa Anda memiliki pengaturan yang valid setelah peralihan peran, ubah untuk database utama dan database siaga. Non-CDB `DGMGRL> DGMGRL> edit database orcl_d set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database orcl_d LogXptMode; LogXptMode = 'fastsync ' DGMGRL> edit database orcl_a set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database orcl_a logxptmode; LogXptMode = 'fastsync ' DGMGRL>`  CDB  `DGMGRL> edit database rdscdb_b set property logxptmode=fastsyn c;DGMGRL> edit database`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	rdscdb_b set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database rdscdb_b LogXptMode; LogXptMode = 'fastsync' DGMGRL> edit database rdscdb_a set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database rdscdb_a logxptmode; LogXptMode = 'fastsync' DGMGRL>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah mode perlindungan ke MaxAvailability.	Ubah mode perlindungan `MaxAvailability` dengan menghubungkan ke `DGMGRL` dari simpul utama. Non-CDB `DGMGRL> edit configura tion set protectio n mode as maxavaila bility; Succeeded. DGMGRL> show configura tion; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 38 seconds ago) DGMGRL>`  CDB  `DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 57 seconds ago) DGMGRL>	

Ubah status replika dari mount menjadi read-only dan aktifkan redo apply

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Stop redo mendaftar untuk database siaga.	Replika baca dibuat dalam MOUNT mode secara default. Untuk membukanya dalam mode read-only, pertama-tama Anda harus mematikan redo apply dengan menghubungkan ke DGMGRL dari node primer atau siaga. Non-CDB DGMGRL> show database orcl_dDGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 11.00 KByte/s	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Real Time Query: OFF Instance(s): ORCL Database Status: SUCCESS DGMGRL> edit database orcl_d set state=app ly-off; Succeeded. DGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY- OFF Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 42 seconds (computed 1 second ago) Average Apply Rate: (unknown) Real Time Query: OFF Instance(s): ORCL Database Status: SUCCESS DGMGRL> CDB DGMGRL> show configura tionDGMGRL> show configuration Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 57 seconds ago) DGMGRL> show database rdscdb_b; Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 2.00 KByte/s Real Time Query: OFF Instance(s): RDSCDB Database Status: SUCCESS DGMGRL> edit database rdscdb_b set state=apply-off; Succeeded. DGMGRL> show database rdscdb_b; Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-OFF Transport Lag: 0 seconds (computed 1 second ago)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> Apply Lag:
	0 seconds (computed 1
	second ago)
	  Average Apply Rate:
	(unknown)
	  Real Time Query:
	OFF
	  Instance(s):
	    RDSCDB
	Database Status:
	SUCCESS</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buka contoh replika baca dalam mode hanya-baca.	Connect ke database siaga dengan menggunakan entri TNS, dan buka dalam mode read-only dengan menghubungkannya dari node primer atau siaga.
	
	Non-CDB
	
	`$ sqlplus RDS_DATAGUARD@RDS_CUSTOM_ORCL_D as sysdg`
	`-bash-4.2$ sqlplus RDS_DATAGUARD@RDS_CUSTOM_ORCL_D as sysdg`
	`SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 30 23:00:14 2022`
	`Version 19.10.0.0.0`
	`Copyright (c) 1982, 2020, Oracle. All rights reserved.`
	`Enter password:`
	`Last Successful login time: Fri Sep 30 2022 22:48:27 +00:00`
	`Connected to:`
	`Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production`
	`Version 19.10.0.0.0`
	`SQL> select open_mode from v$database;`
	`OPEN_MODE`
	`--------------------`
	`MOUNTED`
	`SQL> alter database open read only;`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Database altered. SQL> select open_mode from v$database; OPEN_MODE -------------------- READ ONLY SQL> CDB -bash-4.2$ sqlplus C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B as sysdg SQL*Plus: Release 19.0.0.0.0 - Productio n on Wed Jan 11 21:14:07 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2022, Oracle. All rights reserved. Enter password: Last Successful login time: Wed Jan 11 2023 21:12:05 +00:00 Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.16.0.0.0 SQL> select name,open _mode from v$database; NAME OPEN_MODE --------- --------- ----------- RDSCDB MOUNTED SQL> alter database open read only; Database altered.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	SQL> select name,open _mode from v$database; NAME OPEN_MODE --------- ---------- ----------- RDSCDB READ ONLY SQL>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan redo apply pada instance replika baca.	Aktifkan redo apply pada instance replika baca dengan menggunakan DGMGR L dari node primer atau siaga. Non-CDB `$ dgmgrl RDS_DATAG UARD@RDS_CUSTOM_OR CL_D DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Sep 30 23:02:16 2022 Version 19.10.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "ORCL_D" Connected as SYSDG. DGMGRL> edit database orcl_d set state=apply-on; DGMGRL> edit database orcl_d set state=app ly-on; Succeeded. DGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY-ON`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 496.00 KByte/s Real Time Query: ON Instance(s): ORCL Database Status: SUCCESS DGMGRL> CDB -bash-4.2$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B -bash-4.2$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 11 21:21:11 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_B " Connected as SYSDG.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	DGMGRL> edit database rdscdb_b set state=apply-on; Succeeded. DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 35.00 KByte/s Real Time Query: ON Instance(s): RDSCDB Database Status: SUCCESS DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 16.00 KByte/s Real Time Query: ON Instance(s): RDSCDB	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Database Status: SUCCESS DGMGRL>`	

Sumber daya terkait

- Mengkonfigurasi Amazon RDS sebagai Oracle Database PeopleSoft (whitepaper AWS)
- Panduan Oracle Data Guard Broker (dokumentasi referensi Oracle)
- Konsep dan Administrasi Data Guard (dokumentasi referensi Oracle)

Informasi tambahan

Pilih mode perlindungan basis data Anda

Oracle Data Guard menyediakan tiga mode perlindungan untuk mengonfigurasi lingkungan Data Guard Anda berdasarkan ketersediaan, perlindungan, dan persyaratan kinerja Anda. Tabel berikut merangkum ketiga mode ini.

Modus perlindungan	Ulangi pengaturan transportasi	Deskripsi
PERFORMA MAKSIMAL	`ASYNC`	Untuk transaksi yang terjadi pada database utama, data redo ditransmisikan secara asinkron dan ditulis ke log redo database siaga. Oleh karena itu, dampak kinerjanya minimal. `MaxPerformance` tidak dapat memberikan RPO = 0 karena pengiriman log asinkron.
PERLINDUNGAN MAKSIMAL	`SYNC+AFFIRM`	Untuk transaksi pada database utama, data redo

		ditransmisikan secara sinkron dan ditulis ke database siaga redo log on disk sebelum transaksi diakui. Jika database siaga menjadi tidak tersedia, database utama menutup dirinya sendiri untuk memastikan transaksi dilindungi.
KETERSEDIAAN MAKSIMUM	`SYNC+AFFIRM`	Ini mirip dengan `MaxProtection` mode, kecuali ketika tidak ada pengakuan yang diterima dari database siaga. Dalam hal ini, ia beroperasi seolah-olah berada dalam `MaxPerformance` mode untuk mempertahankan ketersediaan basis data utama hingga dapat menulis aliran redo ke database siaga yang disinkronkan lagi.

`SYNC+NOAFFIRM`	Untuk transaksi pada database utama, redo ditransmisikan secara sinkron ke database siaga, dan primer hanya menunggu pengakuan bahwa pengulangan telah diterima pada siaga, bukan bahwa itu telah ditulis ke disk siaga. Mode ini, yang juga dikenal sebagai`FastSync`, dapat memberikan manfaat kinerja dengan mengorban kan potensi paparan kehilanga n data dalam kasus khusus beberapa kegagalan simultan.

Baca replika di RDS Custom for Oracle dibuat dengan mode perlindungan kinerja maksimum, yang juga merupakan mode perlindungan default untuk Oracle Data Guard. Mode kinerja maksimum memberikan dampak kinerja terendah pada database utama, yang dapat membantu Anda memenuhi persyaratan tujuan titik pemulihan (RPO) yang diukur dalam hitungan detik.

Untuk bekerja untuk mencapai tujuan nol kehilangan data (RPO = 0), Anda dapat menyesuaikan mode perlindungan Oracle Data Guard `MaxAvailability` dengan `SYNC+NOAFFIRM` pengaturan untuk redo transport untuk kinerja yang lebih baik. Karena komit pada database utama diakui hanya setelah vektor redo yang sesuai berhasil ditransmisikan ke database siaga, latensi jaringan antara instance utama dan replika dapat menjadi penting untuk beban kerja yang sensitif terhadap komitmen. Sebaiknya lakukan pengujian beban untuk beban kerja Anda guna menilai dampak kinerja saat replika baca disesuaikan untuk dijalankan dalam `MaxAvailability` mode.

Menerapkan replika baca di Availability Zone yang sama dengan database utama memberikan latensi jaringan yang lebih rendah dibandingkan dengan menerapkan replika baca di Availability Zone yang berbeda. Namun, menerapkan replika primer dan baca di Availability Zone yang sama mungkin tidak memenuhi persyaratan HA Anda karena, jika tidak tersedianya Availability Zone, instance primer dan instance replika baca akan terpengaruh.

Menganalisis dependensi objek untuk migrasi database sebagian dari Oracle ke PostgreSQL

Situs web, Amazon Web Services

Ringkasan

Pola ini menjelaskan pentingnya mengidentifikasi dan mengelola dependensi sistem secara sistematis saat memigrasikan sebagian database Oracle ke Amazon Relational Database Service (Amazon RDS) atau Amazon Aurora PostgreSQL. Dalam migrasi sebagian, hanya subset objek database dan data dari database asli yang dimigrasikan, sedangkan database sumber terus beroperasi dan melayani aplikasi yang bergantung pada komponen yang tidak bermigrasi.

Anda harus mengidentifikasi dan menganalisis ruang lingkup migrasi saat menangani database skala besar yang memiliki aplikasi yang digabungkan erat dengan dependensi hulu dan hilir. Untuk memulai migrasi sebagian, identifikasi objek lingkup, termasuk tabel, pemicu, tampilan, prosedur tersimpan, fungsi, dan paket. Proses identifikasi ruang lingkup mengikuti pendekatan komprehensif:

- Objek lingkup tingkat pertama diidentifikasi melalui referensi langsung dalam kode aplikasi dan pekerjaan khusus modul kritis.
- Objek tingkat kedua diturunkan melalui analisis ketergantungan yang komprehensif.

Ketika Anda memahami bagaimana bagian-bagian yang berbeda dari sistem Anda berinteraksi, Anda dapat merencanakan urutan yang benar untuk memindahkan komponen database dengan lebih baik dan mengurangi risiko kegagalan migrasi. Tabel berikut mencantumkan berbagai jenis analisis ketergantungan.

Jenis analisis	Area fokus	Tujuan
Ketergantungan objek	• Tabel • Tampilan • Prosedur Tersimpan • Fungsi • Pemicu	Mengidentifikasi hubungan antara objek database dan struktur hierarkisnya

Dependensi segmen	• Hubungan kunci asing • Gantungan kunci primer • Referensi lintas skema	Memetakan hubungan data dan menjaga integritas referensial
Dependensi keamanan	• Izin pengguna • Hirarki peran • Hak istimewa objek	Memastikan migrasi kontrol akses yang tepat dan pemeliharaan keamanan
Pola akses	• Operasi baca • Operasi tulis	Menentukan pola interaksi database

Untuk menjaga konsistensi antara sistem sumber dan target, buat mekanisme sinkronisasi data selama masa transisi. Anda juga harus memodifikasi kode aplikasi dan fungsi untuk menangani distribusi data di kedua sumber Oracle dan target database PostgreSQL.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Database Oracle (sumber)
- Amazon RDS atau Amazon Aurora PostgreSQL (target)

Versi produk

- Oracle 19c atau yang lebih baru
- PostgreSQL 16 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Oracle 19c atau yang lebih baru

Tumpukan teknologi target

- Amazon RDS atau Amazon Aurora PostgreSQL

Arsitektur target

Diagram berikut menunjukkan proses migrasi dari database Oracle lokal ke Amazon RDS for Oracle, yang melibatkan hal-hal berikut:

- Mengidentifikasi dependensi database
- Migrasi kode database dan objek menggunakan AWS Schema Conversion Tool ()AWS SCT
- Migrasi data menggunakan AWS Database Migration Service ()AWS DMS
- Mereplikasi perubahan yang sedang berlangsung melalui change data capture (CDC) menggunakan AWS DMS

Untuk informasi selengkapnya, lihat Mengintegrasikan AWS Database Migration Service dengan AWS Schema Conversion Tool dalam AWS dokumentasi.

Alat

Layanan AWS

- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di. AWS Cloud
- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke dalam AWS Cloud atau di antara kombinasi pengaturan cloud dan lokal.

Layanan lainnya

- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi yang menyederhanakan pengembangan dan pengelolaan database Oracle baik dalam penerapan tradisional maupun berbasis cloud. Untuk pola ini, Anda dapat menggunakan SQL* Plus.

Praktik terbaik

Untuk praktik terbaik tentang penyediaan dan migrasi database Oracle, lihat Praktik terbaik untuk bermigrasi ke Amazon RDS for Oracle.

Epik

Identifikasi dependensi objek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel objek.	Identifikasi objek yang penting untuk fungsiona litas aplikasi, dan buat tabel bernama`DEPENDENT _ANALYSIS_BASELINE` . Tambahkan catatan untuk setiap objek ke tabel. Sebagai contoh, lihat bagian Informasi tambahan.	Insinyur data, DBA
Buat prosedur database.	Buat prosedur tersimpan bernama `sp_object _dependency_analys is` untuk menganalisis dependensi objek di kedua arah (maju dan mundur) dengan menggunakan data dari tabel. `DBA_DEPEN DENCIES` Sebagai contoh, lihat bagian Informasi tambahan.	Insinyur data, DBA
Jalankan prosedurnya.	Jalankan skrip di setiap tingkat berturut-turut sampai tidak ada dependensi objek baru yang ditemukan. Semua dependensi dan	DBA, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	level disimpan dalam tabel. `DEPENDENT_ANALYSIS_BASELINE`	

Buat prosedur untuk dependensi tingkat segmen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel ketergantungan.	Buat tabel dependensi tingkat segmen bernama. `REFERENTIAL_ANALYSIS_BASELINE` Ketika semua dependensi tingkat objek ditemukan, periksa tabel induk dengan menanyakan tabel. `DEPENDENT_ANALYSIS_BASELINE` `DBA_CONSTRAINT` Kecualikan dependensi di mana tabel dasar direferensikan oleh tabel lain. Penimbunan ulang menangani hubungan ini. Berikut ini adalah contoh skrip: `CREATE TABLE REFERENTIAL_ANALYSIS_BASELINE (CHILD_OWNER VARCHAR2(50 BYTE), CHILD_NAME VARCHAR2(100 BYTE), PARENT_OWNER VARCHAR2(50 BYTE), PARENT_NAME VARCHAR2(50 BYTE),`	Insinyur data, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`REFERENCE_PATH VARCHAR2(1000 BYTE));`	
Buat prosedur database.	Buat prosedur yang disebut`SP_OBJECT _REFERENTIAL_ANALY SIS` , dan hasilkan analisis referensial untuk semua objek yang diidentifikasi. Sebagai contoh, lihat bagian Informasi tambahan.	Insinyur data, DBA
Jalankan prosedurnya.	Jalankan prosedur untuk mendapatkan dependensi referensial. Hasilkan detail objek analisis referensial di`REFERENTIAL_ANALYS IS_BASELINE` .	Insinyur data, DBA

Identifikasi objek yang membaca dan menulis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel untuk membaca dan menulis objek.	Gunakan skrip berikut untuk membuat tabel objek baca bernama `TABLE_READ_OBJECT_ DETAILS` dan tabel objek tulis bernama`TABLE_WRI TE_OBJECT_DETAILS` : `CREATE TABLE TABLE_REA D_OBJECT_DETAILS (OWNER VARCHAR2(50 BYTE),`	Insinyur data, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`TAB_NAME VARCHAR2(50 BYTE), READER_OWNER VARCHAR2(50 BYTE), READER_NAME VARCHAR2(50 BYTE), READER_TYPE VARCHAR2(50 BYTE));`
	
	`CREATE TABLE TABLE_WRITE_OBJECT_DETAILS (TABLE_NAME VARCHAR2(100 BYTE), WRITEOBJ_OWNER VARCHAR2(100 BYTE), WRITEOBJ_NAME VARCHAR2(100 BYTE), WRITEOBJ_TYPE VARCHAR2(100 BYTE), LINE VARCHAR2(100 BYTE), TEXT VARCHAR2(4000 BYTE), OWNER VARCHAR2(50 BYTE));`	

Buat prosedur untuk analisis.	Buat prosedur `SP_READER_OBJECTS_ANALYSIS` dan `SP_WRITER_OBJECTS_ANALYSIS` untuk menganalisis objek baca dan tulis objek, masing-masing. Prosedur ini menggunakan pencocokan pola untuk menemukan objek terkait. Sebagai contoh, lihat bagian Informasi tambahan.	Insinyur data, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan prosedurnya.	Jalankan prosedur ini untuk mengidentifikasi objek dependen.	DBA, Insinyur data

Tinjau hak istimewa basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel untuk meninjau hak istimewa.	Buat tabel untuk menganalisis hak istimewa bernama`OBJECT_PRIVS_ANALYSIS`. Untuk menangkap hak istimewa objek secara rekursif dalam `DEPENDENT_ANALYSIS_BASELINE` tabel, gunakan skrip berikut: `CREATE TABLE OBJECT_PRIVS_ANALYSIS (OWNER VARCHAR2(50 BYTE), OBJECT_NAME VARCHAR2(50 BYTE), USER_NAME VARCHAR2(50 BYTE), PRIVS VARCHAR2(50 BYTE));`	Insinyur data, DBA
Buat prosedur untuk meninjau hak istimewa.	Buat prosedur bernama`SP_OBJECT_PRIVS_ANALYSIS`. Hasilkan analisis hak istimewa untuk objek yang diidentifikasi.	DBA, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Sebagai contoh, lihat bagian Informasi tambahan.	
Jalankan prosedurnya.	Jalankan prosedur untuk menangkapnya di `OBJECT_PRIVS_ANALYSIS` tabel.	DBA, Insinyur data

Pemecahan masalah

Isu	Solusi
Tidak dapat mengakses tabel kamus	Pastikan bahwa pengguna yang membuat objek analisis dapat mengakses tabel DBA.

Sumber daya terkait

Dokumentasi AWS

- Dokumentasi Amazon RDS dan Aurora
- Basis data Oracle 19c ke buku pedoman migrasi Amazon Aurora PostgreSQL
- Apa itu AWS Database Migration Service?
- Apa itu AWS Schema Conversion Tool?

Dokumentasi lainnya

- Objek database Oracle

Informasi tambahan

Script untuk **`DEPENDENT_ANALYSIS_BASELINE`**

```
CREATE TABLE DEPENDENT_ANALYSIS_BASELINE
(OWNER VARCHAR2(128 BYTE) NOT NULL ENABLE,
OBJECT_NAME VARCHAR2(128 BYTE) NOT NULL ENABLE,
```

```
OBJECT_TYPE VARCHAR2(20 BYTE),
DEPEDNCY_LEVEL NUMBER,
PROJECT_NEED VARCHAR2(20 BYTE),
CATAGORY VARCHAR2(4000 BYTE),
COMMENTS VARCHAR2(4000 BYTE),
CATAGORY1 CLOB,
COMMENTS1 CLOB,
CUSTOMER_COMMENTS VARCHAR2(1000 BYTE),
BACKFILL_TO_GUS VARCHAR2(1000 BYTE),
BACKFILL_NEAR_REAL_TIME_OR_BATCH VARCHAR2(1000 BYTE),
PK_EXISTS VARCHAR2(3 BYTE),
UI_EXISTS VARCHAR2(3 BYTE),
LOB_EXISTS VARCHAR2(3 BYTE),
MASTER_LINK VARCHAR2(100 BYTE),
CONSTRAINT PK_DEPENDENT_ANALYSIS_BASELINE PRIMARY KEY (OWNER,OBJECT_NAME,OBJECT_TYPE));
```

Prosedur untuk **SP_WRITER_OBJECTS_ANALYSIS**

```
CREATE OR REPLACE PROCEDURE SP_WRITER_OBJECTS_ANALYSIS IS
BEGIN
  EXECUTE IMMEDIATE 'TRUNCATE TABLE TABLE_WRITE_OBJECT_DETAILS';
  FOR I IN (SELECT OWNER, OBJECT_NAME FROM DEPENDENT_ANALYSIS_BASELINE WHERE
 OBJECT_TYPE = 'TABLE')
  LOOP
    INSERT INTO TABLE_WRITE_OBJECT_DETAILS(OWNER, TABLE_NAME, WRITEOBJ_OWNER,
 WRITEOBJ_NAME, WRITEOBJ_TYPE, LINE, TEXT)
    SELECT DISTINCT I.OWNER, I.OBJECT_NAME, OWNER WRITEOBJ_OWNER, NAME, TYPE, LINE,
 TRIM(TEXT)
    FROM DBA_SOURCE
    WHERE UPPER(TEXT) LIKE '%' || I.OBJECT_NAME || '%'
      AND (UPPER(TEXT) LIKE '%INSERT%' || I.OBJECT_NAME || '%'
        OR UPPER(TEXT) LIKE '%UPDATE%' || I.OBJECT_NAME || '%'
        OR UPPER(TEXT) LIKE '%DELETE%' || I.OBJECT_NAME || '%'
        OR UPPER(TEXT) LIKE '%UPSERT%' || I.OBJECT_NAME || '%'
        OR UPPER(TEXT) LIKE '%MERGE%' || I.OBJECT_NAME || '%')
      AND UPPER(TEXT) NOT LIKE '%PROCEDURE%'
      AND UPPER(TEXT) NOT LIKE 'PROCEDURE%'
      AND UPPER(TEXT) NOT LIKE '%FUNCTION%'
      AND UPPER(TEXT) NOT LIKE 'FUNCTION%'
      AND UPPER(TEXT) NOT LIKE '%TRIGGER%'
      AND UPPER(TEXT) NOT LIKE 'TRIGGER%'
      AND UPPER(TRIM(TEXT)) NOT LIKE '%AFTER UPDATE%'
      AND UPPER(TRIM(TEXT)) NOT LIKE 'BEFORE UPDATE%'
```

```
      AND UPPER(TRIM(TEXT)) NOT LIKE 'BEFORE INSERT%'
      AND UPPER(TRIM(TEXT)) NOT LIKE 'AFTER INSERT%'
      AND UPPER(TRIM(TEXT)) NOT LIKE 'BEFORE DELETE%'
      AND UPPER(TRIM(TEXT)) NOT LIKE 'AFTER DELETE%'
      AND UPPER(TRIM(TEXT)) NOT LIKE '%GGLOGADM.GG_LOG_ERROR%'
      AND (TRIM(TEXT) NOT LIKE '/*%' AND TRIM(TEXT) NOT LIKE '--%' )
      AND (OWNER, NAME, TYPE) IN (SELECT OWNER, NAME, TYPE FROM DBA_DEPENDENCIES WHERE
 REFERENCED_NAME = I.OBJECT_NAME);
  END LOOP;
END;
```

Script untuk **SP_READER_OBJECTS_ANALYSIS**

```
CREATE OR REPLACE PROCEDURE SP_READER_OBJECTS_ANALYSIS IS
BEGIN
  EXECUTE IMMEDIATE 'TRUNCATE TABLE TABLE_READ_OBJECT_DETAILS';
  FOR I IN (SELECT OWNER, OBJECT_NAME FROM DEPENDENT_ANALYSIS_BASELINE WHERE
 OBJECT_TYPE = 'TABLE')
  LOOP
    INSERT INTO TABLE_READ_OBJECT_DETAILS
    SELECT DISTINCT i.owner, i.object_name, owner, name, type
    FROM dba_dependencies
    WHERE referenced_name = I.OBJECT_NAME
    AND referenced_type = 'TABLE'
    AND type NOT IN ('SYNONYM', 'MATERIALIZED VIEW', 'VIEW')
    AND (owner, name, type) NOT IN (
      SELECT DISTINCT owner, trigger_name, 'TRIGGER'
      FROM dba_triggers
      WHERE table_name = I.OBJECT_NAME
      AND table_owner = i.owner
      UNION ALL
      SELECT DISTINCT owner, name, type
      FROM dba_source
      WHERE upper(text) LIKE '%' || I.OBJECT_NAME || '%'
      AND (upper(text) LIKE '%INSERT %' || I.OBJECT_NAME || '%'
        OR upper(text) LIKE '%UPDATE% ' || I.OBJECT_NAME || '%'
        OR upper(text) LIKE '%DELETE %' || I.OBJECT_NAME || '%'
        OR upper(text) LIKE '%UPSERT %' || I.OBJECT_NAME || '%'
        OR upper(text) LIKE '%MERGE %' || I.OBJECT_NAME || '%')
      AND upper(text) NOT LIKE '%PROCEDURE %'
      AND upper(text) NOT LIKE 'PROCEDURE %'
      AND upper(text) NOT LIKE '%FUNCTION %'
      AND upper(text) NOT LIKE 'FUNCTION %'
```

```
      AND upper(text) NOT LIKE '%TRIGGER %'
      AND upper(text) NOT LIKE 'TRIGGER %'
      AND upper(trim(text)) NOT LIKE 'BEFORE INSERT %'
      AND upper(trim(text)) NOT LIKE 'BEFORE UPDATE %'
      AND upper(trim(text)) NOT LIKE 'BEFORE DELETE %'
      AND upper(trim(text)) NOT LIKE 'AFTER INSERT %'
      AND upper(trim(text)) NOT LIKE 'AFTER UPDATE %'
      AND upper(trim(text)) NOT LIKE 'AFTER DELETE %'
      AND (trim(text) NOT LIKE '/*%' AND trim(text) NOT LIKE '--%'));
  END LOOP;
END;
```

Script untuk **SP_OBJECT_REFERENTIAL_ANALYSIS**

```
CREATE OR REPLACE PROCEDURE SP_OBJECT_REFERENTIAL_ANALYSIS IS
BEGIN
  EXECUTE IMMEDIATE 'TRUNCATE TABLE REFERENTIAL_ANALYSIS_BASELINE';
  INSERT INTO REFERENTIAL_ANALYSIS_BASELINE
  WITH rel AS (
    SELECT DISTINCT c.owner, c.table_name, c.r_owner r_owner,
      (SELECT table_name FROM dba_constraints
       WHERE constraint_name = c.r_constraint_name
       AND owner = c.r_owner) r_table_name
    FROM dba_constraints c
    WHERE constraint_type = 'R'
    AND c.owner NOT IN (SELECT username FROM dba_users WHERE oracle_maintained = 'Y')
    AND c.r_owner NOT IN (SELECT username FROM dba_users WHERE oracle_maintained =
 'Y')),
  tab_list AS (
    SELECT OWNER, object_name
    FROM DEPENDENT_ANALYSIS_BASELINE
    WHERE UPPER(OBJECT_TYPE) = 'TABLE')
  SELECT DISTINCT owner child_owner, table_name child, r_owner parent_owner,
    r_table_name parent, SYS_CONNECT_BY_PATH(r_table_name, ' -> ') || ' -> ' ||
 table_name PATH
  FROM rel
  START WITH (r_owner, r_table_name) IN (SELECT * FROM tab_list)
  CONNECT BY NOCYCLE (r_owner, r_table_name) = ((PRIOR owner, PRIOR table_name))
  UNION
  SELECT DISTINCT owner child_owner, table_name child, r_owner parent_owner,
    r_table_name parent, SYS_CONNECT_BY_PATH(table_name, ' -> ') || ' -> ' ||
 r_table_name PATH
  FROM rel
```

```
  START WITH (owner, table_name) IN (SELECT * FROM tab_list)
  CONNECT BY NOCYCLE (owner, table_name) = ((PRIOR r_owner, PRIOR r_table_name));
END;
```

Script untuk **SP_OBJECT_PRIVS_ANALYSIS**

```
CREATE OR REPLACE PROCEDURE SP_OBJECT_PRIVS_ANALYSIS IS
  V_SQL VARCHAR2(4000);
  V_CNT NUMBER;
BEGIN
  V_SQL := 'TRUNCATE TABLE OBJECT_PRIVS_ANALYSIS';
  EXECUTE IMMEDIATE V_SQL;
  FOR I IN (SELECT OWNER, OBJECT_NAME FROM DEPENDENT_ANALYSIS_BASELINE WHERE
 OBJECT_TYPE = 'TABLE')
  LOOP
    INSERT INTO OBJECT_PRIVS_ANALYSIS(OWNER, OBJECT_NAME, USER_NAME, PRIVS)
    WITH obj_to_role AS (
      SELECT DISTINCT GRANTEE role_name,
        DECODE(privilege, 'SELECT', 'READ', 'REFERENCE', 'READ', 'INSERT', 'WRITE',
               'UPDATE', 'WRITE', 'DELETE', 'WRITE', privilege) privs
      FROM DBA_TAB_PRIVS t, DBA_ROLES r
      WHERE OWNER = I.OWNER
      AND TYPE = 'TABLE'
      AND TABLE_NAME = I.OBJECT_NAME
      AND t.GRANTEE = r.ROLE
      AND r.ROLE IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED = 'N')
    )
    SELECT I.OWNER, I.OBJECT_NAME, grantee, privs
    FROM (
      -- Recursively Role to User mapping with privilege
      SELECT DISTINCT grantee, privs
      FROM (SELECT rp.granted_role, rp.grantee, privs,
        (SELECT DECODE(COUNT(*), 0, 'ROLE', 'USER')
         FROM (SELECT 'User' FROM DBA_users WHERE username = rp.GRANTEE)) grantee_type
        FROM DBA_role_privs rp, obj_to_role r
        WHERE rp.granted_role = r.role_name
        AND grantee IN ((SELECT USERNAME FROM DBA_USERS WHERE ORACLE_MAINTAINED = 'N')
                        UNION (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED =
 'N'))
        AND granted_role IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED = 'N')
        START WITH granted_role IN (SELECT DISTINCT role_name FROM obj_to_role)
        CONNECT BY granted_role = PRIOR grantee)
      WHERE grantee_type = 'USER'
```

```
    )
    UNION
    (
      -- Direct Object grants to User
      SELECT I.OWNER, I.OBJECT_NAME, GRANTEE,
        DECODE(privilege, 'SELECT', 'READ', 'REFERENCE', 'READ', 'INSERT', 'WRITE',
               'UPDATE', 'WRITE', 'DELETE', 'WRITE', privilege) privs
      FROM DBA_TAB_PRIVS, DBA_USERS
      WHERE GRANTEE = USERNAME
      AND OWNER = I.OWNER
      AND TYPE = 'TABLE'
      AND TABLE_NAME = I.OBJECT_NAME
    )
    ORDER BY 2 DESC;
  END LOOP;
END;
```

Prosedur untuk **SP_OBJECT_DEPENDENCY_ANALYSIS**

```
CREATE OR REPLACE PROCEDURE SP_OBJECT_DEPENDENCY_ANALYSIS (v_level NUMBER) IS
  TYPE typ IS RECORD (
    schema VARCHAR2(100),
    obj_type VARCHAR2(100),
    obj_name VARCHAR2(100),
    path VARCHAR2(5000)
  );
  TYPE array IS TABLE OF typ;
  l_data array;
  c SYS_REFCURSOR;
  l_errors NUMBER;
  l_errno NUMBER;
  l_msg VARCHAR2(4000);
  l_idx NUMBER;
  l_level NUMBER;
BEGIN
  l_level := v_level + 1;
  OPEN c FOR
    WITH obj_list AS (
      SELECT owner schema_name, object_type, object_name
      FROM DEPENDENT_ANALYSIS_BASELINE
      WHERE depedncy_level = v_level
    ),
    fw_dep_objects AS (
```

```
      SELECT level lvl, owner, name, type, referenced_owner, referenced_name,
        referenced_type, SYS_CONNECT_BY_PATH(name, ' -> ') || ' -> ' || referenced_name
PATH
      FROM dba_dependencies
      START WITH (owner, CASE WHEN type = 'PACKAGE BODY' THEN 'PACKAGE' ELSE type END,
name)
        IN (SELECT schema_name, object_type, object_name FROM obj_list)
      CONNECT BY NOCYCLE (owner, type, name) =
        ((PRIOR referenced_owner, PRIOR referenced_type, PRIOR referenced_name))
    ),
    bw_dep_objects AS (
      SELECT level lvl, owner, name, type, referenced_owner, referenced_name,
        referenced_type, SYS_CONNECT_BY_PATH(name, ' <- ') || ' <- ' || referenced_name
PATH
      FROM dba_dependencies
      START WITH (referenced_owner, CASE WHEN referenced_type = 'PACKAGE BODY' THEN
'PACKAGE'
        ELSE referenced_type END, referenced_name) IN (SELECT schema_name, object_type,
object_name FROM obj_list)
      CONNECT BY NOCYCLE (referenced_owner, referenced_type, referenced_name) =
        ((PRIOR owner, PRIOR type, PRIOR name))
    )
    SELECT * FROM (
      (SELECT DISTINCT referenced_owner schema, referenced_type obj_type,
        referenced_name obj_name, path FROM fw_dep_objects)
      UNION
      (SELECT DISTINCT owner schema, type obj_type, name obj_name, path
       FROM bw_dep_objects)
    )
    WHERE schema IN (SELECT username FROM all_users WHERE oracle_maintained = 'N')
    ORDER BY obj_type;

 LOOP
    FETCH c BULK COLLECT INTO l_data LIMIT 100;
    BEGIN
      FORALL i IN 1..l_data.count SAVE EXCEPTIONS
        INSERT INTO DEPENDENT_ANALYSIS_BASELINE (
          owner, object_name, object_type, catagory, depedncy_level, project_need,
comments
        )
        VALUES (
          l_data(i).schema,
          l_data(i).obj_name,
```

```
          CASE WHEN l_data(i).obj_type = 'PACKAGE BODY' THEN 'PACKAGE' ELSE
 l_data(i).obj_type END,
          'level ' || l_level || ' dependency',
          l_level,
          '',
          'from dependency proc' || l_data(i).path
        );
    EXCEPTION
      WHEN OTHERS THEN
        l_errors := sql%bulk_exceptions.count;
        FOR i IN 1..l_errors LOOP
          l_errno := sql%bulk_exceptions(i).error_code;
          l_msg := SQLERRM(-l_errno);
          l_idx := sql%bulk_exceptions(i).error_index;
          UPDATE DEPENDENT_ANALYSIS_BASELINE
          SET catagory1 = catagory1 || ', found in level' || l_level || ' dependent of
 ' || l_data(l_idx).path,
              comments1 = comments1 || ', from dependency proc exception ' ||
 l_data(i).path
          WHERE owner = l_data(l_idx).schema
          AND object_name = l_data(l_idx).obj_name
          AND object_type = l_data(l_idx).obj_type;
        END LOOP;
    END;
    EXIT WHEN c%NOTFOUND;
  END LOOP;
  CLOSE c;
END;
```

Menilai kinerja kueri untuk memigrasi database SQL Server ke MongoDB Atlas di AWS

Battulga Purevragchaa, Amazon Web Services

Krishnakumar Sathyanarayana, AS Inc PeerIslands

Babu Srinivasan, MongoDB

Ringkasan

Pola ini memberikan panduan untuk memuat MongoDB dengan data dekat dunia nyata dan menilai kinerja kueri MongoDB yang sedekat mungkin dengan skenario produksi. Penilaian memberikan masukan untuk membantu Anda merencanakan migrasi ke MongoDB dari database relasional. Pola ini menggunakan PeerIslands Test Data Generator dan Performance Analyzer untuk menguji kinerja kueri.

Pola ini sangat berguna untuk migrasi Microsoft SQL Server ke MongoDB, karena melakukan transformasi skema dan memuat data dari instance SQL Server saat ini ke MongoDB bisa sangat kompleks. Sebagai gantinya, Anda dapat memuat data dekat dunia nyata ke MongoDB, memahami kinerja MongoDB, dan menyempurnakan desain skema sebelum memulai migrasi yang sebenarnya.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Keakraban dengan MongoDB Atlas
- Sasaran skema MongoDB
- Pola kueri yang khas

Batasan

- Waktu muat data dan kinerja akan dibatasi oleh ukuran instance cluster MongoDB. Kami menyarankan Anda memilih instans yang direkomendasikan untuk penggunaan produksi guna memahami kinerja dunia nyata.
- PeerIslands Test Data Generator dan Performance Analyzer saat ini hanya mendukung pemuatan dan kueri data online. Pemrosesan batch offline (misalnya, memuat data ke MongoDB dengan menggunakan konektor Spark) belum didukung.

- PeerIslands Test Data Generator dan Performance Analyzer mendukung hubungan lapangan dalam koleksi. Itu tidak mendukung hubungan di seluruh koleksi.

Edisi produk

- Pola ini mendukung MongoDB Atlas dan MongoDB Enterprise Advanced.

Arsitektur

Tumpukan teknologi target

- MongoDB Atlas atau MongoDB Enterprise Advanced

Arsitektur

PeerIslands Test Data Generator dan Performance Analyzer dibuat dengan menggunakan Java dan Angular, dan menyimpan data yang dihasilkan di Amazon Elastic Block Store (Amazon EBS). Alat ini terdiri dari dua alur kerja: pembuatan data uji dan pengujian kinerja.

- Dalam pembuatan data pengujian, Anda membuat template, yang merupakan representasi JSON dari model data yang harus dihasilkan. Setelah membuat template, Anda dapat menghasilkan data dalam kumpulan target, seperti yang ditentukan oleh konfigurasi pembuatan beban.
- Dalam pengujian kinerja, Anda membuat profil. Profil adalah skenario pengujian multi-tahap di mana Anda dapat mengonfigurasi operasi membuat, membaca, memperbarui, dan menghapus (CRUD), jalur agregasi, bobot untuk setiap operasi, dan durasi setiap tahap. Setelah Anda membuat profil, Anda dapat menjalankan pengujian kinerja pada database target berdasarkan konfigurasi.

PeerIslands Test Data Generator dan Performance Analyzer menyimpan datanya di Amazon EBS, sehingga Anda dapat menghubungkan Amazon EBS ke MongoDB dengan menggunakan mekanisme koneksi yang didukung MongoDB, termasuk peering, daftar izin, dan titik akhir pribadi. Secara default, alat ini tidak menyertakan komponen operasional; namun, alat ini dapat dikonfigurasi dengan Amazon Managed Service untuk Prometheus, Amazon Managed Grafana, Amazon, CloudWatch dan AWS Secrets Manager jika diperlukan.

Alat

- PeerIslands Test Data Generator dan Performance Analyzer mencakup dua komponen. Komponen Test Data Generator membantu Anda menghasilkan data dunia nyata yang sangat spesifik pelanggan berdasarkan skema MongoDB Anda. Alat ini sepenuhnya digerakkan oleh UI dengan pustaka data yang kaya dan dapat digunakan untuk menghasilkan miliaran catatan dengan cepat di MongoDB. Alat ini juga menyediakan kemampuan untuk mengimplementasikan hubungan antar bidang dalam skema MongoDB. Komponen Performance Analyzer membantu Anda menghasilkan kueri dan agregasi yang sangat spesifik pelanggan, dan melakukan pengujian kinerja yang realistis pada MongoDB. Anda dapat menggunakan Performance Analyzer untuk menguji kinerja MongoDB dengan profil beban kaya dan kueri berparameter untuk kasus penggunaan spesifik Anda.

Praktik terbaik

Lihat sumber daya berikut:

- Praktik Terbaik Desain Skema MongoDB (situs web Pengembang MongoDB)
- Praktik Terbaik Menerapkan MongoDB Atlas di AWS (situs web MongoDB)
- Menghubungkan Aplikasi dengan Aman ke Pesawat Data Atlas MongoDB dengan AWS (postingan blog PrivateLink AWS)
- Panduan Praktik Terbaik untuk Kinerja MongoDB (situs web MongoDB)

Epik

Memahami data sumber Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memahami jejak database dari sumber SQL Server saat ini.	Pahami jejak SQL Server Anda saat ini. Hal ini dapat dicapai dengan menjalankan query terhadap `INFORMATION_SCHEMA` database. Tentukan jumlah tabel dan ukuran setiap tabel. Analisis indeks yang terkait dengan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	setiap tabel. Untuk informasi lebih lanjut tentang analisis SQL, lihat posting blog SQL2Mongo: Perjalanan Migrasi Data di situs web. PeerIslands	
Memahami skema sumber.	Tentukan skema tabel dan representasi bisnis data (misalnya, kode pos, nama, dan mata uang). Gunakan diagram relasi entitas (ER) yang ada atau buat diagram ER dari database yang ada. Untuk informasi lebih lanjut, lihat posting blog SQL2Mongo : Perjalanan Migrasi Data di PeerIslands situs web.	DBA
Memahami pola kueri.	Dokumentasikan 10 kueri SQL teratas yang Anda gunakan. Anda dapat menggunakan tabel performance_schema.events_statements_summary_by_digest yang tersedia dalam database untuk memahami kueri teratas. Untuk informasi lebih lanjut, lihat posting blog SQL2Mongo : Perjalanan Migrasi Data di PeerIslands situs web.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memahami komitmen SLA.	Dokumentasikan perjanjian tingkat layanan target (SLAs) untuk operasi database. Ukuran umum termasuk latensi kueri dan kueri per detik. Langkah-langkah dan targetnya biasanya tersedia dalam dokumen persyaratan non-fungsional (NFR).	DBA

Tentukan skema MongoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan skema target.	Tentukan berbagai opsi untuk skema MongoDB target. Untuk informasi lebih lanjut, lihat Skema dalam dokumentasi MongoDB Atlas. Pertimbangkan praktik terbaik dan pola desain berdasarkan hubungan tabel.	Insinyur MongoDB
Tentukan pola kueri target.	Tentukan kueri MongoDB dan jalur agregasi. Kueri ini setara dengan kueri teratas yang Anda ambil untuk beban kerja SQL Server Anda. Untuk memahami cara membuat pipeline agregasi MongoDB, lihat dokumentasi MongoDB.	Insinyur MongoDB
Tentukan jenis instance MongoDB.	Tentukan ukuran instance yang Anda rencanakan untuk digunakan untuk	Insinyur MongoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengujian. Untuk panduan, lihat dokumentasi MongoDB.	

Siapkan database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan cluster MongoDB Atlas.	Untuk menyiapkan cluster MongoDB di AWS, ikuti petunjuk dalam dokumentasi MongoDB.	Insinyur MongoDB
Buat pengguna di database target.	Konfigurasikan cluster MongoDB Atlas untuk akses dan keamanan jaringan dengan mengikuti instruksi dalam dokumentasi MongoDB.	Insinyur MongoDB
Buat peran yang sesuai di AWS dan konfigurasikan kontrol akses berbasis peran untuk Atlas.	Jika diperlukan, siapkan pengguna tambahan dengan mengikuti petunjuk dalam dokumentasi MongoDB. Konfigurasikan otentikasi dan otorisasi melalui peran AWS.	Insinyur MongoDB
Siapkan Kompas untuk akses MongoDB Atlas.	Siapkan utilitas GUI MongoDB Compass untuk kemudahan navigasi dan akses.	Insinyur MongoDB

Mengatur beban dasar dengan menggunakan Test Data Generator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Generator Data Uji.	Instal PeerIsland Test Data Generator di lingkungan Anda.	Insinyur MongoDB
Konfigurasikan Test Data Generator untuk menghasilkan data yang sesuai.	Buat template dengan menggunakan pustaka data untuk menghasilkan data spesifik untuk setiap bidang dalam skema MongoDB. Untuk informasi lebih lanjut, lihat MongoDB Data Generator & Perf. Video penganalisis.	Insinyur MongoDB
Generator Data Uji skala horizontal untuk menghasilkan beban yang dibutuhkan.	Gunakan template yang Anda buat untuk memulai pembuatan beban terhadap koleksi target dengan mengonfigurasi paralelisme yang diperlukan. Tentukan kerangka waktu dan skala untuk menghasilkan data yang diperlukan.	Insinyur MongoDB
Validasi beban di MongoDB Atlas.	Periksa data yang dimuat ke MongoDB Atlas.	Insinyur MongoDB
Hasilkan indeks yang diperlukan di MongoDB.	Tentukan indeks sesuai kebutuhan, berdasarkan pola kueri. Untuk praktik terbaik, lihat dokumentasi MongoDB.	Insinyur MongoDB

Melakukan pengujian kinerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan profil beban di Performance Analyzer.	Buat profil pengujian kinerja di Performance Analyzer dengan mengonfigurasi kueri tertentu dan bobot yang sesuai, durasi uji coba, dan tahapannya. Untuk informasi lebih lanjut, lihat MongoDB Data Generator & Perf. Video penganalisis.	Insinyur MongoDB
Jalankan pengujian kinerja.	Gunakan profil pengujian kinerja yang Anda buat untuk memulai pengujian terhadap pengumpulan target dengan mengonfigurasi paralelisme yang diperlukan. Skala alat uji kinerja secara horizontal untuk menjalankan kueri terhadap MongoDB Atlas.	Insinyur MongoDB
Catat hasil tes.	Rekam latensi P95, P99 untuk kueri.	Insinyur MongoDB
Sesuaikan skema dan pola kueri Anda.	Ubah indeks dan pola kueri untuk mengatasi masalah kinerja apa pun.	Insinyur MongoDB

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.	Hapus semua sumber daya sementara yang Anda	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	gunakan untuk Test Data Generator dan Performance Analyzer.	
Perbarui hasil tes kinerja.	Pahami kinerja kueri MongoDB dan bandingkan dengan Anda. SLAs Jika perlu, sesuaikan skema MongoDB dan jalankan kembali prosesnya.	Insinyur MongoDB
Akhiri proyek.	Tutup proyek dan berikan umpan balik.	Insinyur MongoDB

Sumber daya terkait

- GitHub repositori: S3toAtlas
- Skema: Desain Skema MongoDB
- Pipa agregasi: Pipa agregasi MongoDB
- Ukuran MongoDB Atlas: Pemilihan tingkat ukuran
- Video: Generator & Kinerja Data MongoDB. Analyzer
- Referensi: Dokumentasi MongoDB
- Tutorial: Panduan pengembang MongoDB, MongoDB Jumpstart
- AWS Marketplace: Atlas MongoDB di AWS Marketplace
- Solusi Mitra AWS: Atlas MongoDB pada Penerapan Referensi AWS

Sumber daya tambahan:

- Analisis SQL
- Forum Komunitas Pengembang MongoDB
- Pertanyaan Penyetelan Kinerja MongoDB
- Analisis Operasional dengan Atlas dan Redshift

- Modernisasi aplikasi dengan MongoDB Atlas dan AWS Elastic Beanstalk

Mengotomatiskan blue/green penyebaran database global Amazon Aurora dengan menggunakan prinsip IAc

Ishwar Chathaiwale, ANKIT JAIN, dan Ramu Jagini, Amazon Web Services

Ringkasan

Mengelola pembaruan database, migrasi, atau upaya penskalaan dapat menjadi tantangan bagi organisasi yang menjalankan beban kerja penting di database global Amazon Aurora. Memastikan bahwa operasi ini dilakukan dengan mulus, tanpa downtime, sangat penting untuk menjaga ketersediaan layanan dan menghindari gangguan bagi pengguna Anda.

Strategi blue/green penyebaran menawarkan solusi untuk tantangan ini dengan memungkinkan Anda menjalankan dua lingkungan yang identik secara bersamaan: biru (lingkungan saat ini) dan hijau (lingkungan baru). Sebuah blue/green strategi memungkinkan Anda untuk menerapkan perubahan, melakukan pengujian, dan mengalihkan lalu lintas antar lingkungan dengan risiko minimal dan downtime.

Pola ini membantu Anda mengotomatiskan proses blue/green penyebaran untuk database global Aurora dengan menggunakan prinsip infrastruktur sebagai kode (IAc). Ini menggunakan AWS CloudFormation, AWS Lambda, dan Amazon Route 53 untuk menyederhanakan blue/green penerapan. Untuk meningkatkan keandalan, ia menggunakan pengidentifikasi transaksi global (GTIDs) untuk replikasi. Replikasi berbasis GTID memberikan konsistensi data dan kemampuan failover yang lebih baik antar lingkungan dibandingkan dengan replikasi log biner (binlog).

> **Note**
>
> Pola ini mengasumsikan bahwa Anda menggunakan cluster database global Aurora MySQL yang kompatibel dengan Edition. Jika Anda menggunakan Aurora PostgreSQL kompatibel sebagai gantinya, silakan gunakan PostgreSQL setara dengan perintah MySQL.

Dengan mengikuti langkah-langkah dalam pola ini, Anda dapat:

- Menyediakan database global Aurora hijau: Menggunakan CloudFormation template, Anda membuat lingkungan hijau yang mencerminkan lingkungan biru Anda yang ada.
- Siapkan replikasi berbasis GTID: Anda mengonfigurasi replikasi GTID agar lingkungan biru dan hijau tetap disinkronkan.

- Beralih lalu lintas dengan mulus: Anda menggunakan Route 53 dan Lambda untuk secara otomatis mengalihkan lalu lintas dari biru ke lingkungan hijau setelah sinkronisasi penuh.
- Selesaikan penerapan: Anda memvalidasi bahwa lingkungan hijau beroperasi penuh sebagai basis data utama, dan kemudian menghentikan replikasi dan membersihkan sumber daya sementara.

Pendekatan dalam pola ini memberikan manfaat ini:

- Mengurangi waktu henti selama pembaruan atau migrasi basis data penting: Otomatisasi memastikan transisi yang mulus antar lingkungan dengan gangguan layanan minimal.
- Mengaktifkan rollback cepat: Jika masalah muncul setelah lalu lintas beralih ke lingkungan hijau, Anda dapat dengan cepat kembali ke lingkungan biru dan mempertahankan kontinuitas layanan.
- Meningkatkan pengujian dan verifikasi: Lingkungan hijau dapat sepenuhnya diuji tanpa mempengaruhi lingkungan biru hidup, yang mengurangi kemungkinan kesalahan dalam produksi.
- Memastikan konsistensi data: Replikasi berbasis GTID membuat lingkungan biru dan hijau Anda tetap sinkron, yang mencegah kehilangan data atau ketidakkonsistenan selama migrasi.
- Menjaga kelangsungan bisnis: Mengotomatiskan blue/green penerapan Anda membantu menghindari pemadaman yang lama dan kerugian finansial dengan menjaga layanan Anda tetap tersedia selama pembaruan atau migrasi.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Sumber cluster database global yang kompatibel dengan Aurora MySQL (lingkungan biru). Database global menyediakan konfigurasi Multi-wilayah untuk ketersediaan tinggi dan pemulihan bencana. Untuk petunjuk untuk menyiapkan kluster database global, lihat dokumentasi Aurora.
- Replikasi berbasis GTID diaktifkan pada cluster sumber.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

- Aurora MySQL kompatibel 8.0 atau yang lebih baru

Arsitektur

Diagram ini menggambarkan hal sebagai berikut:

- Pengaturan basis data global: Kluster basis data global Aurora digunakan secara strategis di dua. Wilayah AWS Konfigurasi ini memungkinkan distribusi geografis dan redundansi Regional untuk meningkatkan kemampuan pemulihan bencana.
- Replikasi Wilayah primer ke sekunder: Mekanisme replikasi logis memastikan sinkronisasi data yang mulus dari Wilayah primer ke Wilayah sekunder. Replikasi ini mempertahankan konsistensi data dengan latensi minimal di seluruh jarak geografis.
- Replikasi berbasis GTID antar cluster: Replikasi berbasis GTID mempertahankan konsistensi transaksional dan aliran data yang teratur antara cluster primer biru dan cluster primer hijau, dan memastikan sinkronisasi data yang andal.
- Replikasi primer ke sekunder biru: Replikasi logis membentuk jalur data yang kuat antara cluster primer biru dan cluster sekundernya. Replikasi ini memungkinkan sinkronisasi data berkelanjutan dan ketersediaan tinggi.
- Konfigurasi DNS Route 53: Route 53 catatan zona yang dihosting mengelola resolusi DNS untuk semua titik akhir database cluster biru dan hijau. Konfigurasi ini menyediakan pemetaan titik akhir yang mulus dan memungkinkan perutean lalu lintas yang efisien selama skenario failover.

Alat

Layanan AWS

- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.
- CloudFormationmembantu Anda memodelkan dan mengatur AWS sumber daya sehingga Anda dapat menghabiskan lebih sedikit waktu untuk mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi yang berjalan AWS. Anda membuat template yang menjelaskan semua AWS sumber daya yang Anda inginkan, dan CloudFormation mengurus penyediaan dan konfigurasi sumber daya tersebut untuk Anda.

- AWS Lambdaadalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.

Praktik terbaik

Kami menyarankan Anda meninjau AWS dokumentasi secara menyeluruh untuk memperdalam pemahaman Anda tentang strategi penerapan biru/hijau, replikasi berbasis GTID, dan kebijakan perutean tertimbang di Route 53. Pengetahuan ini sangat penting untuk menerapkan dan mengelola migrasi database Anda secara efektif, memastikan konsistensi data, dan mengoptimalkan perutean lalu lintas. Dengan mendapatkan pemahaman yang komprehensif tentang AWS fitur dan praktik terbaik ini, Anda akan lebih siap untuk menangani pembaruan masa depan, meminimalkan waktu henti, dan mempertahankan lingkungan database yang tangguh dan aman.

Untuk pedoman penggunaan pola Layanan AWS untuk ini, lihat AWS dokumentasi berikut:

- Praktik terbaik dengan Amazon Aurora MySQL
- CloudFormation praktik terbaik
- Praktik terbaik untuk bekerja dengan AWS Lambda fungsi
- Praktik terbaik untuk Amazon Route 53

Epik

Ciptakan lingkungan hijau

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cadangan snapshot dari cluster biru.	Dalam blue/green penerapan , lingkungan hijau mewakili versi baru yang identik dari lingkungan database (biru) Anda saat ini. Anda menggunakan lingkunga n hijau untuk menguji pembaruan dengan aman,	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memvalidasi perubahan, dan memastikan stabilitas sebelum mengalihkan lalu lintas produksi. Ini bertindak sebagai landasan pementasan untuk menerapkan perubahan database dengan gangguan minimal terhadap lingkungan hidup. Untuk membuat lingkunga n hijau, pertama-tama Anda membuat snapshot dari cluster utama (biru) di database global yang kompatibel dengan Aurora MySQL Anda. Snapshot ini berfungsi sebagai dasar untuk menciptakan lingkungan hijau. Untuk membuat snapshot: 1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Relational Database Service (Amazon RDS). 2. Pilih klaster utama (biru) Anda. 3. Pilih Tindakan, Ambil snapshot. 4. Berikan nama untuk snapshot, seperti`blue-green-demo` , dan mulai proses pencadangan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Atau, Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk membuat snapshot: `aws rds create-db-cluster-snapshot --db-cluster-snapshot-identifier blue-green-demo --db-cluster-identifier ex-global-cluster --region eu-west-1` Pastikan snapshot selesai dengan sukses sebelum melanjutkan ke langkah berikutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan CloudFormation template untuk database global Anda dan sumber dayanya.	Generator CloudFormation IAc membantu Anda menghasil kan CloudFormation template dari AWS sumber daya yang ada. Gunakan fitur ini untuk membuat CloudFormation template untuk database global Aurora MySQL yang kompatibel dengan Aurora yang ada dan sumber daya yang terkait. Template ini mengonfigurasi grup subnet, grup keamanan, grup parameter, dan pengaturan lainnya. 1. Ikuti petunjuk dalam CloudFormation dokumenta si untuk menavigasi ke alat dan menghubungkannya ke AWS lingkungan Anda. 2. Pilih database global Aurora Anda dan sumber daya terkait untuk menghasilkan template.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah CloudFormation template untuk lingkungan hijau.	Sesuaikan CloudFormation template untuk mencermin kan pengaturan untuk lingkungan hijau. Ini termasuk memperbarui nama sumber daya dan pengidentifikasi untuk memastikan bahwa lingkungan hijau beroperasi secara independen dari cluster biru. 1. Perbarui `DBCluster Identifier` dan `DBInstanceIdentifi er` properti untuk mewakili lingkungan hijau. 2. Ubah nama sumber daya lain (misalnya, grup subnet dan grup keamanan) untuk menghindari konflik dengan lingkungan biru yang ada. 3. Aktifkan replikasi berbasis GTID dalam template dengan mengonfigurasi parameter yang benar, seperti yang dijelaskan dalam dokumentasi Aurora. 4. Ubah `SnapshotI dentifier` properti untuk menentukan Wilayah AWS, ID akun Anda, dan nama snapshot dari langkah sebelumnya:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`SnapshotIdentifier: arn:aws:rds:<region>:<account-id>:snapshot:<snapshot-name>` **Note** Jika Anda menggunakan `SnapshotIdentifier` properti untuk memulihkan cluster DB, hindari menentukan properti seperti `GlobalClusterIdentifier`, `MasterUsername`, atau `MasterUserPassword`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan CloudFormation tumpukan untuk membuat sumber daya untuk lingkungan hijau.	Pada langkah ini, Anda menerapkan CloudFormation template yang disesuaikan untuk membuat sumber daya untuk lingkungan hijau. Untuk menyebarkan CloudFormation tumpukan: 1. Buka konsol AWS CloudFormation. 2. Di kanan atas, pilih Buat tumpukan, Dengan sumber daya baru (standar). 3. Unggah CloudForm ation templat Anda yang dimodifikasi atau tentukan URL templat. Pilih Berikutny a. 4. Masukkan nama tumpukan sepertiGreenClus terStack , dan berikan parameter yang diperluka n (misalnya,GreenClus terIdentifier). Pilih Berikutnya. 5. Konfigurasikan opsi tumpukan tambahan sesuai kebutuhan, dan centang kotak untuk mengetahu i yang CloudFormation mungkin membuat sumber daya AWS Identity and	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Access Management (IAM). Pilih Berikutnya. 6. Tinjau detail tumpukan. 7. Pilih Kirim. CloudFormation memulai proses menciptakan sumber daya lingkungan hijau. Proses ini mungkin memakan waktu beberapa menit untuk menyelesaikannya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi CloudFormation tumpukan dan sumber daya.	Saat penerapan CloudFormation tumpukan selesai, Anda harus memverifikasi bahwa lingkungan hijau telah berhasil dibuat: 1. Di bagian Output dari CloudFormation tumpukan, periksa titik akhir cluster database dan instance database untuk memverifikasi pengaturan yang benar. 2. Buka konsol Amazon RDS dan konfirmasikan bahwa cluster database Aurora baru (lingkungan hijau) tersedia. 3. Pastikan bahwa semua sumber daya terkait, seperti subnet dan grup keamanan, telah dibuat dan ditautkan ke lingkungan hijau. Setelah verifikasi, lingkungan hijau Anda siap untuk pengaturan lebih lanjut, termasuk replikasi dari lingkungan biru.	DBA

Konfigurasikan replikasi berbasis GTID

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi pengaturan GTID pada cluster biru.	GTIDs menyediakan metode yang sangat andal untuk mereplikasi data antara lingkungan biru dan hijau Anda. Replikasi berbasis GTID menawarkan pendekatan yang tangguh dan disederhanakan dengan menetapkan pengenal unik untuk setiap transaksi di lingkungan biru. Metode ini memastikan bahwa sinkronisasi data antar lingkungan mulus, konsisten, dan lebih mudah dikelola daripada replikasi binlog tradisional. Sebelum Anda mengonfigurasi replikasi, Anda perlu memastikan bahwa replikasi berbasis GTID diaktifkan dengan benar di cluster biru. Langkah ini menjamin bahwa setiap transaksi di lingkungan biru dilacak secara unik dan dapat direplikasi di lingkungan hijau. Untuk mengonfirmasi bahwa GTID diaktifkan: 1. Di konsol Amazon RDS, tinjau grup parameter yang ditetapkan ke cluster biru.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Verifikasi bahwa parameter berikut diatur: • `gtid-mode = ON` • `enforce_gtid_consistency = ON` Pengaturan ini memungkinkan pelacakan GTID untuk semua transaksi future di lingkungan biru. Setelah Anda mengonfirmasi pengaturan ini, Anda dapat mulai mengatur replikasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna replikasi.	Untuk mereplikasi data dari lingkungan biru ke lingkunga n hijau, Anda perlu membuat pengguna replikasi khusus di cluster biru. Pengguna ini akan bertanggung jawab untuk mengelola proses replikasi. Untuk mengatur pengguna replikasi: 1. Connect ke cluster biru dengan menggunakan klien MySQL. 2. Jalankan perintah berikut untuk membuat pengguna replikasi: `CREATE USER 'repl_use r'@'%' IDENTIFIED BY 'repl_password'; GRANT REPLICATI ON SLAVE ON . TO 'repl_user'@'%'; FLUSH PRIVILEGES;` Pengguna ini sekarang memiliki izin yang diperlukan untuk mereplikasi data antara dua lingkungan.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan replikasi berbasis GTID pada cluster hijau.	Langkah selanjutnya adalah mengkonfigurasi cluster hijau untuk replikasi berbasis GTID. Pengaturan ini memastikan bahwa lingkungan hijau akan terus mencerminkan semua transaksi yang terjadi di lingkungan biru. Untuk mengkonfigurasi cluster hijau: 1. Connect ke cluster hijau dengan menggunakan klien MySQL. 2. Jalankan perintah berikut untuk mengkonfigurasi replikasi: `CHANGE MASTER TO MASTER_HOST='blue-cluster-endpoint', MASTER_USER='repl_user', MASTER_PASSWORD='repl_password', MASTER_AUTO_POSITION=1;` di mana: • Ganti `blue-cluster-endpoint` dengan titik akhir cluster biru Anda. • `MASTER_AUTO_POSITION=1` Pengaturan menginstruksikan MySQL untuk menggunak	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an replikasi berbasis GTID. Secara otomatis memposisikan cluster hijau untuk mereplika si transaksi cluster biru tanpa harus melacak log dan posisi secara manual.	
Mulai replikasi pada cluster hijau.	Anda sekarang dapat memulai proses replikasi. Pada cluster hijau, jalankan perintah: `START SLAVE;` Hal ini memungkinkan lingkungan hijau untuk mulai menyinkronkan data, dan menerima dan menerapkan transaksi dari lingkungan biru.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi proses replikasi.	Untuk memverifikasi bahwa lingkungan hijau secara akurat mereplikasi data dari cluster biru: 1. Jalankan perintah berikut pada cluster hijau untuk memeriksa status replikasi: `SHOW SLAVE STATUS\G;` 2. Tinjau output untuk memverifikasi hal berikut: • `Slave_IO_Running = Yes` • `Slave_SQL_Running = Yes` • `Executed_Gtid_Set` Nilai `Retrieved_Gtid_Set` dan disinkronkan dengan cluster biru. up-to-date • Tidak ada kesalahan replikasi di `Last_Error` lapangan. Jika semua indikator benar, replikasi berbasis GTID berfungsi dengan lancar, dan lingkungan hijau sepenuhnya disinkronkan dengan lingkungan biru.	DBA

Beralih lalu lintas dari klaster biru ke hijau

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kebijakan perutean tertimbang Route 53.	Setelah Anda memverifikasi konsistensi data antara lingkungan biru dan hijau, Anda dapat mengalihkan lalu lintas dari cluster biru ke cluster hijau. Transisi ini harus lancar dan harus meminimalkan downtime dan memastikan integritas database aplikasi Anda. Untuk mengatasi persyaratan ini, Anda dapat menggunakan Route 53 untuk perutean DNS dan Lambda untuk mengotomatiskan peralihan lalu lintas. Selain itu, rencana rollback yang terdefinisi dengan baik memastikan bahwa Anda dapat kembali ke cluster biru jika terjadi masalah. Langkah pertama adalah mengkonfigurasi routing tertimbang di Route 53. Perutean tertimbang memungkinkan Anda untuk mengontrol distribusi lalu lintas antara cluster biru dan hijau, dan secara bertahap mengalihkan lalu lintas dari satu lingkungan ke lingkungan lainnya.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk mengonfigurasi perutean tertimbang: 1. Buka konsol Route 53 dan pilih zona host Anda. 2. Buat dua catatan DNS (CNAMEs) untuk database: satu catatan untuk cluster biru dan satu catatan untuk cluster hijau. 3. Tetapkan bobot awal: • Tetapkan bobot awal yang rendah (seperti 5 persen) untuk cluster hijau untuk mengirim sebagian kecil lalu lintas untuk pengujian. • Tetapkan bobot yang lebih tinggi (seperti 95 persen) untuk cluster biru, sehingga mempertahankan sebagian besar lalu lintas. Konfigurasi ini memungkinkan Anda untuk melakukan transisi bertahap yang mengurangi risiko dan mendukung pengujian real-time sebelum Anda beralih sepenuhnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang kebijakan perutean tertimbang, lihat dokumentasi Route 53.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menerapkan fungsi Lambda untuk memantau kelambatan replikasi.	Untuk memastikan bahwa lingkungan hijau sepenuhnya disinkronkan dengan lingkungan biru, gunakan fungsi Lambda yang memantau jeda replikasi antar cluster. Fungsi ini dapat memeriksa status replikasi, khususnya metrik Seconds_Behind_Master, untuk menentukan apakah cluster hijau siap menangani semua lalu lintas.
	
	Berikut contoh fungsi Lambda yang dapat Anda gunakan:
	
	`import boto3`
	
	`def check_replication_lag(event, context):`
	`client = boto3.client('rds')`
	`response = client.describe_db_instances(DBInstanceIdentifier='green-cluster-instance')`
	`replication_status = response['DBInstances'][0]['ReadReplicaDBInstanceIdentifiers']`
	`if replication_status:`
	`lag = replication_status[0]['ReplicationLag']`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre> return lag return -1</pre> Fungsi ini memeriksa lag replikasi dan mengembalikan nilai. Jika lag nol, cluster hijau sepenuhnya sinkron dengan cluster biru.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Otomatiskan penyesuaian berat DNS dengan menggunakan Lambda.	Ketika lag replikasi mencapai nol, saatnya untuk mengalihkan semua lalu lintas ke cluster hijau. Anda dapat mengotomatiskan transisi ini dengan menggunakan fungsi Lambda lain yang menyesuaikan bobot DNS di Route 53 untuk mengarahkan 100 persen lalu lintas ke cluster hijau.
	
	Berikut adalah contoh fungsi Lambda yang mengotomatiskan sakelar lalu lintas:
	
	`import boto3`
	
	`def switch_traffic(event, context):`
	`route53 = boto3.client('route53')`
	`lag = check_replication_lag(event, context)`
	`if lag == 0:`
	`response = route53.change_resource_record_sets(`
	`HostedZoneId='YOUR_HOSTED_ZONE_ID',`
	`ChangeBatch={`
	`'Changes': [`
	`{`
	`'Action': 'UPSERT',`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>'ResourceRecordSet':
	{
	'Name': 'db.example.com',
	'Type': 'CNAME',
	'SetIdentifier':
	'GreenCluster',
	'Weight': 100,
	'TTL': 60,
	'ResourceRecords':
	[{'Value': 'green-cluster-endpoint'}]
	}
	},
	{
	'Action': 'UPSERT',
	'ResourceRecordSet':
	{
	'Name': 'db.example.com',
	'Type': 'CNAME',
	'SetIdentifier':
	'BlueCluster',
	'Weight': 0,
	'TTL': 60,
	'ResourceRecords':</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>[{'Value': 'blue-clu
	ster-endpoint'}]
	
	}
	}
	]
	}
	)
	return response</pre>
	Fungsi ini memeriksa lag replikasi dan memperbarui bobot DNS Route 53 saat jeda nol untuk sepenuhny a mengalihkan lalu lintas ke cluster hijau.
	
	Note
	Selama proses cutover, Jika cluster biru mengalami lalu lintas tulis yang padat, pertimbangkan untuk menghentikan sementara operasi penulisan selama cutover. Ini memastika n bahwa replikasi menyusul, dan mencegah inkonsist ensi data antara cluster biru dan hijau.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi sakelar lalu lintas.	Setelah fungsi Lambda menyesuaikan bobot DNS, Anda harus memverifikasi bahwa semua lalu lintas diarahkan ke cluster hijau dan bahwa sakelar berhasil. Untuk memverifikasi: 1. Pantau catatan DNS Route 53 untuk mengonfirmasi bahwa lalu lintas sedang diarahkan ke cluster hijau. Untuk informasi selengkap nya, lihat dokumentasi Route 53. 2. Periksa kinerja aplikasi dengan mengonfirmasi bahwa pengguna dilayani dari lingkungan hijau. 3. Verifikasi koneksi database untuk mengonfirmasi bahwa klaster hijau menangani semua permintaan database. 4. Pantau CloudWatch metrik Amazon untuk mengetahu i tanda-tanda latensi, kelambatan replikasi, atau penurunan kinerja. Untuk informasi lebih lanjut, lihat dokumentasi Aurora.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jika semuanya berjalan seperti yang diharapkan, sakelar lalu lintas selesai.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jika Anda mengalami masalah, putar kembali perubahan.	Memiliki rencana rollback sangat penting jika ada masalah yang muncul setelah sakelar lalu lintas. Berikut cara cepat kembali ke cluster biru jika perlu: 1. Kembalikan bobot DNS di Route 53: Gunakan fungsi Lambda yang sama atau sesuaikan bobot DNS Route 53 secara manual untuk mengarahkan 100 persen lalu lintas kembali ke cluster biru. 2. Pantau kinerja aplikasi: Segera pantau log aplikasi, CloudWatch metrik, dan kinerja database untuk mengonfirmasi bahwa peralihan kembali ke lingkungan biru telah menyelesaikan masalah. 3. Identifikasi dan selesaikan Masalah: Selidiki dan atasi masalah apa pun dengan klaster hijau sebelum Anda mencoba sakelar lalu lintas lain. Dengan menerapkan rencana rollback ini, Anda dapat memastikan gangguan minimal pada pengguna Anda	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	jika terjadi masalah yang tidak terduga.	

Validasi dan hentikan replikasi berbasis GTID

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan replikasi berbasis GTID pada cluster hijau.	Setelah Anda mengalihkan lalu lintas dari lingkungan biru ke lingkungan hijau, Anda harus memvalidasi keberhasi lan transisi dan memastikan bahwa cluster hijau berfungsi seperti yang diharapkan. Selain itu, replikasi berbasis GTID antara cluster biru dan hijau harus dihentikan, karena lingkungan hijau sekarang berfungsi sebagai database utama. Menyelesaikan tugas-tugas ini memastikan bahwa lingkungan Anda aman, efisien, dan dioptimalkan untuk operasi yang sedang berlangsung. Untuk menghentikan replikasi: 1. Gunakan klien MySQL untuk terhubung ke cluster hijau. 2. Jalankan perintah SQL berikut untuk menghenti kan proses replikasi pada cluster hijau:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`STOP SLAVE;`
	3. (Opsional) Jika diinginkan, Anda dapat mengatur ulang konfigurasi replikasi untuk menghapus pengaturan replikasi sisa:
	`RESET SLAVE ALL;`
	Ketika Anda menghentikan replikasi, cluster hijau menjadi sepenuhnya independen dan beroperasi sebagai lingkungan database utama untuk beban kerja Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Membersihkan sumber daya sementara atau tidak terpakai yang dibuat selama migrasi dari klaster biru ke hijau memastikan bahwa lingkungan Anda tetap dioptimalkan, aman, dan hemat biaya. Pembersih an termasuk menyesuai kan pengaturan keamanan, mengambil cadangan akhir, dan menonaktifkan sumber daya yang tidak perlu. Untuk membersihkan sumber daya: 1. Perbarui grup keamanan: Konfigurasikan grup keamanan yang terkait dengan cluster biru dan hijau untuk mencermin kan lingkungan primer baru (hijau). Batasi akses ke lingkungan biru jika tidak lagi diperlukan, dan verifikasi bahwa pengatura n keamanan klaster hijau mengikuti praktik terbaik. 2. Buat cadangan akhir cluster hijau: Setelah migrasi selesai, ambil snapshot akhir dari cluster hijau untuk dijadikan cadangan. Anda dapat menggunak	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	an snapshot ini untuk memulihkan lingkungan di masa depan jika perlu.
	
	`aws rds create-db-snapshot --db-instance-identifier green-cluster-instance --db-snapshot-identifier green-cluster-final-snapshot`
	
	3. Tinjau dan hapus sumber daya sementara: Tinjau sumber daya sementara yang dibuat selama migrasi, seperti grup keamanan sementara, snapshot, atau konfigurasi lainnya. Hapus sumber daya yang tidak lagi diperlukan untuk mencegah biaya yang tidak perlu. Misalnya, hapus cluster biru jika tidak lagi diperlukan:
	
	`aws rds delete-db-cluster --db-cluster-identifier blue-cluster-identifier --skip-final-snapshot`
	
	Membersihkan sumber daya membantu menjaga lingkungan yang aman dan efisien, mengurangi biaya, dan memastikan bahwa hanya	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	infrastruktur yang diperlukan yang tersisa.	

Sumber daya terkait

CloudFormation:

- CloudFormation Panduan Pengguna
- CloudFormation praktik terbaik
- Hasilkan templat dari sumber daya yang ada dengan IAc Generator
- Impor seluruh aplikasi ke CloudFormation (posting AWS blog)

Amazon Aurora:

- Panduan Pengguna Amazon Aurora
- Mengelola cluster DB Amazon Aurora

Strategi penyebaran biru/hijau:

- Ikhtisar Penerapan Amazon Aurora Blue/Green

Replikasi berbasis GTID:

- Menggunakan replikasi berbasis GTID (dokumentasi Amazon RDS)

AWS Lambda:

- AWS Lambda Panduan Pengembang
- Praktik terbaik untuk bekerja dengan AWS Lambda fungsi

Rute Amazon 53:

- Panduan Pengembang Amazon Route 53
- Perutean tertimbang

Alat klien MySQL:

- PyMySQL

Otomatiskan replikasi instans Amazon RDS di seluruh Akun AWS

Parag Nagwekar dan Arun Chandapillai, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara mengotomatiskan proses mereplikasi, melacak, dan memutar kembali instans DB Amazon Relational Database Service (Amazon RDS) Anda di berbagai tempat dengan menggunakan dan. Akun AWS AWS Step Functions AWS Lambda Anda dapat menggunakan otomatisasi ini untuk melakukan replikasi skala besar instans RDS DB tanpa dampak kinerja atau overhead operasional—terlepas dari ukuran organisasi Anda. Anda juga dapat menggunakan pola ini untuk membantu organisasi Anda mematuhi strategi tata kelola data wajib atau persyaratan kepatuhan yang meminta agar data Anda direplikasi dan berlebihan di berbagai bidang. Akun AWS Wilayah AWS Replikasi lintas akun data Amazon RDS dalam skala besar adalah proses manual yang tidak efisien dan rawan kesalahan yang dapat memakan biaya dan memakan waktu, tetapi otomatisasi dalam pola ini dapat membantu Anda mencapai replikasi lintas akun dengan aman, efektif, dan efisien.

Prasyarat dan batasan

Prasyarat

- Dua Akun AWS
- Instans RDS DB, aktif dan berjalan di sumbernya Akun AWS
- Grup subnet untuk instans RDS DB di tujuan Akun AWS
- Kunci AWS Key Management Service (AWS KMS) yang dibuat di sumber Akun AWS dan dibagikan dengan akun tujuan (Untuk informasi selengkapnya tentang detail kebijakan, lihat bagian Informasi tambahan dari pola ini.)
- Sebuah AWS KMS key di tujuan Akun AWS untuk mengenkripsi database di akun tujuan

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

- Python 3.9 (menggunakan) AWS Lambda
- PostgreSQL 11.3, 13.x, dan 14.x

Arsitektur

Tumpukan teknologi

- Amazon Relational Database Service (Amazon RDS)
- Amazon Simple Notification Service (Amazon SNS)
- AWS Key Management Service (AWS KMS)
- AWS Lambda
- AWS Secrets Manager
- AWS Step Functions

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk menggunakan Step Functions untuk mengatur replikasi instans RDS DB yang dijadwalkan dan sesuai permintaan dari akun sumber (akun A) ke akun tujuan (akun B).

Di akun sumber (akun A dalam diagram), mesin status Step Functions melakukan hal berikut:

1. Membuat snapshot dari instans RDS DB di akun A.
2. Menyalin dan mengenkripsi snapshot dengan akun AWS KMS key A. Untuk memastikan enkripsi dalam perjalanan, snapshot dienkripsi apakah instans DB dienkripsi atau tidak.
3. Berbagi snapshot DB dengan akun B dengan memberikan akses akun B ke snapshot.
4. Mendorong pemberitahuan ke topik SNS, dan kemudian topik SNS memanggil fungsi Lambda di akun B.

Di akun tujuan (akun B dalam diagram), fungsi Lambda menjalankan mesin status Step Functions untuk mengatur hal-hal berikut:

1. Menyalin snapshot bersama dari akun A ke akun B, saat menggunakan AWS KMS key dari akun A untuk mendekripsi data terlebih dahulu dan kemudian mengenkripsi data dengan menggunakan akun dalam B. AWS KMS key

2. Membaca rahasia dari Secrets Manager untuk menangkap nama instans DB saat ini.
3. Mengembalikan instans DB dari snapshot dengan nama baru dan default untuk AWS KMS key Amazon RDS.
4. Membaca endpoint dari database baru dan memperbarui rahasia di Secrets Manager dengan endpoint database baru, dan kemudian tag instance DB sebelumnya sehingga dapat dihapus nanti.
5. Menyimpan instance N terbaru dari database dan menghapus semua instance lainnya.

Alat

Layanan AWS

- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi Lambda dan lainnya untuk membangun aplikasi bisnis yang penting. Layanan AWS

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Crossaccount RDS Replication.

Epik

Otomatiskan replikasi instans RDS DB dengan satu klik Akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation tumpukan di akun sumber.	1. Masuk ke Konsol Manajemen AWS akun sumber (akun A) dan buka CloudFormation konsol. 2. Di panel navigasi, pilih Stacks (Tumpukan). 3. Pilih Buat tumpukan, lalu pilih Dengan sumber daya yang ada (sumber daya impor). 4. Pada halaman Identifik asi sumber daya, pilih Berikutnya. 5. Pada halaman Tentukan templat, pilih Unggah templat. 6. Pilih Pilih file, pilih **`Cloudformation-SourceAccountRDS.yaml`** file dari repositori RDS Replikasi GitHub Crossacco unt, dan kemudian pilih Berikutnya. 7. Untuk nama Stack, masukkan nama untuk tumpukan Anda. 8. Di bagian Parameter, tentukan parameter yang	Administrator cloud, Arsitek cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ditentukan dalam template tumpukan: • Untuk DestinationAccount Number, masukkan nomor akun untuk instans RDS DB tujuan Anda. • Untuk KeyName, masukkan Anda AWS KMS key. • Untuk ScheduleE xpression, masukkan ekspresi cron (defaultnya adalah 12:00 pagi setiap hari). • Untuk Sumber DBIdentif ier, masukkan nama database sumber. • Untuk DBSnapshotNama Sumber, masukkan nama snapshot atau terima default. 9. Pilih Berikutnya. 10. Pada halaman Configure stack options, tinggalkan nilai default, lalu pilih Next. 11. Tinjau konfigurasi tumpukan Anda, lalu pilih Kirim. 12. Pilih tab Sumber Daya untuk tumpukan Anda, lalu catat Nama Sumber Daya Amazon (ARN) dari topik SNS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation tumpukan di akun tujuan.	1. Masuk ke Konsol Manajemen AWS akun tujuan (akun B) dan buka CloudFormation konsol. 2. Di panel navigasi, pilih Stacks (Tumpukan). 3. Pilih Buat tumpukan, lalu pilih Dengan sumber daya yang ada (sumber daya impor). 4. Pada halaman Identifik asi sumber daya, pilih Berikutnya. 5. Pada halaman Tentukan templat, pilih Unggah templat. 6. Pilih file, pilih **`Cloudform ation-DestinationA ccountRDS.yaml`** file dari repositori GitHub Crossaccount RDS Replication, dan kemudian pilih Next. 7. Untuk nama Stack, masukkan nama untuk tumpukan Anda. 8. Di bagian Parameter, tentukan parameter yang ditentukan dalam template tumpukan: • Untuk DatabaseName, masukkan nama untuk database Anda.	Arsitek cloud, DevOps insinyur, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk Engine, masukkan jenis mesin database yang cocok dengan basis data sumber. • Untuk DBInstanceKelas, masukkan jenis instance database pilihan atau terima default. • Untuk Subnetgro ups, masukkan grup subnet VPC yang ada. Untuk petunjuk tentang membuat grup subnet, lihat Langkah 2: Membuat grup subnet DB dalam dokumentasi Amazon RDS. • Untuk SecretName, masukkan jalur dan nama rahasia, atau terima default. • Untuk SGID, masukkan ID grup keamanan klaster tujuan Anda. • Untuk KMSKey, masukkan ARN kunci KMS di akun tujuan Anda. • Untuk NoOfOlder Instances, masukkan jumlah salinan lama dari instans RDS DB yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ingin Anda simpan untuk rollback. 9. Pilih Berikutnya. 10. Pada halaman Configure stack options, tinggalkan nilai default, lalu pilih Next. 11. Tinjau konfigurasi tumpukan Anda, lalu pilih Kirim. 12. Pilih tab Sumber Daya untuk tumpukan Anda, lalu catat ID Fisik dan ARN dari. `InvokeStepFunction`	
Verifikasi pembuatan instans RDS DB di akun tujuan.	1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon RDS. 2. Di panel navigasi, pilih Databases, lalu verifikasi bahwa instans RDS DB baru muncul di bawah cluster baru.	Administrator cloud, arsitek Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Berlangganan fungsi Lambda ke topik SNS.	Anda harus menjalankan perintah AWS Command Line Interface (AWS CLI) berikut untuk berlangganan fungsi Lambda di akun tujuan (akun B) ke topik SNS di akun sumber (akun A).
	
	Di akun A, jalankan perintah berikut:
	
	`aws sns add-permission \ --label lambda-access --aws-account-id <DestinationAccount> \ --topic-arn <Arn of SNSTopic > \ --action-name Subscribe ListSubscriptionsB yTopic`
	
	Di akun B, jalankan perintah berikut:
	
	`aws lambda add-permi ssion \ --function-name <Name of InvokeStepFunction > \ --source-arn <Arn of SNSTopic > \ --statement-id function-with-sns \ --action lambda:In vokeFunction \`
		Administrator cloud, arsitek Cloud, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--principal sns.amazonaws.com`
	
	Di akun B, jalankan perintah berikut:
	
	`aws sns subscribe \ --protocol "lambda" \ --topic-arn <Arn of SNSTopic> \ --notification-endpoint <Arn of InvokeStepFunction>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sinkronkan instans RDS DB dari akun sumber dengan akun tujuan.	Memulai replikasi database on-demand dengan memulai mesin status Step Functions di akun sumber. 1. Buka Konsol Step Functions. 2. Di panel navigasi, pilih mesin Negara. 3. Pilih mesin negara Anda. 4. Pada tab Eksekusi, pilih fungsi Anda, lalu pilih Mulai eksekusi untuk memulai alur kerja. **Note** Penjadwal tersedia untuk membantu Anda menjalankan replikasi secara otomatis sesuai jadwal, tetapi penjadwal dimatikan secara default. Anda dapat menemukan nama CloudWatch aturan Amazon untuk penjadwal di tab Sumber daya CloudFormation tumpukan di akun tujuan. Untuk petunjuk tentang cara mengubah aturan	Arsitek cloud, DevOps insinyur, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CloudWatch Peristiwa , lihat Menghapus atau Menonaktifkan Aturan CloudWatch Peristiwa dalam dokumentasi. CloudWatch	
Putar kembali database Anda ke salah satu salinan sebelumnya bila diperlukan.	1. Buka konsol Secrets Manager. 2. Dari daftar rahasia, pilih rahasia yang Anda buat dengan menggunakan CloudFormation template sebelumnya. Aplikasi Anda menggunakan rahasia untuk mengakses database di cluster tujuan. 3. Untuk memperbarui nilai rahasia dari halaman detail, di bagian Nilai rahasia, pilih Ambil nilai rahasia, lalu pilih Edit. 4. Masukkan detail titik akhir database.	Administrator cloud, DBA, DevOps insinyur

Sumber daya terkait

- Replika baca Lintas Wilayah (dokumentasi Amazon RDS)
- Penerapan Biru/Hijau (dokumentasi Amazon RDS)

Informasi tambahan

Anda dapat menggunakan contoh kebijakan berikut untuk membagikan informasi AWS KMS key Anda Akun AWS.

```
{
    "Version": "2012-10-17",
    "Id": "cross-account-rds-kms-key",
    "Statement": [
        {
            "Sid": "Enable user permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<SourceAccount>:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow administration of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<DestinationAccount>:root"
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": [
                    "arn:aws:iam::<DestinationAccount>:root",
                    "arn:aws:iam::<SourceAccount>:root"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey",
                "kms:CreateGrant"
            ],
            "Resource": "*"
        }
    ]
}
```

Mengotomatiskan tugas database dalam edisi SQL Server Express yang berjalan di Amazon EC2 dengan menggunakan AWS Lambda dan Penjadwal Tugas

Subhani Shaik, Amazon Web Services

Ringkasan

Pola ini menunjukkan bagaimana menjadwalkan dan mengelola tugas-tugas database dalam edisi SQL Server Express, yang merupakan versi gratis dari SQL Server. Namun, edisi SQL Server Express tidak memiliki layanan SQL Server Agent yang biasanya menangani operasi database otomatis. Pola ini menjelaskan bagaimana Anda dapat menggunakan Penjadwal Tugas dan Lambda sebagai alternatif untuk mengotomatiskan tugas database dalam edisi SQL Server Express yang berjalan pada instans Amazon Elastic Compute Cloud (Amazon). EC2

Penjadwal Tugas adalah utilitas sistem Windows bawaan yang memfasilitasi pelaksanaan otomatis tugas-tugas rutin. Ini menyediakan mekanisme untuk menjadwalkan dan mengelola operasi otomatis, menghilangkan kebutuhan untuk intervensi manual dalam proses berulang. AWS Lambdaadalah layanan komputasi tanpa server yang secara otomatis menjalankan kode sebagai respons terhadap peristiwa, tanpa mengharuskan Anda mengelola infrastruktur yang mendasarinya.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Virtual Private Cloud (VPC) yang dibuat dengan Amazon Virtual Private Cloud (Amazon VPC)
- EC2 Instans Amazon dengan Windows Server
- Volume Amazon Elastic Block Store (Amazon EBS) yang dilampirkan ke instans EC2 Amazon dengan Windows Server
- Binari SQL Server Express Edition

Batasan

- Untuk informasi tentang batasan fitur edisi SQL Server Express, lihat situs web Microsoft.

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- SQL Server 2016 atau yang lebih baru dengan edisi SQL Server Express

Arsitektur

Diagram berikut menunjukkan EC2 instans Amazon yang berjalan dengan edisi SQL Server Express diinstal. Instans ini dapat diakses melalui klien Remote Desktop Protocol (RDP) atau dari. AWS Systems Manager Session Manager AWS Key Management Service (AWS KMS) menangani enkripsi data untuk volume Amazon EBS untuk memastikan data-at-rest keamanan. Infrastruktur juga mencakup AWS Identity and Access Management (IAM), yang menyediakan kontrol akses dan mengelola izin untuk pelaksanaan fungsi Lambda. Amazon Simple Storage Service (Amazon S3) menyimpan fungsi Lambda.

Alat

Layanan AWS

- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon. EC2
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Systems Manager Session Manageradalah AWS Systems Manager alat yang dikelola sepenuhnya. Dengan Session Manager, Anda dapat mengelola EC2 instans Amazon, perangkat edge, server lokal, dan mesin virtual ()VMs.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat lainnya

- Microsoft SQL Server Management Studio (SSMS) adalah alat untuk mengelola SQL Server, termasuk mengakses, mengkonfigurasi, dan mengelola komponen SQL Server.
- Python adalah bahasa pemrograman komputer tujuan umum. Anda dapat menggunakannya untuk membangun aplikasi, mengotomatiskan tugas, dan mengembangkan layanan di AWS Cloudaplikasi.
- Penjadwal Tugas adalah alat Microsoft yang dapat Anda gunakan untuk menjadwalkan tugas rutin di komputer Anda secara otomatis.

Praktik terbaik

- Praktik terbaik untuk Amazon EC2
- Praktik terbaik untuk menyebarkan Microsoft SQL Server di Amazon EC2
- Praktik terbaik untuk bekerja dengan AWS Lambda fungsi
- Praktik terbaik keamanan di IAM

Epik

Buat EC2 instans Amazon dan instal edisi SQL Server Express

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan EC2 instans Amazon.	Untuk membuat EC2 instans Amazon, buka EC2 konsol	DBA, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon di https://console.aws.amazon.com/ec2/dan pilih Amazon Machine Image (AMI) dari daftar instance yang tersedia untuk Windows Server. Untuk informasi selengkapnya, lihat Meluncurkan EC2 instans Amazon di AWS dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal edisi SQL Server Express.	Untuk menginstal edisi SQL Server Express, selesaikan langkah-langkah berikut: 1. Untuk terhubung ke EC2 instans Amazon Anda, pilih opsi: • Opsi A — Gunakan Remote Desktop Protocol (RDP). Untuk petunjukn ya, lihat Connect to Windows Anda menggunakan klien RDP dalam dokumentasi. AWS • Opsi B — Gunakan EC2 konsol Amazon dan AWS Systems Manager Session Manager. Untuk petunjuknya, lihat Connect ke EC2 instans Amazon menggunakan Session Manager dalam AWS dokumentasi. 2. Untuk mengunduh edisi SQL Server Express yang diperlukan, buka Unduhan SQL Server di situs web Microsoft. 3. Untuk menginstal edisi SQL Server Express, ikuti petunjuk dalam Rencanaka n instalasi SQL Server di situs web Microsoft.	DBA, AWS DevOps

Buat tugas pemeliharaan basis data otomatis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi tugas rutin.	Identifikasi tugas rutin yang ingin Anda otomatiskan. Misalnya, tugas-tugas berikut memenuhi syarat untuk otomatisasi: • Backup database (Log penuh, diferensial, dan transaksi) • Pemeliharaan dan reorganisasi indeks • Pembaruan statistik • Operasi khusus aplikasi • Pembersihan atau pengarsipan data	DBA
Siapkan skrip SQL.	Untuk menyiapkan skrip SQL, lakukan hal berikut: 1. Buat kueri SQL untuk setiap tugas pemeliharaan. Berikut ini adalah contoh query T-SQL untuk melakukan backup database tertentu: `Backup Database <Database_Name> To Disk='C:\Backups\Database_Name.bak'` 2. Simpan file skrip sebagai`<File Name>.sql`. Kemudian, simpan skrip ke lokasi yang dapat diakses di drive lokal server	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di EC2 instance Amazon atau berbagi file jaringan.	
Konfigurasikan izin akses.	Untuk mengonfigurasi izin akses, lakukan hal berikut: 1. Tetapkan izin sistem file yang sesuai. Untuk petunjuk, lihat Mengkonfigurasi izin sistem file untuk akses Database Engine di situs web Microsoft. 2. Periksa apakah akun layanan SQL Server memiliki akses yang diperlukan. Untuk petunjuk, lihat Mengkonfigurasi akun layanan Windows dan izin di situs web Microsoft. 3. Verifikasi konektivitas jaringan untuk berbagi jarak jauh. Untuk informasi selengkapnya, lihat Mengakses data menggunakan berbagi file dalam AWS dokumentasi.	DBA

Mengotomatiskan tugas dengan Penjadwal Tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file batch.	• Untuk membuat file batch, gunakan editor teks untuk mengetik perintah	AWS DevOps, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	berikut. Ganti parameter `username` dan `password` dengan nilai Anda sendiri. Kemudian simpan file sebagai`<Name>.bat` .
	
	`sqlcmd -S servername -U username -P password -i <T-SQL query path.sql>`
	
	• Untuk membuat file batch untuk tugas SQL, gunakan editor teks dan ketik perintah berikut. Ganti parameter `ServerName` ,`DatabaseName` ,`username`, dan `password` dengan nilai Anda sendiri. Kemudian simpan file sebagai`<Name>.bat` .
	
	`@echo off`
	`sqlcmd -S [ServerName] -d [DatabaseName] -U username -P password -i "PathToSQLScript\Script.sql" -o "PathToOutput\Output.txt"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas di Penjadwal Tugas.	Untuk membuat tugas di Penjadwal Tugas, gunakan langkah-langkah berikut: 1. Untuk membuka Penjadwal Tugas, ketik taskschd.msc di pencarian Windows. 2. Pilih menu Action, lalu pilih Create Basic Task. 3. Untuk Nama, berikan nama untuk tugas tersebut, lalu pilih Berikutnya. 4. Untuk Trigger, pilih opsi kapan Anda ingin tugas dimulai dan kemudian pilih Berikutnya. 5. Berikan informasi Mulai dan Ulang untuk tugas tersebut, lalu pilih Berikutnya. 6. Untuk bagian Tindakan, pilih Mulai program dan kemudian pilih Berikutnya. 7. Untuk Program/skrip, tentukan jalur untuk file batch yang Anda buat di tugas sebelumnya dan kemudian pilih Berikutnya. 8. Pilih Selesai. Untuk menjalankan tugas secara manual, klik kanan tugas yang baru dibuat dan kemudian pilih Jalankan.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat status tugas.	Untuk melihat status tugas di Penjadwal Tugas, gunakan langkah-langkah berikut: 1. Di Penjadwal Tugas, buka Perpustakaan Penjadwal Tugas, yang menampilkan semua tugas. 2. Untuk melihat status tugas yang Anda buat sebelumnya, pilih tugas dan kemudian buka tab Riwayat.	DBA, AWS DevOps

Mengotomatiskan tugas dengan AWS Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan solusinya.	Untuk menerapkan solusi pola ini, gunakan langkah-langkah berikut: 1. Buat fungsi Lambda. Untuk petunjuk, lihat Membuat fungsi Lambda pertama Anda dalam dokumentasi. AWS 2. Jadwalkan fungsi Lambda. Untuk petunjuk, lihat Memanggil fungsi Lambda pada jadwal dalam AWS dokumentasi. 3. Jalankan kueri T-SQL. Untuk informasi selengkapnya, lihat Tutorial:	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Menggunakan fungsi Lambda untuk mengakses database Amazon RDS dalam dokumentasi. AWS Tutorial ini menjelaskan cara menghubungkan database Amazon RDS dari fungsi Lambda untuk menjalankan kueri SQL	

Pemecahan masalah

Isu	Solusi
Masalah Lambda	Untuk bantuan terkait kesalahan dan masalah yang mungkin Anda temui saat menggunakan AWS Lambda, lihat Memecahkan masalah di Lambda dalam dokumentasi. AWS

Sumber daya terkait

- Jenis EC2 instans Amazon
- AWS Lambda dokumentasi
- AWS Lambda harga
- Penjadwal Tugas untuk pengembang (situs web Microsoft)

Otomatiskan failover dan failback lintas wilayah dengan menggunakan DR Orchestrator Framework

Jitendra Kumar, Pavithra Balasubramanian, dan Oliver Francis, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan DR Orchestrator Framework untuk mengatur dan mengotomatiskan langkah-langkah manual yang rawan kesalahan untuk melakukan pemulihan bencana di seluruh Wilayah Amazon Web Services ().AWS Pola ini mencakup database berikut:

- Amazon Relational Database Service (Amazon RDS) untuk MySQL, Amazon RDS untuk PostgreSQL, atau Amazon RDS for MariaDB
- Amazon Aurora Edisi yang kompatibel dengan MySQL atau Edisi yang kompatibel dengan Amazon Aurora PostgreSQL (menggunakan file terpusat)
- Amazon ElastiCache (Redis OSS)

Untuk mendemonstrasikan fungsionalitas DR Orchestrator Framework, Anda membuat dua instance atau cluster DB. Yang utama ada di Wilayah AWS `us-east-1`, dan yang sekunder ada di`us-west-2`. Untuk membuat sumber daya ini, Anda menggunakan AWS CloudFormation template di `App-Stack` folder repositori aws-cross-region-dr-databases GitHub .

Prasyarat dan batasan

Prasyarat umum

- DR Orchestrator Framework digunakan di primer dan sekunder Wilayah AWS
- Dua ember Layanan Penyimpanan Sederhana Amazon
- Virtual Private Cloud (VPC) dengan dua subnet dan grup keamanan AWS

Prasyarat khusus mesin

- Amazon Aurora — Setidaknya satu database global Aurora harus tersedia dalam dua. Wilayah AWS Anda dapat menggunakan `us-east-1` sebagai Wilayah utama, dan digunakan `us-west-2` sebagai Wilayah sekunder.

- Amazon ElastiCache (Redis OSS) — Datastore ElastiCache global harus tersedia dalam dua. Wilayah AWS Anda dapat `use us-east-1` sebagai Wilayah utama, dan digunakan `us-west-2` sebagai Wilayah sekunder.

Keterbatasan Amazon RDS

- DR Orchestrator Framework tidak memeriksa kelambatan replikasi sebelum melakukan failover atau failback. Kelambatan replikasi harus diperiksa secara manual.
- Solusi ini telah diuji menggunakan instance database utama dengan satu replika baca. Jika Anda ingin menggunakan lebih dari satu replika baca, uji solusinya secara menyeluruh sebelum menerapkannya di lingkungan produksi.

Keterbatasan Aurora

- Ketersediaan dan dukungan fitur bervariasi di seluruh versi tertentu dari setiap mesin database dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang fitur dan ketersediaan Wilayah untuk replikasi Lintas wilayah, lihat replika baca Lintas Wilayah.
- Database global Aurora memiliki persyaratan konfigurasi khusus untuk kelas instans Aurora DB yang didukung dan jumlah maksimum. Wilayah AWS Untuk informasi selengkapnya, lihat Persyaratan konfigurasi database global Amazon Aurora.
- Solusi ini telah diuji menggunakan instance database utama dengan satu replika baca. Jika Anda ingin menggunakan lebih dari satu replika baca, uji solusinya secara menyeluruh sebelum menerapkannya di lingkungan produksi.

ElastiCache keterbatasan

- Untuk informasi tentang ketersediaan Wilayah untuk Datastore Global dan persyaratan ElastiCache konfigurasi, lihat Prasyarat dan batasan dalam dokumentasi. ElastiCache

Amazon RDS p versi roduct

Amazon RDS mendukung versi mesin berikut:

- MySQL — Amazon RDS mendukung instans DB yang menjalankan versi MySQL berikut: MySQL 8.0 dan MySQL 5.7

- PostgreSQL — Untuk informasi tentang versi Amazon RDS yang didukung untuk PostgreSQL, lihat Versi database PostgreSQL yang tersedia.
- MariaDB - Amazon RDS mendukung instans DB yang menjalankan versi MariaDB berikut:
  - MariaDB 10.11
  - MariaDB 10.6
  - MariaDB 10.5

Versi produk Aurora

- Peralihan basis data global Amazon Aurora memerlukan Aurora MySQL yang kompatibel dengan kompatibilitas MySQL 5.7, versi 2.09.1 dan yang lebih tinggi

  Untuk informasi selengkapnya, lihat Batasan database global Amazon Aurora.

ElastiCache (Redis OSS) versi produk

Amazon ElastiCache (Redis OSS) mendukung versi Redis berikut:

- Redis 7.1 (ditingkatkan)
- Redis 7.0 (ditingkatkan)
- Redis 6.2 (ditingkatkan)
- Redis 6.0 (ditingkatkan)
- Redis 5.0.6 (ditingkatkan)

Untuk informasi selengkapnya, lihat Versi yang didukung ElastiCache (Redis OSS).

Arsitektur

Arsitektur Amazon RDS

Arsitektur Amazon RDS mencakup sumber daya berikut:

- Instans Amazon RDS DB utama yang dibuat di Region (`us-east-1`) utama dengan read/write akses untuk klien
- Replika baca Amazon RDS yang dibuat di Region sekunder (`us-west-2`) dengan akses hanya-baca untuk klien

- DR Orchestrator Framework dikerahkan di Wilayah primer dan sekunder

Diagram menunjukkan yang berikut:

1. Replikasi asinkron antara instance primer dan instance sekunder
2. Akses baca/tulis untuk klien di Wilayah utama
3. Akses hanya-baca untuk klien di Wilayah sekunder

Arsitektur Aurora

Arsitektur Amazon Aurora mencakup sumber daya berikut:

- Cluster Aurora DB utama yang dibuat di Region (`us-east-1`) primer dengan titik akhir penulis aktif
- Cluster Aurora DB yang dibuat di Region sekunder (`us-west-2`) dengan titik akhir penulis yang tidak aktif
- DR Orchestrator Framework dikerahkan di Wilayah primer dan sekunder

Diagram menunjukkan yang berikut:

1. Replikasi asinkron antara cluster primer dan cluster sekunder
2. Cluster DB utama dengan titik akhir penulis aktif
3. Cluster DB sekunder dengan titik akhir penulis yang tidak aktif

ElastiCache Arsitektur (Redis OSS)

Arsitektur Amazon ElastiCache (Redis OSS) mencakup sumber daya berikut:

- Datastore global ElastiCache (Redis OSS) yang dibuat dengan dua cluster:
  1. Cluster utama di Region primer (`us-east-1`)
  2. Cluster sekunder di Region sekunder (`us-west-2`)
- Tautan Amazon Cross-region dengan enkripsi TLS 1.2 antara kedua cluster
- DR Orchestrator Framework dikerahkan di Wilayah primer dan sekunder

Otomatisasi dan skala

DR Orchestrator Framework dapat diskalakan dan mendukung failover atau failback lebih dari satu database secara paralel. AWS

Anda dapat menggunakan kode payload berikut untuk gagal melalui beberapa AWS database di akun Anda. Dalam contoh ini, tiga AWS database (dua database global seperti Aurora MySQL kompatibel atau Aurora PostgreSQL kompatibel, dan satu Amazon RDS untuk MySQL instance) gagal ke Wilayah DR:

```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "PlannedFailoverAurora",
          "resourceName": "Switchover (planned failover) of Amazon Aurora global
 databases (MySQL)",
          "parameters": {
            "GlobalClusterIdentifier": "!Import dr-globaldb-cluster-mysql-global-
identifier",
            "DBClusterIdentifier": "!Import dr-globaldb-cluster-mysql-cluster-
identifier"
          }
        },
        {
          "resourceType": "PlannedFailoverAurora",
          "resourceName": "Switchover (planned failover) of Amazon Aurora global
 databases (PostgreSQL)",
          "parameters": {
            "GlobalClusterIdentifier": "!Import dr-globaldb-cluster-postgres-global-
identifier",
            "DBClusterIdentifier": "!Import dr-globaldb-cluster-postgres-cluster-
identifier"
          }
        },
        {
          "resourceType": "PromoteRDSReadReplica",
          "resourceName": "Promote RDS for MySQL Read Replica",
          "parameters": {
```

```
            "RDSInstanceIdentifier": "!Import rds-mysql-instance-identifier",
            "TargetClusterIdentifier": "!Import rds-mysql-instance-global-arn"
          }
        }
      ]
    }
  ]
}
```

Alat

AWS layanan

- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.
- Amazon ElastiCache membantu Anda mengatur, mengelola, dan menskalakan lingkungan cache dalam memori terdistribusi di. AWS Cloud Pola ini menggunakan Amazon ElastiCache (Redis OSS).
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan. Dalam pola ini, fungsi Lambda digunakan oleh AWS Step Functions untuk melakukan langkah-langkah.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud Pola ini mendukung Amazon RDS for MySQL, Amazon RDS for PostgreSQL, dan Amazon RDS for MariaDB.
- AWS SDK for Python (Boto3)membantu Anda mengintegrasikan aplikasi, pustaka, atau skrip Python Anda. Layanan AWS Dalam pola ini, Boto3 APIs digunakan untuk berkomunikasi dengan instance database atau database global.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting. Dalam pola ini, mesin status Step Functions digunakan untuk mengatur dan menjalankan failover Cross-region dan failback dari instance database atau database global.

Repositori kode

Kode untuk pola ini tersedia di repositori aws-cross-region-dr-databases pada. GitHub

Epik

Instal Kerangka Orkestrator DR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Untuk mengkloning repositori, jalankan perintah berikut: `git clone https://github.com/aws-samples/aws-cross-region-dr-databases.git`	AWS DevOps, administrator AWS

Package Lambda berfungsi kode dalam arsip file.zip.	Buat file arsip untuk fungsi Lambda untuk menyertakan dependensi DR Orchestrator Framework:
	`cd <YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts`
	
	`bash scripts/deploy-orchestrator-sh.sh`
		Administrator AWS

Buat ember S3.	Bucket S3 diperlukan untuk menyimpan DR Orchestrator Framework bersama dengan konfigurasi terbaru Anda. Buat dua bucket S3, satu di Region utama (`us-east-1`), dan satu di Region sekunder (`us-west-2`): • `dr-orchestrator-xxxxx-us-east-1`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `dr-orchestrator-xx xxxx-us-west-2` Ganti `xxxxxx` dengan nilai acak untuk membuat nama bucket unik.	
Buat subnet dan grup keamanan.	Di Region (`us-east-1`) primer dan Region sekunder (`us-west-2`), buat dua subnet dan satu grup keamanan untuk penerapan fungsi Lambda di VPC Anda: • `subnet-XXXXXXX` • `subnet-YYYYYYY` • `sg-XXXXXXXXXXX`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file parameter DR Orchestrator.	Di `<YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts/cloudformation` folder, perbarui file parameter DR Orchestrator berikut: • `Orchestrator-Deployer-parameters-us-east-1.json` • `Orchestrator-Deployer-parameters-us-west-2.json` Gunakan nilai parameter berikut, ganti `x` dan `y` dengan nama sumber daya Anda: `[ { "ParameterKey": "TemplateStoreS3BucketName", "ParameterValue": "dr-orchestrator-xxxxxx-us-east-1" }, { "ParameterKey": "TemplateVPCId", "ParameterValue": "vpc-xxxxxx" }, {`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"ParameterKey": "TemplateLambdaSubnetID1", "ParameterValue": "subnet-xxxxx" }, { "ParameterKey": "TemplateLambdaSubnetID2", "ParameterValue": "subnet-yyyyy" }, { "ParameterKey": "TemplateLambdaSecurityGroupID", "ParameterValue": "sg-xxxxxxxxxx" }]	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode DR Orchestrator Framework ke bucket S3.	Kode akan lebih aman di ember S3 daripada di direktori lokal. Unggah `DR-Orchestration-artifacts` direktori, termasuk semua file dan subfolder, ke bucket S3. Untuk mengunggah kode, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS. 2. Arahkan ke konsol Amazon S3. 3. Pilih`dr-orchestrator-xxxxxx-us-east-1` `bucket`. 4. Pilih Unggah, lalu pilih Tambah folder. 5. Pilih `DR-Orchestration-artifacts` folder. 6. Pilih Unggah. 7. Pilih `dr-orchestrator-xxxxxx-us-west-2` ember. 8. Ulangi langkah 4—7.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan Kerangka Orkestrator DR di Wilayah utama.	Untuk menerapkan DR Orchestrator Framework di Region (`us-east-1`) primer, jalankan perintah berikut:
	`cd <YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts/cloudformation`
	
	`aws cloudformation deploy \`
	`--region us-east-1 \`
	`--stack-name dr-orchestrator \`
	`--template-file Orchestrator-Deployer.yaml \`
	`--parameter-overrides file://Orchestrator-Deployer-parameters-us-east-1.json \`
	`--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \`
	`--disable-rollback`
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan Kerangka Orkestrator DR di Wilayah sekunder.	Di Region sekunder (`us-west-2`), jalankan perintah berikut:
	`cd <YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts/cloudformation`
	
	`aws cloudformation deploy \`
	`--region us-west-2 \`
	`--stack-name dr-orchestrator \`
	`--template-file Orchestrator-Deployer.yaml \`
	`--parameter-overrides file://Orchestrator-Deployer-parameters-us-west-2.json \`
	`--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \`
	`--disable-rollback`
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi penyebaran.	Jika CloudFormation perintah berjalan berhasil, ia mengembalikan output berikut: `Successfully created/ updated stack - dr-orchestrator` Atau, Anda dapat menavigasi ke CloudFormation konsol dan memverifikasi status `dr-orchestrator` tumpukan.	Administrator AWS

Membuat instance database atau cluster

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat subnet database dan grup keamanan.	Di VPC Anda, buat dua subnet dan satu grup keamanan untuk instans DB atau database global di Wilayah primer (`us-east-1`) dan sekunder (`us-west-2`): • `subnet-XXXXXX` • `subnet-XXXXXX` • `sg-XXXXXXXXXX`	Administrator AWS
Perbarui file parameter untuk instance atau cluster DB primer.	Di `<YOUR LOCAL GIT FOLDER>/App-Stack` folder, perbarui file parameter untuk Wilayah utama. Amazon RDS	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Dalam `RDS-MySQL-parameter-us-east-1.json` file, perbarui `SubnetIds` dan `DBSecurityGroup` dengan nama sumber daya yang Anda buat: `{ "Parameters": { "SubnetIds": "subnet-xxxxxx,subnet-xxxxxx", "DBSecurityGroup": "sg-xxxxxxxxxx", "MySqlGlobalIdentifier":"rds-mysql-instance", "InitialDatabaseName": "mysqldb", "DBPortNumber": "3789", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2", "KMSKeyAliasName": "rds/rds-mysql-instance-KmsKeyId" } }`  Amazon Aurora  Dalam `Aurora-MySQL-parameter-us-east-1.json` file, perbarui `SubnetIds` dan `DBSecurityGroup` dengan nama sumber daya yang Anda buat:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>{
	  "Parameters": {
	    "SubnetIds":
	 "subnet1-xxxxxx,su
	bnet2-xxxxxx",
	    "DBSecurityGroup":
	 "sg-xxxxxxxxxx",
	    "GlobalClusterIden
	tifier":"dr-globaldb-
	cluster-mysql",
	    "DBClusterName":"d
	bcluster-01",
	    "SourceDBClusterNa
	me":"dbcluster-02",
	    "DBPortNumber":
	 "3787",
	    "DBInstanceClass":
	"db.r5.large",
	    "InitialDatabaseNa
	me": "sampledb",
	    "PrimaryRegion":
	 "us-east-1",
	    "SecondaryRegion":
	 "us-west-2",
	    "KMSKeyAliasName":
	 "rds/dr-globaldb-c
	luster-mysql-KmsKe
	yId"
	  }
	}</pre>
	Amazon ElastiCache (Redis OSS)
	
	Dalam `ElastiCache-parameter-us-east-1.json` file, perbarui `SubnetIds` dan `DBSecurit`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yGroup dengan nama sumber daya yang Anda buat. `{ "Parameters": { "CacheNodeType": "cache.m5.large", "DBSecurityGroup": "sg-xxxxxxxxxx", "SubnetIds": "subnet-xxxxxx,subnet-xxxxxx", "EngineVersion": "5.0.6", "GlobalReplicationGroupIdSuffix": "demo-redis-global-datastore", "NumReplicas": "1", "NumShards": "1", "ReplicationGroupId": "demo-redis-cluster", "DBPortNumber": "3788", "TransitEncryption": "true", "KMSKeyAliasName": "elasticache/demo-redis-global-datastore-KmsKeyId", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2" } }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan instans atau cluster DB Anda di Wilayah utama.	Untuk menerapkan instance atau cluster Anda di Region (`us-east-1`) primer, jalankan perintah berikut berdasarkan mesin database Anda.
	
	Amazon RDS
	
	`cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack`
	
	`aws cloudformation deploy \`
	`--region us-east-1 \`
	`--stack-name rds-mysql-app-stack \`
	`--template-file RDS-MySQL-Primary.yaml \`
	`--parameter-overrides file://RDS-MySQL-parameter-us-east-1.json \`
	`--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \`
	`--disable-rollback`
	
	Amazon Aurora
	
	`cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack`
	
	`aws cloudformation deploy \`
	`--region us-east-1 \`
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--stack-name aurora-mysql-app-stack \`
	`--template-file Aurora-MySQL-Primary.yaml \`
	`--parameter-overrides file://Aurora-MySQL-parameter-us-east-1.json \`
	`--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \`
	`--disable-rollback`
	
	Amazon ElastiCache (Redis OSS)
	
	`cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack`
	
	`aws cloudformation deploy \`
	`--region us-east-1 --stack-name elasticache-ds-app-stack \`
	`--template-file ElastiCache-Primary.yaml \`
	`--parameter-overrides file://ElastiCache-parameter-us-east-1.json \`
	`--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \`
	`--disable-rollback`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Verifikasi bahwa CloudFormation sumber daya berhasil digunakan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file parameter untuk instance atau cluster DB sekunder.	Di `<YOUR LOCAL GIT FOLDER>/App-Stack` folder, perbarui file parameter untuk Wilayah sekunder. Amazon RDS Dalam `RDS-MySQL-parameter-us-west-2.json` file, perbarui `SubnetIDs` dan `DBSecurityGroup` dengan nama sumber daya yang Anda buat. Perbarui `PrimaryRegionKMSKeyArn` dengan nilai yang `MySQLKmsKeyId` diambil dari bagian Output dari CloudFormation tumpukan untuk instance DB utama: `{ "Parameters": { "SubnetIds": "subnet-aaaaaaaaa, subnet-bbbbbbbbb", "DBSecurityGroup": "sg-ccccccccc", "MySqlGlobalIdentifier":"rds-mysql-instance", "InitialDatabaseName": "mysqldb", "DBPortNumber": "3789", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2",`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"KMSKeyAliasName": "rds/rds-mysql-instance-KmsKeyId",
	"PrimaryRegionKMSKeyArn":"arn:aws:kms:us-east-1:xxxxxxxxx:key/mrk-xxxxxxxxxxxxxxxxx"
	}
	}</pre>
	Amazon Aurora
	
	Dalam `Aurora-MySQL-parameter-us-west-2.json` file, perbarui `SubnetIDs` dan `DBSecurityGroup` dengan nama sumber daya yang Anda buat. Perbarui `PrimaryRegionKMSKeyArn` dengan nilai yang `AuroraKmsKeyId` diambil dari bagian Output dari CloudFormation tumpukan untuk instance DB utama:
	
	<pre>{
	"Parameters": {
	"SubnetIds": "subnet1-aaaaaaaaa,subnet2-bbbbbbbbb",
	"DBSecurityGroup": "sg-ccccccccc",
	"GlobalClusterIdentifier":"dr-globaldb-cluster-mysql",
	"DBClusterName":"dbcluster-01",	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"SourceDBClusterName":"dbcluster-02",
	"DBPortNumber": "3787",
	"DBInstanceClass": "db.r5.large",
	"InitialDatabaseName": "sampledb",
	"PrimaryRegion": "us-east-1",
	"SecondaryRegion": "us-west-2",
	"KMSKeyAliasName": "rds/dr-globaldb-cluster-mysql-KmsKeyId"
	}
	}</pre>
	Amazon ElastiCache (Redis OSS)
	
	Dalam `ElastiCache-parameter-us-west-2.json` file, perbarui `SubnetIDs` dan `DBSecurityGroup` dengan nama sumber daya yang Anda buat. Perbarui `PrimaryRegionKMSKeyArn` dengan nilai yang `ElastiCacheKmsKeyId` diambil dari bagian Output dari CloudFormation tumpukan untuk instance DB utama:
	
	<pre>{
	"Parameters": {</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"CacheNodeType": "cache.m5.large", "DBSecurityGroup": "sg-ccccccccc", "SubnetIds": "subnet-aaaaaaaaa, subnet-bbbbbbbbb", "EngineVersion": "5.0.6", "GlobalReplicationGroupIdSuffix": "demo-redis-global-datastore", "NumReplicas": "1", "NumShards": "1", "ReplicationGroupId": "demo-redis-cluster", "DBPortNumber": "3788", "TransitEncryption": "true", "KMSKeyAliasName": "elasticache/demo-redis-global-datastore-KmsKeyId", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2" } }	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan instans atau cluster DB Anda di Wilayah sekunder.	Jalankan perintah berikut, berdasarkan mesin database Anda.
	
	Amazon RDS
	
	`cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack`
	
	`aws cloudformation deploy \`
	`--region us-west-2 \`
	`--stack-name rds-mysql-app-stack \`
	`--template-file RDS-MySQL-DR.yaml \`
	`--parameter-overrides file://RDS-MySQL-parameter-us-west-2.json \`
	`--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \`
	`--disable-rollback`
	
	Amazon Aurora
	
	`cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack`
	
	`aws cloudformation deploy \`
	`--region us-west-2 \`
	`--stack-name aurora-mysql-app-stack \`
	`--template-file Aurora-MySQL-DR.yaml \`
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--parameter-overrides file://Aurora-MySQL-parameter-us-west-2.json \ --capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \ --disable-rollback`
	
	Amazon ElastiCache (Redis OSS)
	
	`cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack`
	
	`aws cloudformation deploy \ --region us-west-2 \ --stack-name elasticache-ds-app-stack \ --template-file ElastiCache-DR.yaml \ --parameter-overrides file://ElastiCache-parameter-us-west-2.json \ --capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \ --disable-rollback`
	
	Verifikasi bahwa CloudFormation sumber daya berhasil digunakan.	

Sumber daya terkait

- Strategi pemulihan bencana untuk database pada AWS (Strategi Panduan AWS Preskriptif)
- Otomatiskan solusi DR Anda untuk database relasional di AWS(panduan panduan AWS preskriptif)
- Menggunakan basis data global Amazon Aurora
- Replikasi di seluruh Wilayah AWS menggunakan datastores global
- Otomatiskan solusi DR Anda untuk database relasional di AWS(panduan panduan AWS preskriptif)

Secara otomatis mencadangkan database SAP HANA menggunakan Systems Manager dan EventBridge

Ambarish Satarkar dan Gaurav Rath, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara mengotomatiskan backup database SAP HANA menggunakan AWS Systems Manager, Amazon, Amazon Simple Storage Service (EventBridgeAmazon S3), dan AWS Backint Agent untuk SAP HANA.

Pola ini menyediakan pendekatan berbasis skrip shell menggunakan `BACKUP DATA` perintah dan menghilangkan kebutuhan untuk memelihara skrip dan konfigurasi pekerjaan untuk setiap instance sistem operasi (OS) di berbagai sistem.

> Catatan: Mulai April 2023, AWS Backup mengumumkan dukungan untuk database SAP HANA di Amazon Elastic Compute Cloud (Amazon). EC2 Untuk informasi selengkapnya, lihat database SAP HANA di cadangan EC2 instans Amazon.
>
> Berdasarkan kebutuhan organisasi Anda, Anda dapat menggunakan layanan AWS Backup untuk secara otomatis mencadangkan database SAP HANA Anda atau Anda dapat menggunakan pola ini.

Prasyarat dan batasan

Prasyarat

- Instans SAP HANA yang ada dengan rilis yang didukung dalam status berjalan pada instans Amazon Elastic Compute Cloud EC2 (Amazon) terkelola yang dikonfigurasi untuk Systems Manager
- Agen Systems Manager (Agen SSM) 2.3.274.0 atau yang lebih baru diinstal
- Bucket S3 yang tidak mengaktifkan akses publik
- Sebuah `hdbuserstore` kunci bernama `SYSTEM`
- Peran AWS Identity and Access Management (IAM) agar runbook Otomasi berjalan sesuai jadwal
- `AmazonSSMManagedInstanceCore`dan `ssm:StartAutomationExecution` kebijakan dilampirkan pada peran layanan Systems Manager Automation.

Batasan

- AWS Backint Agent untuk SAP HANA tidak mendukung deduplikasi.
- AWS Backint Agent untuk SAP HANA tidak mendukung kompresi data.

Versi produk

AWS Backint Agent didukung pada sistem operasi berikut:

- Server Perusahaan SUSE Linux
- SUSE Linux Enterprise Server untuk SAP
- Red Hat Enterprise Linux untuk SAP

AWS Backint Agent mendukung database berikut:

- SAP HANA 1.0 SP12 (simpul tunggal dan beberapa node)
- SAP HANA 2.0 dan yang lebih baru (simpul tunggal dan beberapa node)

Arsitektur

Tumpukan teknologi target

- AWS Backint Agent
- Amazon S3
- AWS Systems Manager
- Amazon EventBridge
- SAP HANA

Arsitektur target

Diagram berikut menunjukkan skrip instalasi yang menginstal AWS Backint Agent, bucket S3, dan Systems Manager dan EventBridge, yang menggunakan dokumen Command untuk menjadwalkan pencadangan reguler.

Otomatisasi dan skala

- Beberapa Agen AWS Backint dapat diinstal dengan menggunakan runbook Automation Systems Manager.
- Setiap runbook Systems Manager dapat menskalakan ke n jumlah instans SAP HANA, berdasarkan pemilihan target.
- EventBridge dapat mengotomatiskan cadangan SAP HANA.

Alat

- AWS Backint Agent for SAP HANA adalah aplikasi mandiri yang terintegrasi dengan alur kerja yang ada untuk mencadangkan database SAP HANA Anda ke bucket S3 yang Anda tentukan dalam file konfigurasi. AWS Backint Agent mendukung pencadangan penuh, inkremental, dan diferensial dari database SAP HANA. Ini berjalan pada server database SAP HANA, di mana backup dan katalog ditransfer dari database SAP HANA ke AWS Backint Agent.
- Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge mengirimkan aliran data real-time dari aplikasi Anda, aplikasi perangkat lunak sebagai layanan (SaaS), dan layanan AWS ke target seperti fungsi AWS Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun lain.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web.
- AWS Systems Manager membantu Anda melihat dan mengontrol infrastruktur di AWS. Menggunakan konsol Systems Manager, Anda dapat melihat data operasional dari beberapa layanan AWS dan mengotomatiskan tugas operasional di seluruh sumber daya AWS Anda.

Kode

Kode untuk pola ini tersedia di aws-backint-automated-backup GitHub repositori.

Epik

Buat sistem kunci hdbuserstore

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat kunci hdbuserstore.	1. Navigasi ke `/usr/sap/<SID>/HDB<InstNo>/exe`.
	2. Jalankan perintah berikut, dengan XX sebagai nomor instans database SAP HANA.
	`hdbuserstore -i set SYSTEM <hostname>:3XX13@SYSTEMDB SYSTEM`
	Misalnya, untuk host SAP HANA `saphanadb` dengan nomor instance`00`, jalankan perintah berikut.
	`hdbuserstore -i set SYSTEM saphanadb:30013@SYSTEMDB SYSTEM`
		Administrator AWS, Administrator SAP HANA

Instal AWS Backint Agent

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS Backint Agent.	Ikuti petunjuk di Instal dan konfigurasikan AWS Backint Agent untuk SAP HANA di	Administrator AWS, administrator SAP HANA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dokumentasi AWS Backint Agent.	

Buat dokumen Command Systems Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dokumen Systems Manager Command.	1. Masuk ke AWS Management Console dan buka AWS Systems Manager Console. 2. Pilih Dokumen, dan pilih Dimiliki oleh saya. 3. Konfirmasikan bahwa Anda berada di Wilayah AWS yang sama dengan database SAP HANA Anda. 4. Pilih Buat dokumen, Perintah atau sesi untuk membuat dokumen Anda. 5. Gunakan nama yang unik dan deskriptif, tanpa spasi (misalnya, SAP Hana-backup). 6. Pastikan bahwa jenis Dokumen diatur ke dokumen Command. 7. Di bawah header Konten, ada beberapa kode contoh. Pastikan Anda memilih jenis kode JSON, dan ganti kode dengan kode dari `HDB_Backup_SSM_Doc`	Administrator AWS, administrator SAP HANA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ument.json` file dari GitHub repositori. 8. Pilih Buat dokumen. 9. Periksa dokumen Anda di bagian Dimiliki oleh saya.	

Jadwalkan pencadangan pada frekuensi reguler

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jadwalkan pencadangan reguler menggunakan Amazon. EventBridge	1. Buka EventBridge konsol Amazon, pilih Aturan, dan pilih Buat aturan. 2. Pada Tentukan detail aturan layar, masukkan nama dan deskripsi unik untuk aturan Anda, dan gunakan bus acara default. 3. Di bawah Jenis aturan, pilih Jadwal, dan pilih Berikutnya. 4. Pada layar Tentukan jadwal, pilih pola jadwal yang sesuai dan ekspresi cron atau rate berdasarkan frekuensi yang diperlukan. 5. Pada layar Pilih target, untuk jenis Target, pilih layanan AWS. Di bawah Pilih target, pilih Systems Manager Run Command. 6. Pilih dokumen yang Anda buat sebelumnya.	Administrator AWS, administrator SAP HANA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Di bawah kunci Target dan nilai Target, berikan ID instance. Anda dapat menggunakan nama tag dan nilai tag untuk menambahkan beberapa instance. 8. Di bawah Konfigurasi parameter otomatisasi, pilih Konstan untuk pencadangan tambahan atau diferensial. Jika Anda ingin cadangan penuh, pilih Tidak Ada Parameter. 9. Pilih apakah akan membuat peran baru atau menggunakan peran yang sudah ada. Jika Anda menggunakan peran yang ada, pastikan bahwa itu memiliki kebijakan yang diperlukan untuk memanggil target. 10. Pertahankan pengaturan tambahan default, dan pilih Berikutnya. 11. Layar Konfigurasi tag adalah opsional. Pilih berikutnya. 12. Pada layar Tinjau dan buat, tinjau pengaturan aturan, dan pilih Buat. Aturan harus berhasil dibuat.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda dapat memverifikasi keberhasilan pencadangan dari jalur bucket S3. `s3:/<your_bucket_name>/<target folder>/<SID>/usr/sap/<SID>/SYS/global/hdb/backint/DB_<SID>/` Anda juga dapat memverifikasi cadangan dari katalog cadangan SAP HANA.	

Sumber daya terkait

- AWS Backint Agent untuk SAP HANA
- Instal dan konfigurasikan AWS Backint Agent untuk SAP HANA

Secara otomatis menghasilkan model PynamoDB dan fungsi CRUD untuk Amazon DynamoDB dengan menggunakan aplikasi Python

Vijit Vashishtha, Dheeraj Alimchandani, dan Dhananjay Karanjkar, Amazon Web Services

Ringkasan

Adalah umum untuk meminta entitas dan membuat, membaca, memperbarui, dan menghapus (CRUD) fungsi operasi untuk secara efisien melakukan operasi database Amazon DynamoDB. PynamoDB adalah antarmuka berbasis Python yang mendukung Python 3. Ini juga menyediakan fitur seperti dukungan untuk transaksi Amazon DynamoDB, serialisasi nilai atribut otomatis dan deserialisasi, dan kompatibilitas dengan kerangka kerja Python umum, seperti Flask dan Django. Pola ini membantu pengembang bekerja dengan Python dan DynamoDB dengan menyediakan perpustakaan yang merampingkan pembuatan otomatis model PynamoDB dan fungsi operasi CRUD. Meskipun menghasilkan fungsi CRUD penting untuk tabel database, ia juga dapat merekayasa balik model PynamoDB dan fungsi CRUD dari tabel Amazon DynamoDB. Pola ini dirancang untuk menyederhanakan operasi database dengan menggunakan aplikasi berbasis Python.

Berikut ini adalah fitur utama dari solusi ini:

- Skema JSON ke model PynamoDB - Secara otomatis menghasilkan model PynamodB dengan Python dengan mengimpor file skema JSON.
- Pembuatan fungsi CRUD - Secara otomatis menghasilkan fungsi untuk melakukan operasi CRUD pada tabel DynamoDB.
- Rekayasa balik dari DynamoDB - Gunakan pynamoDB object-relational mapping (ORM) untuk merekayasa balik model PynamoDB dan fungsi CRUD untuk tabel Amazon DynamoDB yang ada.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Python versi 3.8 atau yang lebih baru, diunduh dan diinstal
- Jinja2 versi 3.1.2 atau yang lebih baru, diunduh dan diinstal
- Tabel Amazon DynamoDB yang ingin Anda hasilkan ORM
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- PynamoDB versi 5.4.1 atau yang lebih baru, diinstal

Arsitektur

Tumpukan teknologi target

- Skrip JSON
- Aplikasi Python
- Model PynamoDB
- Contoh basis data Amazon DynamoDB

Arsitektur target

1. Anda membuat file skema JSON masukan. File skema JSON ini mewakili atribut dari tabel DynamoDB masing-masing yang Anda ingin membuat model PynamoDB dari dan fungsi CRUD untuk. Ini berisi tiga kunci penting berikut:
   - `name`—Nama tabel DynamoDB target.
   - `region`— Wilayah AWS Tempat tabel dihosting
   - `attributes`Atribut yang merupakan bagian dari tabel target, seperti kunci partisi (juga dikenal sebagai atribut hash), kunci sortir, indeks sekunder lokal, indeks sekunderglobal, dan atribut non-kunci apa pun. Alat ini mengharapkan skema input hanya menyediakan atribut non-kunci karena aplikasi mengambil atribut kunci langsung dari tabel target. Untuk contoh cara menentukan atribut dalam file skema JSON, lihat bagian Informasi tambahan dari pola ini.
2. Jalankan aplikasi Python dan berikan file skema JSON sebagai masukan.
3. Aplikasi Python membaca file skema JSON.
4. Aplikasi Python terhubung ke tabel DynamoDB untuk mendapatkan skema dan tipe data. Aplikasi ini menjalankan operasi describe_table dan mengambil atribut kunci dan indeks untuk tabel.
5. Aplikasi Python menggabungkan atribut dari file skema JSON dan tabel DynamoDB. Ini menggunakan mesin template Jinja untuk menghasilkan model PynamoDB dan fungsi CRUD yang sesuai.
6. Anda mengakses model PynamoDB untuk melakukan operasi CRUD pada tabel DynamoDB.

Alat

Layanan AWS

- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.

Alat-alat lainnya

- Jinja adalah mesin templating extensible yang mengkompilasi template ke dalam kode Python yang dioptimalkan. Pola ini menggunakan Jinja untuk menghasilkan konten dinamis dengan menyematkan placeholder dan logika dalam template.
- PynamoDB adalah antarmuka berbasis Python untuk Amazon DynamoDB.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia dalam model PynamoDB yang GitHub dihasilkan otomatis dan repositori fungsi CRUD. Repositori dibagi menjadi dua bagian utama: paket controller dan template.

Controller paket

Paket Python controller berisi logika aplikasi utama yang membantu menghasilkan model PynamoDB dan fungsi CRUD. Ini berisi yang berikut:

- `input_json_validator.py`- Skrip Python ini memvalidasi file skema JSON masukan dan membuat objek Python yang berisi daftar tabel DynamoDB target dan atribut yang diperlukan untuk masing-masing.
- `dynamo_connection.py`— Skrip ini membuat koneksi ke tabel DynamoDB dan menggunakan operasi untuk mengekstrak atribut `describe_table` yang diperlukan untuk membuat model PynamoDB.
- `generate_model.py`- Script ini berisi kelas Python `GenerateModel` yang menciptakan model PynamoDB berdasarkan file skema JSON masukan dan operasi. `describe_table`
- `generate_crud.py`— Untuk tabel DynamoDB yang didefinisikan dalam file skema JSON, skrip ini menggunakan `GenerateCrud` operasi untuk membuat kelas Python.

Template

Direktori Python ini berisi template Jinja berikut:

- `model.jinja`- Template Jinja ini berisi ekspresi template untuk menghasilkan skrip model PynamoDB.
- `crud.jinja`- Template Jinja ini berisi ekspresi template untuk menghasilkan skrip fungsi CRUD.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Masukkan perintah berikut untuk mengkloning model PynamoDB yang dihasilkan otomatis dan repositori fungsi CRUD. `git clone https://github.com/aws-samples/amazon-reverse-engineer-dynamodb.git`	Pengembang aplikasi

Siapkan lingkungan Python.	1. Arahkan ke direktori tingkat atas di repositori kloning.
	`cd amazon-reverse-engineer-dynamodb`
	2. Masukkan perintah berikut untuk menginstal pustaka dan paket yang diperlukan.
	`pip install -r requirements.txt`
		Pengembang aplikasi

Hasilkan model PynamoDB dan fungsi CRUD

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah file skema JSON.	1. Arahkan ke direktori tingkat atas di repositori kloning. `cd amazon-reverse-engineer-dynamodb` 2. Buka `test.json` file di editor pilihan Anda. Anda dapat menggunakan file ini sebagai referensi untuk membuat file skema JSON Anda sendiri, atau Anda dapat memperbarui nilai dalam file ini agar sesuai dengan lingkungan Anda. 3. Ubah nama, Wilayah AWS, dan atribut nilai untuk tabel DynamoDB target Anda. **Note** Jika Anda mendefinisikan tabel yang tidak ada dalam file skema JSON, solusi ini tidak menghasilkan model atau fungsi CRUD untuk tabel itu. 4. Simpan dan tutup file `test.json` . Kami menyarankan Anda	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menyimpan file ini dengan nama baru.	
Jalankan aplikasi Python.	Masukkan perintah berikut untuk menghasilkan model PynamoDB dan fungsi CRUD, `<input_schema.json>` di mana nama file skema JSON Anda. `python main.py --file <input_schema.json>`	Pengembang aplikasi

Verifikasi model PynamoDB dan fungsi CRUD

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi model PynamodB yang dihasilkan.	1. Di direktori tingkat atas dari repositori kloning, masukkan perintah berikut untuk menavigasi ke repositori. `models` `cd models` 2. Secara default, solusi ini menamai file model PynamoDB. `demo_model.py` Validasi bahwa file ini ada.	Pengembang aplikasi
Verifikasi fungsi CRUD yang dihasilkan.	1. Di direktori tingkat atas dari repositori kloning, masukkan perintah berikut	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk menavigasi ke repositori. `crud` `cd crud` 2. Secara default, solusi ini memberi nama skrip`demo_crud.py` . Validasi bahwa file ini ada. 3. Gunakan kelas Python dalam `demo_crud.py` file untuk melakukan operasi CRUD pada tabel DynamoDB target. Konfirmasikan bahwa operasi selesai dengan sukses.	

Sumber daya terkait

- Komponen inti Amazon DynamoDB (dokumentasi DynamoDB)
- Meningkatkan akses data dengan indeks sekunder (dokumentasi DynamoDB)

Informasi tambahan

Contoh atribut untuk file skema JSON

```
[
{
"name": "test_table",
"region": "ap-south-1",
"attributes": [
{
"name": "id",
"type": "UnicodeAttribute"
},
```

```
{
"name": "name",
"type": "UnicodeAttribute"
},
{
"name": "age",
"type": "NumberAttribute"
}
]
}
]
```

Blokir akses publik ke Amazon RDS dengan menggunakan Cloud Custodian

abhay kumar dan Dwarika Patra, Amazon Web Services

Ringkasan

Banyak organisasi menjalankan beban kerja dan layanan mereka di beberapa vendor cloud. Dalam lingkungan cloud hybrid ini, infrastruktur cloud membutuhkan tata kelola cloud yang ketat, selain keamanan yang disediakan oleh penyedia cloud individu. Database cloud seperti Amazon Relational Database Service (Amazon RDS) adalah salah satu layanan penting yang harus dipantau untuk setiap kerentanan akses dan izin. Meskipun Anda dapat membatasi akses ke database Amazon RDS dengan mengonfigurasi grup keamanan, Anda dapat menambahkan lapisan perlindungan kedua untuk melarang tindakan seperti akses publik. Memblokir akses publik membantu Anda dengan Peraturan Perlindungan Data Umum (GDPR), Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA), Institut Standar dan Teknologi Nasional (NIST), dan Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS).

Cloud Custodian adalah mesin aturan sumber terbuka yang dapat Anda gunakan untuk memberlakukan pembatasan akses untuk sumber daya Amazon Web Services (AWS) seperti Amazon RDS. Dengan Cloud Custodian, Anda dapat menetapkan aturan yang memvalidasi lingkungan berdasarkan standar keamanan dan kepatuhan yang ditetapkan. Anda dapat menggunakan Cloud Custodian untuk mengelola lingkungan cloud Anda dengan membantu memastikan kepatuhan terhadap kebijakan keamanan, kebijakan tag, dan pengumpulan sampah sumber daya dan manajemen biaya yang tidak digunakan. Dengan Cloud Custodian, Anda dapat menggunakan satu antarmuka untuk menerapkan tata kelola di lingkungan cloud hybrid. Misalnya, Anda dapat menggunakan antarmuka Cloud Custodian untuk berinteraksi dengan AWS Microsoft Azure, mengurangi upaya bekerja dengan mekanisme seperti AWS Config, grup AWS keamanan, dan kebijakan Azure.

Pola ini memberikan instruksi untuk menggunakan Cloud Custodian on AWS untuk memberlakukan pembatasan aksesibilitas publik pada instans Amazon RDS.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS

- Sebuah key pair
- AWS Lambda dipasang

Arsitektur

Diagram berikut menunjukkan Cloud Custodian yang menerapkan kebijakan ke AWS Lambda, AWS CloudTrail memulai `CreateDBInstance` acara, dan setelan fungsi Lambda ke `PubliclyAccessible` false di Amazon RDS.

Alat

Layanan AWS

- AWS CloudTrailmembantu Anda mengaudit tata kelola, kepatuhan, dan risiko operasional Anda Akun AWS.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud

Alat lainnya

- Cloud Custodian menyatukan alat dan skrip yang digunakan banyak organisasi untuk mengelola akun cloud publik mereka menjadi satu alat open source. Ini menggunakan mesin aturan stateless untuk definisi dan penegakan kebijakan, dengan metrik, output terstruktur, dan pelaporan terperinci untuk infrastruktur cloud. Ini terintegrasi erat dengan runtime tanpa server untuk memberikan remediasi dan respons waktu nyata dengan overhead operasional yang rendah.

Epik

Mengatur AWS CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS CLI.	Untuk menginstal AWS CLI, ikuti instruksi dalam AWS dokumentasi.	Administrator AWS

Siapkan AWS kredensil.	Konfigurasikan pengaturan yang AWS CLI digunakan untuk berinteraksi AWS, termasuk Wilayah AWS dan format output yang ingin Anda gunakan.
	`$>aws configure`
	`AWS Access Key ID [None]: <your_access_key_id>`
	`AWS Secret Access Key [None]: <your_secret_access_key>`
	`Default region name [None]:`
	`Default output format [None]:`
	Lihat informasi yang lebih lengkap dalam dokumentasi AWS.
		Administrator AWS

Buat peran IAM.	Untuk membuat peran IAM dengan peran eksekusi Lambda, jalankan perintah berikut. `aws iam create-role --role-name lambda-`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ex --assume-role-poli cy-document '{"Versio n": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": {"Service ": "lambda.amazonaws. com"}, "Action": "sts:AssumeRole"}]}`	

Siapkan Cloud Custodian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Cloud Custodian.	Untuk menginstal Cloud Custodian untuk sistem operasi dan lingkungan Anda, ikuti petunjuk dalam dokumentasi Cloud Custodian.	DevOps insinyur
Periksa skema Cloud Custodian.	Untuk melihat daftar lengkap sumber daya Amazon RDS yang dapat digunakan untuk menjalankan kebijakan, gunakan perintah berikut. `custodian schema aws.rds`	DevOps insinyur
Buat kebijakan Cloud Custodian.	Simpan kode yang ada di bawah file kebijakan Cloud Custodian di bagian Informasi tambahan menggunakan ekstensi YAMM.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan tindakan Cloud Custodian untuk mengubah flag yang dapat diakses publik.	1. Temukan kode kustodian (misalnya,`/Users/abcd/custodian/lib/python3.9/site-packages/c7n/resources/rds.py`). 2. Temukan `RDSSetPublicAvailability` kelas di`rds.py`, dan modifikasi kelas ini dengan menggunakan kode yang ada di bawah file rds.py sumber daya c7n di bagian Informasi tambahan.	DevOps insinyur
Lakukan lari kering.	(Opsional) Untuk memeriksa sumber daya mana yang diidentifikasi oleh kebijakan tanpa menjalankan tindakan apa pun pada sumber daya, gunakan perintah berikut. `custodian run -dryrun <policy_name>.yaml -s <output_directory>`	DevOps insinyur

Terapkan kebijakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan kebijakan dengan menggunakan Lambda.	Untuk membuat fungsi Lambda yang akan menjalankan kebijakan, gunakan perintah berikut.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`custodian run -s policy.yaml` Kebijakan ini kemudian akan diprakarsai oleh AWS CloudTrail `CreateDBInstance` acara tersebut. Akibatnya, AWS Lambda akan mengatur flag yang dapat diakses publik `false` untuk instance yang sesuai dengan kriteria.	

Sumber daya terkait

- AWS Lambda situs web
- Situs Amazon RDS
- Dokumentasi Cloud Custodian

Informasi tambahan

File YAMM kebijakan Cloud Custodian

```
policies:
  - name: "block-public-access"
    resource: rds
    description: |
      This Enforcement blocks public access for RDS instances.
    mode:
      type: cloudtrail
      events:
        - event: CreateDBInstance # Create RDS instance cloudtrail event
          source: rds.amazonaws.com
          ids: requestParameters.dBInstanceIdentifier
      role: arn:aws:iam::1234567890:role/Custodian-compliance-role
```

```
    filters:
      - type: event
        key: 'detail.requestParameters.publiclyAccessible'
        value: true
    actions:
      - type: set-public-access
        state: false
```

file rds.py sumber daya c7n

```
@actions.register('set-public-access')
 class RDSSetPublicAvailability(BaseAction):

     schema = type_schema(
         "set-public-access",
         state={'type': 'boolean'})
     permissions = ('rds:ModifyDBInstance',)

     def set_accessibility(self, r):
         client = local_session(self.manager.session_factory).client('rds')
         waiter = client.get_waiter('db_instance_available')
         waiter.wait(DBInstanceIdentifier=r['DBInstanceIdentifier'])
         client.modify_db_instance(
             DBInstanceIdentifier=r['DBInstanceIdentifier'],
             PubliclyAccessible=self.data.get('state', False))


     def process(self, rds):
         with self.executor_factory(max_workers=2) as w:
             futures = {w.submit(self.set_accessibility, r): r for r in rds}
             for f in as_completed(futures):
                 if f.exception():
                     self.log.error(
                         "Exception setting public access on %s  \n %s",
                         futures[f]['DBInstanceIdentifier'], f.exception())
         return rds
```

Integrasi CSPM Security Hub

Cloud Custodian dapat diintegrasikan AWS Security Hub CSPMuntuk mengirim temuan keamanan dan mencoba tindakan remediasi. Untuk informasi selengkapnya, lihat Mengumumkan Integrasi Cloud Custodian dengan. AWS Security Hub CSPM

Konfigurasikan otentikasi Windows untuk Amazon RDS untuk Microsoft SQL Server menggunakan AWS Managed Microsoft AD

Ramesh Babu Donti, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara mengonfigurasi otentikasi Windows untuk Amazon Relational Database Service (Amazon RDS) untuk instance SQL Server menggunakan (). AWS Directory Service for Microsoft Active Directory AWS Managed Microsoft AD Otentikasi Windows memungkinkan pengguna untuk terhubung ke instance RDS menggunakan kredensyal domain mereka alih-alih nama pengguna dan kata sandi khusus database.

Anda dapat mengaktifkan otentikasi Windows baik saat membuat database RDS SQL Server baru atau dengan menambahkannya ke instance database yang ada. Instance database terintegrasi dengan AWS Managed Microsoft AD untuk menyediakan otentikasi terpusat dan otorisasi bagi pengguna domain yang mengakses database SQL Server.

Konfigurasi ini meningkatkan keamanan dengan memanfaatkan infrastruktur Active Directory yang ada dan menghilangkan kebutuhan untuk mengelola kredensi database terpisah untuk pengguna domain.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan izin yang sesuai
- Virtual Private Cloud (VPC) dengan berikut ini:
  - Gateway Internet dan tabel rute yang dikonfigurasi
  - Gateway NAT di subnet publik (jika akses internet diperlukan untuk instance)
- AWS Identity and Access Management Peran (IAM):
  - Peran domain dengan kebijakan AWS terkelola berikut:
    - `AmazonSSMManagedInstanceCore`untuk mengaktifkan AWS Systems Manager
    - `AmazonSSMDirectoryServiceAccess`untuk memberikan izin untuk menggabungkan instance ke direktori
  - Peran pemantauan yang ditingkatkan RDS (jika pemantauan yang ditingkatkan diaktifkan)

- Grup keamanan:
  - Grup keamanan layanan direktori untuk memungkinkan port komunikasi Active Directory
  - Grup keamanan Amazon Elastic Compute Cloud (Amazon EC2) untuk memungkinkan komunikasi RDP 3389 dan domain
  - Grup keamanan RDS untuk mengizinkan port SQL Server 1433 dari sumber resmi
- Konektivitas jaringan:
  - Resolusi DNS yang tepat dan konektivitas jaringan antar subnet

Batasan

- Untuk informasi tentang dukungan Wilayah AWS tersebut AWS Managed Microsoft AD dengan RDS untuk SQL Server, lihat Ketersediaan wilayah dan versi.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Tumpukan teknologi sumber

- Direktori Aktif lokal atau AWS Managed Microsoft AD

Tumpukan teknologi target

- Amazon EC2
- Amazon RDS for Microsoft SQL Server
- AWS Managed Microsoft AD

Arsitektur target

Arsitekturnya meliputi:

- Peran IAM yang menggabungkan EC2 instans Amazon ke domain. AWS Managed Microsoft AD
- Instans Amazon EC2 Windows untuk administrasi dan pengujian database.

- VPC Amazon dengan subnet pribadi untuk meng-host instans Amazon RDS dan sumber daya internal di seluruh Availability Zone.
- Grup keamanan untuk kontrol akses jaringan:
  - Grup keamanan Amazon RDS untuk mengontrol akses masuk ke port SQL Server `1433` dari sumber resmi.
  - Grup EC2 keamanan Amazon untuk mengelola akses RDP melalui port `3389` dan port komunikasi domain.
  - Grup keamanan Layanan Direktori untuk komunikasi Active Directory melalui port`53,88,389`, dan`445`.
- AWS Managed Microsoft AD untuk menyediakan layanan otentikasi dan otorisasi terpusat untuk sumber daya Windows.
- Contoh database Amazon RDS for SQL Server di subnet pribadi dengan otentikasi Windows diaktifkan.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud
- AWS Directory Servicemenyediakan beberapa cara untuk menggunakan Microsoft Active Directory (AD) dengan yang lain Layanan AWS seperti Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS) untuk SQL Server, dan FSx Amazon untuk Windows File Server.
- AWS Directory Service for Microsoft Active Directorymemungkinkan beban kerja dan sumber daya yang sadar direktori Anda AWS untuk menggunakan Microsoft Active Directory di file. AWS Cloud
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Layanan lainnya

- Microsoft SQL Server Management Studio (SSMS) adalah alat untuk mengelola SQL Server, termasuk mengakses, mengkonfigurasi, dan mengelola komponen SQL Server.

Praktik terbaik

- Untuk praktik terbaik umum, lihat Praktik terbaik untuk Amazon RDS.

Epik

Konfigurasikan AWS Managed Microsoft AD

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan jenis direktori.	1. Dari Konsol Manajemen AWS, navigasikan ke AWS Directory Service. 2. Pilih Siapkan direktori. 3. Pilih AWS Managed Microsoft ADuntuk jenis direktori. 4. Pilih Buat domain AD AWS terkelola baru, lalu pilih Berikutnya.	DBA, DevOps insinyur
Konfigurasikan informasi direktori.	Di bagian Informasi direktori , masukkan informasi yang diperlukan, dan simpan nilai opsional: 1. Di bawah Edisi, pilih edisi yang memenuhi persyaratan Anda. 2. Di bawah nama DNS Direktori, masukkan nama domain yang sepenuhnya memenuhi syarat (FQDN).	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di bawah Kata sandi admin, tetapkan kata sandi untuk akun administrator, lalu pilih Berikutnya.	
Konfigurasikan VPC dan subnet.	1. Di bawah Networking, pilih VPC target (minimal, Anda harus mengatur dua subnet secara terpisah). AWS Zona Ketersediaan 2. Di bawah Jenis jaringan, pilih IPv4 saja. 3. Di bawah Subnet, pilih dua subnet pribadi secara terpisah AWS Zona Ketersediaan, lalu pilih Berikutnya.	DBA, DevOps insinyur
Tinjau dan buat direktori.	1. Tinjau nilai konfigurasi, lalu pilih Buat direktori. 2. Tunggu status direktori berubah menjadi Aktif.	DBA, DevOps insinyur

Membuat dan mengonfigurasi EC2 instans Amazon

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan AMI untuk Windows.	1. Dari Konsol Manajemen AWS, navigasikan ke EC2. 2. Pilih Luncurkan instans. 3. Di bawah Nama dan tag, masukkan nama dan tag apa pun yang berlaku.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pilih Amazon Machine Image (AMI) untuk Windows Server yang memenuhi kebutuhan Anda. 5. Di bawah Jenis instans, pilih tipe berukuran tepat. 6. Di bawah Key pair (login), pilih key pair yang ada atau buat yang baru.	
Mengkonfigurasi pengaturan jaringan.	1. Di bawah Pengatura n jaringan, pilih VPC yang sama dengan yang digunakan. AWS Directory Service 2. Pilih subnet pribadi. 3. Di bawah Firewall (grup keamanan), buat grup yang memungkinkan port RDP 3389 dan komunikasi domain.	DBA, DevOps insinyur
Konfigurasikan penyimpanan.	Konfigurasikan volume Amazon EBS sesuai kebutuhan.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan detail lanjutan dan luncurkan instance.	1. Perluas bagian Detail lanjutan. 2. Untuk direktori Gabung Domain, pilih yang dibuat sebelumnya AWS Managed Microsoft AD. 3. Untuk profil instans IAM, pilih peran dengan kebijakan `AmazonSSMManagedInstanceCore` dan `AmazonSSMDirectoryServiceAccess`. 4. Tinjau semua nilai konfigurasi, lalu pilih Launch instance.	DBA, DevOps insinyur

Buat dan konfigurasikan RDS untuk SQL Server dengan otentikasi Windows

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database dan konfigurasikan opsi mesin.	1. Arahkan ke konsol Aurora dan RDS, dan pilih Buat database. 2. Di bawah Opsi mesin, pilih Microsoft SQL Server. 3. Untuk Jenis manajemen basis data, pilih Amazon RDS. 4. Untuk Edition, pilih SQL Server yang memenuhi kebutuhan Anda.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Untuk versi Engine, pilih versi terbaru yang didukung.	
Pilih templat.	Pilih contoh template yang memenuhi kebutuhan Anda.	DBA, DevOps insinyur
Konfigurasikan pengaturan basis data.	1. Di bagian Pengaturan di bawah pengidentifikasi instans DB, masukkan nama unik. 2. Di bawah nama pengguna Master, konfigurasikan kredensyal administrator. 3. Di bawah Manajemen kredensyal, pilih Dikelola di AWS Secrets Manager atau Dikelola sendiri.	DBA, DevOps insinyur
Konfigurasikan instance.	Di bagian konfigurasi Instans, di bawah kelas instans DB, pilih ukuran instans yang memenuhi persyaratan Anda.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan penyimpanan.	1. Di bagian Penyimpanan, di bawah Jenis penyimpanan, pilih jenis yang memenuhi kebutuhan Anda. Kami merekomendasikan gp3, io1, atau io2. 2. Tetapkan nilai awal seperti yang diperlukan untuk penyimpanan yang dialokasikan, IOPS yang Disediakan, dan throughput Penyimpanan. 3. (Opsional) Perluas bagian Konfigurasi penyimpan an tambahan, dan pilih Aktifkan penskalaan otomatis penyimpanan.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan konektivitas.	1. Di bagian Konektivitas, pilih apakah Anda ingin mengatur koneksi ke sumber daya komputasi untuk database. 2. Untuk VPC, pilih VPC yang sama dengan yang dimiliki. AWS Directory Service 3. Untuk grup subnet DB, pilih grup yang mencakup beberapa Availability Zone. 4. Untuk akses publik, pilih No. 5. Untuk grup keamanan VPC (firewall), pilih grup yang ada atau buat grup baru yang memungkinkan akses melalui port SQL Server. `1433` 6. Pilih Availability Zone pilihan Anda. 7. Perluas bagian Konfigurasi tambahan, dan pilih apakah Anda ingin menggunakan port database khusus.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan otentikasi Windows.	1. Di bagian otentikasi Microsoft SQL Server Windows, pilih kotak centang Aktifkan otentikasi Microsoft SQL Server Windows. 2. Untuk jenis otentikasi Windows, pilih AWS Managed Microsoft AD. 3. Untuk Direktori, pilih Browse directory, dan pilih AWS Managed Microsoft AD.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pemantauan.	1. Di bagian Pemantauan, pilih wawasan basis data standar atau lanjutan. 2. Di bawah Wawasan kinerja, pilih kotak centang Aktifkan wawasan kinerja. 3. Pilih periode dan AWS KMS kunci retensi. 4. Di bawah Pengaturan pemantauan tambahan, pilih kotak centang Pemantauan yang ditingkat kan. 5. (Opsional) Di bawah Ekspor log, pilih kotak centang Kesalahan log. Catatan: Metrik berguna saat Anda ingin melihat bagaimana proses atau utas yang berbeda menggunak an CPU. Anda juga dapat mengekspor log kesalahan ke Amazon CloudWatch jika log kesalahan diaktifkan.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan tambahan.	1. Perluas bagian Konfigurasi tambahan. 2. Untuk grup parameter DB, dan grup opsi DB, pilih nilai default atau kustom. 3. Tetapkan zona waktu pilihan Anda. 4. Untuk Collation, tetapkan nilai. Nilai default-nya `SQL_Latin1_General_CP1_CI_AS` . 5. Di bawah Backup: • Pilih kotak centang Aktifkan pencadangan otomatis. Ini menciptakan snapshot database. • Untuk periode retensi Backup, pilih jumlah hari yang diperlukan. • Untuk jendela Backup, pilih nilai. • (Opsional) Untuk replikasi Backup, pilih Aktifkan replikasi di tempat lain. Wilayah AWS • Pilih kotak centang Aktifkan enkripsi untuk mengenkripsi instance menggunakan. AWS KMS 6. Di bawah jendela Pemeliharaan, pilih kotak centang	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pilih jendela, dan atur waktu yang diinginkan. 7. Pilih kotak centang Aktifkan perlindungan penghapusan.	
Tinjau biaya dan buat database.	Tinjau bagian Perkiraan biaya bulanan, lalu pilih Buat database.	DBA, DevOps insinyur

Konfigurasikan akses database dan uji koneksi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke mesin Windows.	Connect ke mesin Windows Anda dan luncurkan SQL Server Management Studio. 1. Gunakan RDP untuk terhubung ke mesin Windows Anda menggunakan kredensil AWS Managed Microsoft AD 2. Luncurkan SSMS dengan memasukkan SSMS di menu Start, dan pilih SQL Server Management Studio.	DBA, DevOps insinyur
Konfigurasikan koneksi SSMS.	Siapkan koneksi database menggunakan otentikasi Windows. 1. Ketika kotak dialog Connect to server muncul (atau dengan menavigasi ke Object explorer, Connect,	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mesin Database), atur tipe Server ke mesin Database. 2. Masukkan titik akhir RDS SQL Server (misalnya,) `your-rds-instance.region.rds.amazonaws.com` 3. Pilih otentikasi Windows.	
Konfigurasikan pengaturan keamanan.	Tetapkan parameter keamanan yang diperlukan untuk SSMS versi 20 atau yang lebih baru. 1. Di tab Properti koneksi, atur Enkripsi ke Wajib. 2. Pilih kotak centang Trust server certificate. 3. Biarkan nama Host di bidang sertifikat kosong. 4. (Opsional) Atur nama database dan sesuaikan batas waktu koneksi sesuai kebutuhan.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat login Windows.	1. Siapkan dan uji otentikasi Windows untuk pengguna domain.
	2. Untuk membuat koneksi awal, pilih Connect.
	3. Di jendela kueri, jalankan yang berikut ini:
	`CREATE LOGIN [<domainName>\<user_name>] FROM WINDOWS;`
	`GO`
		DBA, DevOps insinyur

Uji otentikasi Windows.	1. Keluar dari EC2 instans Amazon. 2. Masuk kembali ke EC2 instance menggunakan kredensi domain Anda. 3. Luncurkan SSMS. 4. Connect menggunakan Windows Authentication. 5. Verifikasi bahwa koneksi berhasil.	DBA, DevOps insinyur

Sumber daya terkait

- Membuat Anda AWS Managed Microsoft AD
- Bergabung dengan instans Amazon EC2 Windows ke AWS Managed Microsoft AD
- Wilayah dan mesin DB yang Didukung untuk otentikasi Kerberos di Amazon RDS
- Apa itu SQL Server Management Studio (SSMS)?

Mengonfigurasikan akses lintas akun ke Amazon DynamoDB

Shashi Dalmia, Imhoertha Ojior, dan Esteban Serna Parra, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah untuk mengonfigurasi akses lintas akun ke Amazon DynamoDB dengan menggunakan kebijakan berbasis sumber daya. Untuk beban kerja yang menggunakan DynamoDB, semakin umum menggunakan strategi isolasi beban kerja untuk meminimalkan ancaman keamanan dan memenuhi persyaratan kepatuhan. Menerapkan strategi isolasi beban kerja seringkali memerlukan akses lintas akun dan lintas wilayah ke sumber daya DynamoDB dengan menggunakan kebijakan berbasis identitas (IAM). AWS Identity and Access Management Ini melibatkan pengaturan izin IAM dan membangun hubungan kepercayaan antara. Akun AWS

Kebijakan berbasis sumber daya untuk DynamoDB sangat menyederhanakan postur keamanan untuk beban kerja lintas akun. Pola ini menyediakan langkah-langkah dan kode sampel untuk menunjukkan bagaimana Anda dapat mengkonfigurasi AWS Lambda fungsi dalam satu Akun AWS untuk menulis data ke tabel database DynamoDB di akun yang berbeda.

Prasyarat dan batasan

Prasyarat

- Dua aktif Akun AWS. Pola ini mengacu pada akun ini sebagai Akun A dan Akun B.
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi untuk mengakses Akun A, untuk membuat tabel DynamoDB. Langkah-langkah lain dalam pola ini memberikan instruksi untuk menggunakan konsol IAM, DynamoDB, dan Lambda. Jika Anda berencana untuk menggunakannya, AWS CLI konfigurasikan untuk mengakses kedua akun.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan arsitektur akun tunggal. AWS Lambda, Amazon Elastic Compute Cloud (Amazon EC2), dan DynamoDB semuanya berada di akun yang sama. Dalam skenario ini, fungsi Lambda dan EC2 instans Amazon dapat mengakses DynamoDB. Untuk memberikan akses ke tabel DynamoDB, Anda dapat membuat kebijakan berbasis identitas di IAM, atau Anda dapat membuat kebijakan berbasis sumber daya di DynamoDB.

Diagram berikut menunjukkan arsitektur multi-akun. Jika sumber daya dalam satu Akun AWS memerlukan akses ke tabel DynamoDB di akun lain, Anda perlu menyiapkan kebijakan berbasis sumber daya di DynamoDB untuk memberikan akses yang diperlukan. Misalnya, dalam diagram berikut, akses ke tabel DynamoDB di Akun A diberikan ke fungsi Lambda di Akun B dengan menggunakan kebijakan berbasis sumber daya.

Pola ini menjelaskan akses lintas akun antara Lambda dan DynamoDB. Anda dapat menggunakan langkah serupa untuk yang lain Layanan AWS jika izin yang sesuai dikonfigurasi di kedua akun. Misalnya, jika ingin memberikan akses fungsi Lambda ke bucket Amazon Simple Storage Service (Amazon S3) di Akun A, Anda dapat membuat kebijakan berbasis sumber daya di Amazon S3 dan menambahkan izin ke peran eksekusi Lambda di Akun B.

Alat

Layanan AWS

- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Kode

Pola ini mencakup kode sampel di bagian Informasi tambahan untuk menunjukkan bagaimana Anda dapat mengonfigurasi fungsi Lambda di Akun B untuk menulis ke tabel DynamoDB di Akun A. Kode

disediakan hanya untuk tujuan ilustrasi dan pengujian. Jika Anda menerapkan pola ini di lingkungan produksi, gunakan kode sebagai referensi, dan sesuaikan untuk lingkungan Anda sendiri.

Praktik terbaik

- Ikuti praktik terbaik untuk kebijakan berbasis sumber daya dalam dokumentasi DynamoDB.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Buat kebijakan dan peran IAM untuk fungsi Lambda di Akun B

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan di Akun B.	Kebijakan IAM ini memungkinkan PutItemtindakan untuk tabel DynamoDB di Akun A. 1. Masuk ke Akun B di Konsol Manajemen AWS. 2. Buka konsol IAM. 3. Di panel navigasi, pilih Kebijakan dan kemudian pilih Buat kebijakan. 4. Pada halaman Tentukan izin, untuk editor kebijakan, pilih JSON. 5. Masukkan kebijakan berikut. `{ "Version": "2012-10-17", "Statement": [ {`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Sid": "Statemen t1",
	"Effect": "Allow",
	"Action": "dynamodb:PutItem",
	"Resource": "arn:aws:dynamodb: <Region>:<Account- A-ID>:table/Table- Account-A"
	}
	]
	}</pre>
	6. Ganti `<Region>` dan `<Account-A-ID>` dengan nilai Anda, lalu pilih Berikutnya.
	7. Untuk nama Kebijakan , masukkan nama unik untuk kebijakan Anda, sepertiDynamoDB-PutItem-Policy .
	8. (Opsional) Tambahkan deskripsi kebijakan.
	9. Pilih Buat kebijakan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran di Akun B.	Fungsi Lambda di Akun B menggunakan peran IAM ini untuk mengakses tabel DynamoDB di Akun A. 1. Buka konsol IAM. 2. Di panel navigasi, pilih Peran, lalu pilih Buat peran. 3. Untuk Pilih entitas tepercaya, pilih Layanan AWS. 4. Di bagian Use case, pilih Lambda. 5. Pilih Berikutnya: Izin. 6. Di kotak Filter kebijakan, masukkan DynamoDB. 7. Dalam daftar kebijakan DynamoDB, pilih. `DynamoDB-PutItem-Policy` 8. Kosongkan kotak Filter kebijakan, lalu masukkan Lambda. 9. Dalam daftar kebijakan Lambda, pilih AWSLambda Jalankan. 10. Pilih Berikutnya: Nama, tinjau, dan buat. 11. Untuk nama Peran, masukkan nama unik untuk peran Anda, seperti `DynamoDB-PutItemAccess` .	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	12. (Opsional) Tambahkan deskripsi peran. 13. (Opsional) Tambahkan metadata ke peran dengan melampirkan tag sebagai pasangan nilai kunci. 14. Pilih Buat peran. Untuk informasi selengkapnya tentang membuat peran, lihat dokumentasi IAM.	
Perhatikan ARN peran.	1. Buka konsol IAM. 2. Di panel navigasi, pilih Peran. 3. Di kotak pencarian, masukkan `DynamoDB-PutItemAccess`, lalu pilih peran. 4. Pada halaman ringkasan untuk peran tersebut, salin Nama Sumber Daya Amazon (ARN). Anda menggunakan ARN saat Anda mengatur fungsi Lambda.	AWS Umum

Buat tabel DynamoDB di Akun A

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat tabel DynamoDB.	Gunakan AWS CLI perintah berikut untuk membuat tabel DynamoDB.
	
	`aws dynamodb create-table \`
	`--table-name Table-Account-A \`
	`--attribute-definitions \`
	`Attribute Name=category,AttributeType=S \`
	`Attribute Name=item,AttributeType=S \`
	`--key-schema \`
	`Attribute Name=category,KeyType=HASH \`
	`Attribute Name=item,KeyType=RANGE \`
	`--provisioned-throughput \`
	`ReadCapacityUnits=5,WriteCapacityUnits=5 \`
	`--resource-policy \`
	`'{`
	`"Version": "2012-10-17",`
	`"Statement": [`
	`{`
	`"Sid": "Statement1",`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Effect": "Allow",
	"Principal": {
	"AWS": "arn:aws:iam::<Account-B-ID>:role/<Role-Name>"
	},
	"Action": "dynamodb:PutItem",
	"Resource": "arn:aws:dynamodb:<Region>:<Account-A-ID>:table/Table-Account-A"
	}
	
	]
	}'</pre>
	Ganti yang berikut ini dalam contoh kode ini:
	• <Account-B-ID> adalah ID Akun B.
	• <Role-Name> adalah nama peran IAM yang Anda buat, sepertiDynamoDB-PutItemAccess .
	• <Region>adalah Wilayah AWS tempat Anda membuat tabel DynamoDB.
	• <Account-A-ID> adalah ID Akun A.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Anda menentukan konfigurasi kebijakan berbasis sumber daya dalam `create-table` pernyataan menggunakan tanda. `--resource-policy` Kebijakan ini mengacu pada ARN untuk tabel DynamoDB di Akun A. Untuk informasi selengkapnya tentang membuat tabel, lihat dokumentasi DynamoDB.	

Buat fungsi Lambda di Akun B

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda untuk menulis data ke DynamoDB.	1. Masuk ke Akun B di Konsol Manajemen AWS. 2. Buka Konsol Lambda. 3. Di panel navigasi, pilih Fungsi, lalu pilih Buat fungsi. 4. Untuk Nama, masukkan `lambda_write_function` .	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Untuk Runtime, pilih Python 3.8 atau yang lebih baru. 6. Di bawah Ubah peran eksekusi default, pilih Gunakan peran yang ada. 7. Untuk peran yang ada, pilih peran IAM yang Anda buat, seperti`DynamoDB-PutItemAccess`. 8. Pilih Buat fungsi. 9. Di tab Kode, tempel kode sampel yang disediakan di bagian Informasi tambahan dari pola ini. Ganti yang berikut ini dalam contoh kode ini: • `<Account-A-ID>`adalah ID Akun A. • `<Region>`adalah Wilayah AWS tempat Anda membuat tabel DynamoDB. 10. Pilih Deploy. 11. Pilih Uji. Ini meminta Anda untuk mengonfigurasi acara pengujian. Buat acara baru dengan nama pilihan Anda, seperti`MyTestEventForWrite`, lalu simpan konfigurasinya. 12. Pilih UjiLagi. Ini menjalankan fungsi Lambda dengan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nama acara yang Anda berikan. 13. Periksa output dari fungsi. Ini harus menunjukkan bahwa fungsi mengakses tabel DynamoDB di Akun A dan mampu menulis data untuk itu. Untuk informasi selengkapnya tentang membuat fungsi Lambda, lihat dokumentasi Lambda.	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya.	Untuk menghindari timbulnya biaya yang terkait dengan sumber daya yang dibuat dalam pola ini, lakukan hal berikut untuk menghapus sumber daya ini: 1. Di Akun B, hapus fungsi Lambda yang Anda buat untuk terhubung ke DynamoDB. Untuk petunjuk, lihat dokumentasi Lambda. 2. Di Akun A, hapus tabel DynamoDB yang Anda	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	buat. Untuk petunjuk, lihat dokumentasi DynamoDB. 3. Untuk praktik terbaik keamanan, hapus kebijakan IAM (`DynamoDB-PutItem-Policy`) saat tidak lagi diperlukan. Untuk informasi selengkapnya, lihat dokumentasi IAM. 4. Untuk praktik terbaik keamanan, hapus peran IAM (`DynamoDB-PutItemAccess`) saat tidak lagi diperlukan. Untuk informasi selengkapnya, lihat dokumentasi IAM.	

Pemecahan masalah

Isu	Solusi
Saat membuat fungsi Lambda, Anda menerima kesalahan. `ResourceNotFoundException`	Konfirmasikan bahwa Anda telah memasukkan Wilayah AWS dan ID Akun A. Ini adalah bagian dari ARN untuk tabel DynamoDB.

Sumber daya terkait

- Memulai dengan DynamoDB (dokumentasi DynamoDB)
- Memulai dengan Lambda (dokumentasi Lambda)
- Menggunakan kebijakan berbasis sumber daya untuk DynamoDB (dokumentasi DynamoDB)
- Membuat kebijakan IAM (dokumentasi IAM)
- Logika evaluasi kebijakan lintas akun (dokumentasi IAM)

- Referensi elemen kebijakan IAM JSON (dokumentasi IAM)

Informasi tambahan

Kode sampel

```
import boto3
from datetime import datetime

dynamodb_client = boto3.client('dynamodb')

def lambda_handler(event, context):
     now = datetime.now().isoformat()
     data = dynamodb_client.put_item(TableName='arn:aws:dynamodb:<Region>:<Account-
A-ID>:table/Table-Account-A', Item={"category": {"S": "Fruit"},"item": {"S":
 "Apple"},"time": {"S": now}})
     return data
```

> **Note**
>
> Ketika klien DynamoDB dipakai, ARN dari tabel DynamoDB disediakan bukan nama tabel. Ini diperlukan agar fungsi Lambda terhubung ke tabel DynamoDB yang benar saat dijalankan.

Konfigurasikan perutean hanya-baca dalam grup ketersediaan Selalu Aktif di SQL Server di AWS

Subhani Shaik, Amazon Web Services

Ringkasan

Pola ini mencakup cara menggunakan replika sekunder siaga di SQL Server Always On dengan menurunkan beban kerja read-only dari replika primer ke replika sekunder.

Pencerminan basis data memiliki one-to-one pemetaan. Anda tidak dapat membaca database sekunder secara langsung, jadi Anda harus membuat snapshot. Fitur grup ketersediaan Selalu Aktif diperkenalkan di Microsoft SQL Server 2012. Dalam versi yang lebih baru, fungsionalitas utama telah diperkenalkan, termasuk perutean hanya-baca. Di grup ketersediaan Selalu Aktif, Anda dapat membaca data langsung dari replika sekunder dengan mengubah mode replika menjadi hanya-baca.

Solusi grup ketersediaan Always On mendukung ketersediaan tinggi (HA), pemulihan bencana (DR), dan alternatif untuk pencerminan basis data. Grup ketersediaan Always On bekerja di tingkat database dan memaksimalkan ketersediaan satu set database pengguna.

SQL Server menggunakan mekanisme routing read-only untuk mengarahkan koneksi read-only yang masuk ke replika baca sekunder. Untuk mencapai ini, Anda harus menambahkan parameter dan nilai berikut dalam string koneksi:

- `ApplicationIntent=ReadOnly`
- `Initial Catalog=<database name>`

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan virtual private cloud (VPC), dua Availability Zone, subnet pribadi, dan grup keamanan
- Dua mesin Amazon Elastic Compute Cloud (Amazon EC2) dengan SQL Server 2019 Enterprise Edition Amazon Machine Image dengan Windows Server Failover Clustering (WSFC) yang dikonfigurasi pada tingkat instans dan grup ketersediaan Selalu Aktif yang dikonfigurasi pada tingkat SQL Server antara node primer (`WSFCNODE1`) dan node sekunder (`WSFCNODE2`), yang merupakan bagian dari direktori AWS Directory Service untuk Microsoft Active Directory bernama `tagechtalk.com`

- Satu atau lebih node dikonfigurasi untuk menerima `read-only` dalam replika sekunder
- Pendengar yang diberi nama `SQLAG1` untuk grup ketersediaan Selalu Aktif
- SQL Server Database Engine berjalan dengan akun layanan yang sama pada dua node
- Studio Manajemen SQL Server (SSMS)
- Database uji bernama `test`

Versi Produk

- SQL Server 2014 dan yang lebih baru

Arsitektur

Tumpukan teknologi target

- Amazon EC2
- AWS Dikelola Microsoft AD
- Amazon FSx

Arsitektur target

Diagram berikut menunjukkan bagaimana pendengar Always On availability group (AG) mengalihkan kueri yang berisi `ApplicationIntent` parameter dalam koneksi ke node sekunder yang sesuai.

1. Permintaan dikirim ke pendengar grup ketersediaan Selalu Aktif.
2. Jika string koneksi tidak memiliki `ApplicationIntent` parameter, permintaan dikirim ke instance utama.
3. Jika string koneksi berisi`ApplicationIntent=ReadOnly`, permintaan dikirim ke instance sekunder dengan konfigurasi routing read-only, yaitu WSFC dengan grup ketersediaan Selalu Aktif.

Alat

Layanan AWS

- AWS Directory Service untuk Microsoft Active Directory memungkinkan beban kerja yang sadar direktori dan sumber daya AWS Anda untuk menggunakan Microsoft Active Directory di AWS Cloud.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon FSx menyediakan sistem file yang mendukung protokol konektivitas standar industri dan menawarkan ketersediaan dan replikasi tinggi di seluruh Wilayah AWS.

Layanan lainnya

- SQL Server Management Studio (SSMS) adalah alat untuk menghubungkan, mengelola, dan mengelola instance SQL Server.
- sqlcmd adalah utilitas baris perintah.

Praktik terbaik

Untuk informasi selengkapnya tentang grup ketersediaan Selalu Aktif, lihat dokumentasi SQL Server.

Epik

Mengatur perutean hanya-baca

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui replika menjadi hanya-baca.	Untuk memperbarui replika primer dan sekunder menjadi hanya-baca, sambungkan ke replika utama dari SSMS, dan jalankan kode Langkah 1 dari bagian Informasi tambahan.	DBA
Buat URL perutean.	Untuk membuat URL routing untuk kedua replika, jalankan kode Langkah 2 dari bagian Informasi tambahan. Dalam	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kode ini, `tagechtalk.com` adalah nama direktori AWS Managed Microsoft AD.	
Buat daftar routing.	Untuk membuat daftar routing untuk kedua replika, jalankan kode Langkah 3 dari bagian Informasi tambahan.	DBA
Validasi daftar routing.	Connect ke instance utama dari SQL Server Management Studio, dan jalankan kode Langkah 4 dari bagian Informasi tambahan untuk memvalidasi daftar routing.	DBA

Uji perutean hanya-baca

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect dengan menggunakan ApplicationIntent parameter.	1. Dari SSMS, sambungkan ke nama pendengar grup ketersediaan Selalu Aktif dengan. `ApplicationIntent=ReadOnly; Initial Catalog=test` 2. Koneksi dibuat dengan replika sekunder. Untuk menguji ini, jalankan perintah berikut untuk menunjukkan nama server yang terhubung.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`SELECT SERVERPROPERTY('ComputerNamePhysicalNetBios')` Output akan menampilkan nama replika sekunder saat ini (`WSFCNODE2`).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan failover.	1. Dari SSMS, sambungkan ke nama pendengar grup ketersediaan Selalu Aktif. 2. Verifikasi bahwa database primer dan sekunder sinkron, tanpa kehilangan data. 3. Lakukan failover sehingga replika primer saat ini menjadi replika sekunder, dan replika sekunder menjadi replika utama. 4. Dari SSMS, sambungkan ke nama pendengar grup ketersediaan Selalu Aktif dengan. `ApplicationIntent=ReadOnly; Initial Catalog=test` 5. Koneksi dibuat dengan replika sekunder. Untuk menguji ini, tunjukkan nama server yang terhubung dengan menjalankan perintah berikut. `SELECT SERVERPROPERTY('ComputerNamePhysicalNetBios')` Ini akan menampilkan nama replika sekunder saat ini (`WSFCNODE1`).	DBA

Connect dengan menggunakan utilitas baris perintah sqlcmd

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect dengan menggunak an sqlcmd.	Untuk terhubung dari sqlcmd, jalankan kode Langkah 5 dari bagian Informasi tambahan di prompt perintah. Setelah Anda terhubung, jalankan perintah berikut untuk menunjukkan nama server yang terhubung. `SELECT SERVERPRO PERTY('Computernam ePhysicalNetBios') .` Output akan menampilkan nama replika sekunder saat ini (`WSFCNODE1 `).	DBA

Pemecahan masalah

Isu	Solusi
Membuat pendengar gagal dengan pesan 'Kluster WSFC tidak dapat membawa sumber daya Nama Jaringan online'.	Untuk selengkapnya, lihat posting blog Microsoft Buat Pendengar Gagal dengan Pesan 'Kluster WSFC tidak dapat membawa sumber daya Nama Jaringan online'.
Masalah potensial, termasuk masalah pendengar lainnya atau masalah akses jaringan.	Lihat Memecahkan Masalah Selalu Pada Konfigurasi Grup Ketersediaan (SQL Server) di dokumentasi Microsoft.

Sumber daya terkait

- Konfigurasikan perutean hanya-baca untuk grup ketersediaan Selalu Aktif

- Memecahkan Masalah Selalu Pada Konfigurasi Grup Ketersediaan (SQL Server)

Informasi tambahan

Langkah 1. Perbarui replika menjadi hanya-baca

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH (SECONDARY_ROLE
 (ALLOW_CONNECTIONS = READ_ONLY))
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (SECONDARY_ROLE
 (ALLOW_CONNECTIONS = READ_ONLY))
GO
```

Langkah 2. Buat URL perutean

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH (SECONDARY_ROLE
 (READ_ONLY_ROUTING_URL = N'TCP://WSFCNode1.tagechtalk.com:1433'))
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (SECONDARY_ROLE
 (READ_ONLY_ROUTING_URL = N'TCP://WSFCNode2.tagechtalk.com:1433'))
GO
```

Langkah 3. Buat daftar perutean

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH
 (PRIMARY_ROLE(READ_ONLY_ROUTING_LIST=('WSFCNODE2','WSFCNODE1')));
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (PRIMARY_ROLE
 (READ_ONLY_ROUTING_LIST=('WSFCNODE1','WSFCNODE2')));
GO
```

Langkah 4. Validasi daftar perutean

```
SELECT AGSrc.replica_server_name AS PrimaryReplica, AGRepl.replica_server_name AS
 ReadOnlyReplica, AGRepl.read_only_routing_url AS RoutingURL , AGRL.routing_priority
 AS RoutingPriority FROM sys.availability_read_only_routing_lists AGRL INNER JOIN
 sys.availability_replicas AGSrc ON AGRL.replica_id = AGSrc.replica_id INNER JOIN
 sys.availability_replicas AGRepl ON AGRL.read_only_replica_id = AGRepl.replica_id
 INNER JOIN sys.availability_groups AV ON AV.group_id = AGSrc.group_id ORDER BY
 PrimaryReplica
```

Langkah 5. Utilitas Perintah SQL

```
sqlcmd -S SQLAG1,1433 -E -d test -K ReadOnly
```

Connect dengan menggunakan terowongan SSH di pgAdmin

Jeevan Shetty dan Bhanu Ganesh Gudivada, Amazon Web Services

Ringkasan

Untuk alasan keamanan, selalu baik untuk menempatkan database di subnet pribadi. Kueri terhadap database dapat dijalankan dengan menghubungkan melalui host bastion Amazon Elastic Compute Cloud EC2 (Amazon) di subnet publik di Amazon Web Services (AWS) Cloud. Ini memerlukan penginstalan perangkat lunak, seperti pgAdmin DBeaver atau, yang biasa digunakan oleh pengembang atau administrator basis data, di host Amazon. EC2

Menjalankan pgAdmin di server Linux dan mengaksesnya melalui browser web memerlukan instalasi dependensi tambahan, pengaturan izin, dan konfigurasi.

Sebagai solusi alternatif, pengembang atau administrator database dapat terhubung ke database PostgreSQL dengan menggunakan pgAdmin untuk mengaktifkan terowongan SSH dari sistem lokal mereka. Dalam pendekatan ini, pgadmin menggunakan host EC2 Amazon di subnet publik sebagai host perantara sebelum menghubungkan ke database. Diagram di bagian Arsitektur menunjukkan pengaturan.

> **Note**
>
> Pastikan bahwa grup keamanan yang dilampirkan ke database PostgreSQL memungkinkan koneksi pada port 5432 dari host Amazon. EC2

Prasyarat dan batasan

Prasyarat

- Akun AWS yang ada
- Virtual Private Cloud (VPC) dengan subnet publik dan subnet pribadi
- EC2 Contoh dengan grup keamanan terlampir
- Database Edisi Amazon Aurora PostgreSQL yang kompatibel dengan grup keamanan terlampir
- Sebuah key pair Secure Shell (SSH) untuk menyiapkan terowongan

Versi produk

- pgAdmin versi 6.2+
- Amazon Aurora PostgreSQL Edisi yang kompatibel dengan versi 12.7+

Arsitektur

Tumpukan teknologi target

- Amazon EC2
- Kompatibel dengan Amazon Aurora PostgreSQL

Arsitektur target

Diagram berikut menunjukkan penggunaan pgAdmin dengan terowongan SSH untuk terhubung melalui gateway internet ke instance, yang terhubung ke EC2 database.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

Layanan lainnya

- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.

Epik

Buat koneksi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server.	Di pgAdmin, pilih Buat, lalu pilih Server. Untuk bantuan tambahan dalam mengatur pgAdmin untuk mendaftar kan server, mengonfigurasi koneksi, dan terhubung melalui tunneling SSH dengan menggunakan Dialog Server, lihat tautan di bagian Sumber daya terkait.	DBA
Berikan nama untuk server.	Pada tab Umum, masukkan nama.	DBA
Masukkan detail database.	Pada tab Koneksi, masukkan nilai untuk yang berikut ini: • Nama/alamat host • Port • Basis data pemeliharaan • Nama Pengguna • Kata Sandi	DBA
Masukkan detail EC2 server Amazon.	Pada tab SSH Tunnel, berikan detail EC2 instance Amazon yang ada di subnet publik. • Setel Gunakan SSH tunneling ke Ya untuk menentukan bahwa pgAdmin harus menggunak an terowongan SSH saat	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menghubungkan ke server yang ditentukan. • Di bidang Host Tunnel, tentukan nama atau alamat IP host SSH (misalnya, 10.xx). • Di bidang port Tunnel, tentukan port host SSH (misalnya, 22). • Di bidang Nama Pengguna, tentukan nama pengguna dengan hak masuk untuk host SSH (misalnya, ec2-user). • Tentukan jenis otentikasi sebagai file Identitas sehingga pgAdmin akan menggunakan file kunci pribadi saat menghubungkan. • Sertakan lokasi file Privacy Enhanced Mail (PEM) di bidang file Identity. File.pem adalah EC2 key pair Amazon.	
Simpan dan sambungkan.	Pilih Simpan untuk menyelesaikan pengaturan dan terhubung ke database Aurora PostgreSQL yang kompatibel dengan menggunakan terowongan SSH.	DBA

Sumber daya terkait

- Dialog Server
- Connect ke Server

Mengkonversi kueri JSON Oracle ke database PostgreSQL SQL

Pinesh Singal dan Lokesh Gurram, Amazon Web Services

Ringkasan

Proses migrasi untuk berpindah dari lokal ke Amazon Web Services (AWS) Cloud menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi kode dari database Oracle menjadi database PostgreSQL. Sebagian besar kode secara otomatis dikonversi oleh AWS SCT. Namun, kueri Oracle terkait JSON tidak dikonversi secara otomatis.

Mulai dari versi Oracle 12.2, Oracle Database mendukung berbagai fungsi JSON yang membantu dalam mengubah data berbasis JSON menjadi data berbasis Row. Namun, AWS SCT tidak secara otomatis mengonversi data berbasis JSON ke dalam bahasa yang didukung oleh PostgreSQL.

Pola migrasi ini terutama berfokus pada mengonversi kueri Oracle terkait JSON secara manual dengan fungsi seperti`JSON_OBJECT`,`JSON_ARRAYAGG`, dan `JSON_TABLE` dari database Oracle ke database PostgreSQL.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Instans database Oracle lokal (aktif dan berjalan)
- Amazon Relational Database Service (Amazon RDS) untuk instans database PostgreSQL atau Amazon Aurora PostgreSQL Edisi yang kompatibel dengan PostgreSQL (aktif dan berjalan)

Batasan

- Kueri terkait JSON memerlukan format dan tetap. `KEY VALUE` Tidak menggunakan format itu mengembalikan hasil yang salah.
- Jika ada perubahan dalam struktur JSON menambahkan baru `KEY` dan `VALUE` pasangan di bagian hasil, prosedur atau fungsi yang sesuai harus diubah dalam kueri SQL.
- Beberapa fungsi terkait JSON didukung di versi Oracle dan PostgreSQL sebelumnya tetapi dengan kemampuan yang lebih sedikit.

Versi produk

- Oracle Database versi 12.2 dan yang lebih baru
- Amazon RDS untuk PostgreSQL atau Aurora PostgreSQL yang kompatibel dengan versi 9.5 dan yang lebih baru
- AWS SCT versi terbaru (diuji menggunakan versi 1.0.664)

Arsitektur

Tumpukan teknologi sumber

- Sebuah instance database Oracle dengan versi 19c

Tumpukan teknologi target

- Instans database yang kompatibel dengan Amazon RDS for PostgreSQL atau Aurora PostgreSQL dengan versi 13

Arsitektur target

1. Gunakan AWS SCT dengan kode fungsi JSON untuk mengonversi kode sumber dari Oracle ke PostgreSQL.
2. Konversi menghasilkan file.sql bermigrasi yang didukung PostgreSQL.
3. Secara manual mengonversi kode fungsi Oracle JSON yang tidak dikonversi ke kode fungsi PostgreSQL JSON.
4. Jalankan file.sql pada instans DB yang kompatibel dengan Aurora PostgreSQL target.

Alat

Layanan AWS

- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.
- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di AWS Cloud.

- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Layanan lainnya

- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi yang menyederhanakan pengembangan dan pengelolaan database Oracle baik dalam penerapan tradisional maupun berbasis cloud.
- pgAdmin atau. DBeaver pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database. DBeaveradalah alat database universal.

Praktik terbaik

Kueri Oracle memiliki tipe `CAST` sebagai default saat menggunakan `JSON_TABLE` fungsi. Praktik terbaik adalah menggunakan `CAST` PostgreSQL juga, menggunakan double greater-than characters (). >>

Untuk informasi selengkapnya, lihat PostgreS_SQL_READ_JSON di bagian Informasi tambahan.

Epik

Menghasilkan data JSON dalam database Oracle dan PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Simpan data JSON dalam database Oracle.	Buat tabel di database Oracle, dan simpan data JSON di kolom. `CLOB` Gunakan Oracle_Table_Creation_Inser t_Script yang ada di bagian Informasi tambahan.	Insinyur migrasi
Simpan data JSON dalam database PostgreSQL.	Buat tabel di database PostgreSQL, dan simpan data JSON di kolom. `TEXT` Gunakan Postgres_Table_Cre	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ation_Insert_Script yang ada di bagian Informasi tambahan.	

Ubah JSON menjadi format ROW

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konversi data JSON pada database Oracle.	Tulis query Oracle SQL untuk membaca data JSON ke dalam format ROW. Untuk detail selengkapnya dan contoh sintaks, lihat Oracle_SQL_READ_JSON di bagian Informasi tambahan.	Insinyur migrasi
Konversi data JSON pada database PostgreSQL.	Tulis kueri PostgreSQL untuk membaca data JSON ke dalam format ROW. Untuk detail selengkapnya dan contoh sintaks, lihat PostgreS_SQL_READ_JSON di bagian Informasi tambahan.	Insinyur migrasi

Secara manual mengkonversi data JSON menggunakan query SQL dan melaporkan output dalam format JSON

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan agregasi dan validasi pada query Oracle SQL.	Untuk mengonversi data JSON secara manual, lakukan gabungan, agregasi, dan validasi pada kueri Oracle SQL, dan laporkan output dalam format JSON. Gunakan	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kode di bawah Oracle_SQL_JSON_AGGREGATION_JOIN di bagian Informasi tambahan. 1. JOIN - Data berformat JSON diteruskan sebagai parameter input ke kueri. Gabungan batin dibuat antara data statis ini dan data JSON di tabel Oracle DB. `aws_test_table` 2. Agregasi dengan validasi — Data JSON memiliki `KEY` dan `VALUE` parameter dengan nilai-nilai sepertiaccountNumber ,, businessUnitId dan parentAccountNumber positionId , yang digunakan untuk `SUM` dan agregasi. `COUNT` 3. Format JSON — Setelah bergabung dan agregasi, data dilaporkan dalam format JSON dengan menggunakan dan. `JSON_OBJECT` `JSON_ARRAYAGG`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan agregasi dan validasi pada query Postgres SQL.	Untuk mengonversi data JSON secara manual, lakukan gabungan, agregasi, dan validasi pada kueri PostgreSQL, dan laporkan output dalam format JSON. Gunakan kode di bawah Postgres_SQL_JSON_AGGREGATION_JOIN di bagian Informasi tambahan. 1. JOIN - Data berformat JSON (`tab1`) dilewatkan sebagai parameter input ke kueri klausa. `WITH` JOIN dibuat antara data statis ini dan data JSON, yang ada di `tab` tabel. JOIN juga dibuat dengan `WITH` klausa, yang memiliki data JSON dalam tabel. `aws_test_pg_table` 2. Agregasi — Data JSON memiliki `KEY` dan `VALUE` parameter dengan nilai-nilai seperti`accountNumber` ,,`parentAccountNumber` , dan `businessUnitId positionId` , yang digunakan untuk `SUM` dan `COUNT` agregasi. 3. Format JSON — Setelah bergabung dan agregasi,	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	data dilaporkan dalam format JSON dengan menggunakan dan. `JSON_BUILD_OBJECT` `JSON_AGG`	

Ubah prosedur Oracle menjadi fungsi PostgreSQL yang berisi kueri JSON

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah kueri JSON dalam prosedur Oracle menjadi baris.	Untuk contoh prosedur Oracle, gunakan kueri Oracle sebelumnya dan kode di bawah Oracle_procedure _with_json_query di bagian Informasi tambahan.	Insinyur migrasi
Ubah fungsi PostgreSQL yang memiliki kueri JSON menjadi data berbasis baris.	Untuk contoh fungsi PostgreSQL, gunakan kueri PostgreSQL sebelumny a dan kode yang ada di bawah PostgreS_Function_ With_JSON_QUERY di bagian Informasi tambahan.	Insinyur migrasi

Sumber daya terkait

- Fungsi Oracle JSON
- Fungsi PostgreSQL JSON
- Contoh Fungsi Oracle JSON
- Contoh fungsi PostgreSQL JSON
- Alat Konversi Skema AWS

Informasi tambahan

Untuk mengonversi kode JSON dari database Oracle ke database PostgreSQL, gunakan skrip berikut, secara berurutan.

1. Oracle_Table_Creation_Insert_Script

```
create table aws_test_table(id number,created_on date default sysdate,modified_on
 date,json_doc clob);

REM INSERTING into EXPORT_TABLE
SET DEFINE OFF;
Insert into aws_test_table (ID,CREATED_ON,MODIFIED_ON,json_doc)
values (1,to_date('02-AUG-2022 12:30:14','DD-MON-YYYY HH24:MI:SS'),to_date('02-AUG-2022
 12:30:14','DD-MON-YYYY HH24:MI:SS'),TO_CLOB(q'[{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0100",
    "arrayPattern" : " -'",
    "a]')
|| TO_CLOB(q'[ccount" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
      "accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "retOdertcaShr ",
      "city" : "new york",
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
```

```
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
]')
|| TO_CLOB(q'[        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}]'));
Insert into aws_test_table (ID,CREATED_ON,MODIFIED_ON,json_doc) values (2,to_date('02-
AUG-2022 12:30:14','DD-MON-YYYY HH24:MI:SS'),to_date('02-AUG-2022 12:30:14','DD-MON-
YYYY HH24:MI:SS'),TO_CLOB(q'[{
  "metadata" : {
    "upperLastNameFirstName" : "PQR XYZ",
    "upperEmailAddress" : "pqr@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "54534343",
    "displayName" : "Xyz, pqr",
    "firstName" : "pqr",
    "lastName" : "Xyz",
    "emailAddress" : "pqr@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0090",
    "arrayPattern" : " -'",
    "account" : {
      "companyId" : "CARS",
      "busin]')
|| TO_CLOB(q'[essUnitId" : 6,
      "accountNumber" : 42001,
      "parentAccountNumber" : 32001,
      "firstName" : "terry",
      "lastName" : "whitlock",
      "street1" : "UO  123",
      "city" : "TOTORON",
      "region" : "NO",
      "postalcode" : "LKM 111",
      "country" : "Canada"
    },
    "products" : [
```

```
      {
        "appUserGuid" : "ia744d7790000016899f8cf3f417d6df6",
        "id" : "0000000014",
        "name" : "ProView eLooseleaf",
      ]')
|| TO_CLOB(q'[  "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}]'));

commit;
```

2. Postgres_Table_Creation_Insert_Script

```
create table aws_test_pg_table(id int,created_on date ,modified_on date,json_doc text);
insert into aws_test_pg_table(id,created_on,modified_on,json_doc)
values(1,now(),now(),'{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0100",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
      "accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "retOdertcaShr ",
      "city" : "new york",
```

```
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}');


insert into aws_test_pg_table(id,created_on,modified_on,json_doc)
values(2,now(),now(),'{
  "metadata" : {
    "upperLastNameFirstName" : "PQR XYZ",
    "upperEmailAddress" : "pqr@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "54534343",
    "displayName" : "Xyz, pqr",
    "firstName" : "pqr",
    "lastName" : "Xyz",
    "emailAddress" : "a*b**@h**.k**",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0090",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "CARS",
      "businessUnitId" : 6,
      "accountNumber" : 42001,
      "parentAccountNumber" : 32001,
      "firstName" : "terry",
      "lastName" : "whitlock",
      "street1" : "UO  123",
      "city" : "TOTORON",
      "region" : "NO",
      "postalcode" : "LKM 111",
```

```
      "country" : "Canada"
    },
    "products" : [
      {
        "appUserGuid" : "ia744d7790000016899f8cf3f417d6df6",
        "id" : "0000000014",
        "name" : "ProView eLooseleaf",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}');
```

3. Oracle_SQL_READ_JSON

Blok kode berikut menunjukkan bagaimana mengkonversi data Oracle JSON ke dalam format baris.

Contoh query dan sintaks

```
SELECT   JSON_OBJECT(
 'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                               ) ) ) ) FROM
    (SELECT   tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE 0 END) tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
 attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
 non_attorney_count,
```

```
           SUM(CASE    WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
clerk_count
       FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
COLUMNS (
         parent_account_number NUMBER PATH
          '$.data.account.parentAccountNumber',
           account_number NUMBER PATH '$.data.account.accountNumber',
           business_unit_id NUMBER PATH '$.data.account.businessUnitId',
           position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
           ) AS tab_data
           INNER JOIN JSON_TABLE ( '{
       "accounts": [{
         "accountNumber": 42000,
         "parentAccountNumber": 32000,
         "businessUnitId": 7
       }, {
         "accountNumber": 42001,
         "parentAccountNumber": 32001,
         "businessUnitId": 6
       }]
     }', '$.accounts[*]' ERROR ON ERROR
     COLUMNS (
     parent_account_number PATH '$.parentAccountNumber',
     account_number PATH '$.accountNumber',
     business_unit_id PATH '$.businessUnitId')
     ) static_data
     ON ( static_data.parent_account_number = tab_data.parent_account_number
          AND static_data.account_number = tab_data.account_number
          AND static_data.business_unit_id = tab_data.business_unit_id )
       GROUP BY
           tab_data.business_unit_id,
           tab_data.parent_account_number,
           tab_data.account_number );
```

Dokumen JSON menyimpan data sebagai koleksi. Setiap koleksi dapat memiliki KEY dan VALUE berpasangan. Setiap VALUE bisa bersarang KEY dan VALUE berpasangan. Tabel berikut memberikan informasi tentang membaca spesifik VALUE dari dokumen JSON.

KUNCI	HIERARKI atau PATH yang akan digunakan untuk mendapatkan NILAI	NILAI

profileType	metadata -> profileType	“P”
positionId	data -> positionId	“0100"
accountNumber	data-> akun -> accountNumber	42000

Pada tabel sebelumnya, KEY profileType adalah a VALUE dari metadataKEY. KEYpositionIdItu adalah VALUE dari dataKEY. KEYaccountNumberItu adalah VALUE dari accountKEY, dan itu account KEY adalah VALUE dari dataKEY.

Contoh dokumen JSON

```
{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
"profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
"positionId" : "0100",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
"accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "retOdertcaShr ",
      "city" : "new york",
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
```

```
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}
```

Kueri SQL yang digunakan untuk mendapatkan bidang yang dipilih dari dokumen JSON

```
select parent_account_number,account_number,business_unit_id,position_id from
 aws_test_table aws,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
COLUMNS (
parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
account_number NUMBER PATH '$.data.account.accountNumber',
business_unit_id NUMBER PATH '$.data.account.businessUnitId',
position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'
)) as sc
```

Dalam query sebelumnya, `JSON_TABLE` adalah fungsi built-in di Oracle yang mengubah data JSON ke dalam format baris. Fungsi JSON_TABLE mengharapkan parameter dalam format JSON.

Setiap item di `COLUMNS` memiliki standar`PATH`, dan ada yang sesuai `VALUE` untuk yang `KEY` diberikan dikembalikan dalam format baris.

Hasil dari query sebelumnya

PARENT_AC COUNT_NUMBER	ACCOUNT_NUMBER	BUSINESS_UNIT_ID	POSISI_ID
32000	42000	7	0100
32001	42001	6	0090

4. PostgreS_SQL_READ_JSON

Contoh query dan sintaks

```
select *
from (
select (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER as
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->>'positionId')::VARCHAR as positionId
from aws_test_pg_table) d ;
```

Di Oracle, PATH digunakan untuk mengidentifikasi spesifik KEY danVALUE. Namun, PostgreSQL menggunakan model untuk membaca dan dari HIERARCHY JSON. KEY VALUE Data JSON yang sama yang disebutkan di bawah Oracle_SQL_Read_JSON digunakan dalam contoh berikut.

Kueri SQL dengan tipe CAST tidak diizinkan

(Jika Anda memaksa mengetikCAST, kueri gagal dengan kesalahan sintaks.)

```
select *
from (
select (json_doc::json->'data'->'account'->'parentAccountNumber') as
 parentAccountNumber,
(json_doc::json->'data'->'account'->'accountNumber')as accountNumber,
(json_doc::json->'data'->'account'->'businessUnitId') as businessUnitId,
(json_doc::json->'data'->'positionId')as positionId
from aws_test_pg_table) d ;
```

Menggunakan satu operator yang lebih besar dari (>) akan mengembalikan yang VALUE ditentukan untuk itu. KEY Misalnya,KEY:positionId, danVALUE:"0100".

Jenis tidak CAST diperbolehkan saat Anda menggunakan operator tunggal yang lebih besar dari ()>.

Kueri SQL dengan tipe CAST diperbolehkan

```
select *
from (
select (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER as
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->>'positionId')::varchar as positionId
from aws_test_pg_table) d ;
```

Untuk menggunakan tipeCAST, Anda harus menggunakan operator ganda yang lebih besar dari pada. Jika Anda menggunakan operator tunggal yang lebih besar dari pada, kueri mengembalikan yang VALUE ditentukan (misalnya,KEY:positionId, danVALUE:"0100"). Menggunakan double greater-than operator (>>) akan mengembalikan nilai aktual yang ditentukan untuk itu KEY (misalnya,KEY:, danVALUE: positionId0100, tanpa tanda kutip ganda).

Dalam kasus sebelumnya, parentAccountNumber adalah type CAST toINT, accountNumber is type CAST toINT, businessUnitId is type CAST toINT, dan positionId type CAST to. VARCHAR

Tabel berikut menunjukkan hasil kueri yang menjelaskan peran operator tunggal yang lebih besar dari (>) dan operator ganda yang lebih besar dari (). >>

Pada tabel tabel pertama, kueri menggunakan single greater-than operator ()>. Setiap kolom dalam tipe JSON dan tidak dapat diubah menjadi tipe data lain.

parentAccountNumber	accountNumber	businessUnitId	PositionId
2003565430	2003564830	7	“0100"
2005284042	2005284042	6	“0090"
2000272719	2000272719	1	“0100"

Pada tabel kedua, query menggunakan double greater-than operator ()>>. Setiap kolom mendukung jenis CAST berdasarkan nilai kolom. Misalnya, INTEGER dalam konteks ini.

parentAccountNumber	accountNumber	businessUnitId	PositionId
2003565430	2003564830	7	0100
2005284042	2005284042	6	0090
2000272719	2000272719	1	0100

5. ORACLE_SQL_JSON_AGGREGATION_BERGABUNG

Contoh kueri

```
SELECT
    JSON_OBJECT(
        'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                               ) ) ) )
FROM
    (SELECT
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE 0 END) tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
 attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
 non_attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
 clerk_count
       FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
 COLUMNS (
          parent_account_number NUMBER PATH
           '$.data.account.parentAccountNumber',
            account_number NUMBER PATH '$.data.account.accountNumber',
            business_unit_id NUMBER PATH '$.data.account.businessUnitId',
            position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
            ) AS tab_data
            INNER JOIN JSON_TABLE ( '{
       "accounts": [{
         "accountNumber": 42000,
         "parentAccountNumber": 32000,
         "businessUnitId": 7
       }, {
         "accountNumber": 42001,
```

```
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }]
      }', '$.accounts[*]' ERROR ON ERROR
      COLUMNS (
      parent_account_number PATH '$.parentAccountNumber',
      account_number PATH '$.accountNumber',
      business_unit_id PATH '$.businessUnitId')
      ) static_data
      ON ( static_data.parent_account_number = tab_data.parent_account_number
           AND static_data.account_number = tab_data.account_number
           AND static_data.business_unit_id = tab_data.business_unit_id )
        GROUP BY
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number
    );
```

Untuk mengonversi data tingkat baris ke dalam format JSON, Oracle memiliki fungsi bawaan seperti`JSON_OBJECT`,,, `JSON_ARRAY` dan. `JSON_OBJECTAGG` `JSON_ARRAYAGG`

- `JSON_OBJECT`menerima dua parameter: `KEY` dan`VALUE`. `KEY`Parameter harus di-hardcode atau statis. `VALUE`Parameter ini berasal dari output tabel.
- `JSON_ARRAYAGG`menerima `JSON_OBJECT` sebagai parameter. Ini membantu dalam mengelompokkan kumpulan `JSON_OBJECT` elemen sebagai daftar. Misalnya, jika Anda memiliki `JSON_OBJECT` elemen yang memiliki beberapa catatan (beberapa `KEY` dan `VALUE` pasangan dalam kumpulan data), `JSON_ARRAYAGG` tambahkan kumpulan data dan buat daftar. Menurut bahasa Struktur Data, `LIST` adalah sekelompok elemen. Dalam konteks ini, `LIST` adalah sekelompok `JSON_OBJECT` elemen.

Contoh berikut menunjukkan satu `JSON_OBJECT` elemen.

```
{
   "taxProfessionalCount": 0,
   "attorneyCount": 0,
   "nonAttorneyCount": 1,
   "clerkCount": 0
}
```

Contoh berikutnya menunjukkan dua `JSON_OBJECT` elemen, dengan `LIST` ditunjukkan oleh kawat gigi persegi (`[ ]`).

```
[
    {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
,
    {
        "taxProfessionalCount": 2,
        "attorneyCount": 1,
        "nonAttorneyCount": 3,
        "clerkCount":4
      }
]
```

Contoh query SQL

```
SELECT
    JSON_OBJECT(
        'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                               )
                        )
                                          )
              )
FROM
    (SELECT
            tab_data.business_unit_id,
            tab_data.parent_account_number,
```

```
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE   0 END
            )      tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN        1    ELSE
0 END
            )      attorney_count,

            SUM(CASE    WHEN tab_data.position_id = '0090' THEN        1    ELSE
0 END
            )      non_attorney_count,

            SUM(CASE    WHEN tab_data.position_id = '0050' THEN        1    ELSE
0 END
            )      clerk_count

        FROM
            aws_test_table scco,  JSON_TABLE ( json_doc, '$' ERROR ON ERROR
            COLUMNS (
            parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
            account_number NUMBER PATH '$.data.account.accountNumber',
            business_unit_id NUMBER PATH '$.data.account.businessUnitId',
            position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
            ) AS tab_data
            INNER JOIN JSON_TABLE ( '{
        "accounts": [{
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }, {
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }]
      }', '$.accounts[*]' ERROR ON ERROR
      COLUMNS (
      parent_account_number PATH '$.parentAccountNumber',
      account_number PATH '$.accountNumber',
      business_unit_id PATH '$.businessUnitId')
      ) static_data ON ( static_data.parent_account_number =
tab_data.parent_account_number
                          AND static_data.account_number = tab_data.account_number
```

```
                          AND static_data.business_unit_id =
 tab_data.business_unit_id )
        GROUP BY
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number
    );
```

Contoh output dari query SQL sebelumnya

```
{
  "accountCounts": [
    {
      "businessUnitId": 6,
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

6. PostgreS_SQL_JSON_AGGREGATION_BERGABUNG

PostgreSQL built-in JSON_BUILD_OBJECT berfungsi JSON_AGG dan mengkonversi data tingkat baris ke dalam format JSON. JSON_BUILD_OBJECTPostgreSQL JSON_AGG dan setara dengan Oracle dan. JSON_OBJECT JSON_ARRAYAGG

Contoh kueri

```
select
JSON_BUILD_OBJECT ('accountCounts',
    JSON_AGG(
        JSON_BUILD_OBJECT ('businessUnitId',businessUnitId
        ,'parentAccountNumber',parentAccountNumber
        ,'accountNumber',accountNumber
        ,'totalOnlineContactsCount',online_contacts_count,
        'countByPosition',
            JSON_BUILD_OBJECT (
            'taxProfessionalCount',tax_professional_count
            ,'attorneyCount',attorney_count
            ,'nonAttorneyCount',non_attorney_count
            ,'clerkCount',clerk_count
            )
        )
    )
)
from (
with tab as (select * from (
select (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER as
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->>'positionId')::varchar as positionId
from aws_test_pg_table) a ) ,
tab1 as ( select
(json_array_elements(b.jc -> 'accounts') ->> 'accountNumber')::integer accountNumber,
(json_array_elements(b.jc -> 'accounts') ->> 'businessUnitId')::integer
 businessUnitId,
(json_array_elements(b.jc -> 'accounts') ->> 'parentAccountNumber')::integer
 parentAccountNumber
from (
select '{
        "accounts": [{
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
```

```
        }, {
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }]
      }'::json as jc) b)
select
tab.businessUnitId::text,
tab.parentAccountNumber::text,
tab.accountNumber::text,
SUM(1) online_contacts_count,
SUM(CASE WHEN tab.positionId::text = '0095' THEN 1 ELSE 0  END)
 tax_professional_count,
SUM(CASE WHEN tab.positionId::text = '0100' THEN 1 ELSE 0  END)      attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0090' THEN      1  ELSE      0 END)
 non_attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0050' THEN      1  ELSE      0 END)
 clerk_count
from tab1,tab
where tab.parentAccountNumber::INTEGER=tab1.parentAccountNumber::INTEGER
and tab.accountNumber::INTEGER=tab1.accountNumber::INTEGER
and tab.businessUnitId::INTEGER=tab1.businessUnitId::INTEGER
GROUP BY      tab.businessUnitId::text,
            tab.parentAccountNumber::text,
            tab.accountNumber::text) a;
```

Contoh output dari query sebelumnya

Output dari Oracle dan PostgreSQL persis sama.

```
{
  "accountCounts": [
    {
      "businessUnitId": 6,
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
```

```
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

7.ORACLE_PROCEDURE_with_JSON_QUERY

Kode ini mengubah prosedur Oracle menjadi fungsi PostgreSQL yang memiliki query JSON SQL. Ini menunjukkan bagaimana query mentransposisi JSON menjadi baris dan sebaliknya.

```
CREATE OR REPLACE PROCEDURE p_json_test(p_in_accounts_json IN varchar2,
 p_out_accunts_json  OUT varchar2)
IS
BEGIN
/*
p_in_accounts_json paramter should have following format:
       {
        "accounts": [{
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }, {
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }]
      }
*/
SELECT
    JSON_OBJECT(
        'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
```

```
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                               ) ) ) )
into p_out_accunts_json
FROM
    (SELECT
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE 0 END) tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
 attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
 non_attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
 clerk_count
        FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
            COLUMNS (
            parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
            account_number NUMBER PATH '$.data.account.accountNumber',
            business_unit_id NUMBER PATH '$.data.account.businessUnitId',
            position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
            ) AS tab_data
            INNER JOIN JSON_TABLE ( p_in_accounts_json, '$.accounts[*]' ERROR ON ERROR

      COLUMNS (
      parent_account_number PATH '$.parentAccountNumber',
      account_number PATH '$.accountNumber',
      business_unit_id PATH '$.businessUnitId')
      ) static_data
      ON ( static_data.parent_account_number = tab_data.parent_account_number
           AND static_data.account_number = tab_data.account_number
           AND static_data.business_unit_id = tab_data.business_unit_id )
        GROUP BY
            tab_data.business_unit_id,
            tab_data.parent_account_number,
```

```
            tab_data.account_number
    );
EXCEPTION
WHEN OTHERS THEN
   raise_application_error(-20001,'Error while running the JSON query');
END;
/
```

Menjalankan prosedur

Blok kode berikut menjelaskan bagaimana Anda dapat menjalankan prosedur Oracle yang dibuat sebelumnya dengan contoh input JSON ke prosedur. Ini juga memberi Anda hasil atau output dari prosedur ini.

```
set serveroutput on;
declare
v_out varchar2(30000);
v_in varchar2(30000):= '{
        "accounts": [{
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }, {
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }]
      }';
begin
  p_json_test(v_in,v_out);
  dbms_output.put_line(v_out);
end;
/
```

Keluaran prosedur

```
{
  "accountCounts": [
    {
      "businessUnitId": 6,
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
```

```
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

8.PostgreS_Function_with_JSON_QUERY

Contoh fungsi

```
CREATE OR REPLACE  FUNCTION f_pg_json_test(p_in_accounts_json  text)
RETURNS text
LANGUAGE plpgsql
AS
$$
DECLARE
 v_out_accunts_json   text;
BEGIN
SELECT
JSON_BUILD_OBJECT ('accountCounts',
    JSON_AGG(
        JSON_BUILD_OBJECT ('businessUnitId',businessUnitId
        ,'parentAccountNumber',parentAccountNumber
        ,'accountNumber',accountNumber
        ,'totalOnlineContactsCount',online_contacts_count,
        'countByPosition',
            JSON_BUILD_OBJECT (
            'taxProfessionalCount',tax_professional_count
```

```
            ,'attorneyCount',attorney_count
            ,'nonAttorneyCount',non_attorney_count
            ,'clerkCount',clerk_count
            ))))
INTO v_out_accunts_json
FROM (
WITH tab AS (SELECT * FROM (
SELECT (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER AS
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER AS accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER AS businessUnitId,
(json_doc::json->'data'->>'positionId')::varchar AS positionId
FROM aws_test_pg_table) a ) ,
tab1 AS ( SELECT
(json_array_elements(b.jc -> 'accounts') ->> 'accountNumber')::integer accountNumber,
(json_array_elements(b.jc -> 'accounts') ->> 'businessUnitId')::integer businessUnitId,
(json_array_elements(b.jc -> 'accounts') ->> 'parentAccountNumber')::integer
 parentAccountNumber
FROM (
SELECT p_in_accounts_json::json AS jc) b)
SELECT
tab.businessUnitId::text,
tab.parentAccountNumber::text,
tab.accountNumber::text,
SUM(1) online_contacts_count,
SUM(CASE WHEN tab.positionId::text = '0095' THEN 1 ELSE 0  END)
 tax_professional_count,
SUM(CASE WHEN tab.positionId::text = '0100' THEN 1 ELSE 0  END)      attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0090' THEN      1  ELSE      0 END)
 non_attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0050' THEN      1  ELSE      0 END)
 clerk_count
FROM tab1,tab
WHERE tab.parentAccountNumber::INTEGER=tab1.parentAccountNumber::INTEGER
AND tab.accountNumber::INTEGER=tab1.accountNumber::INTEGER
AND tab.businessUnitId::INTEGER=tab1.businessUnitId::INTEGER
GROUP BY      tab.businessUnitId::text,
            tab.parentAccountNumber::text,
            tab.accountNumber::text) a;
RETURN v_out_accunts_json;
END;
$$;
```

Menjalankan fungsi

```
select    f_pg_json_test('{
        "accounts": [{
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }, {
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }]
      }')   ;
```

Output fungsi

Output berikut ini mirip dengan output prosedur Oracle. Perbedaannya adalah bahwa output ini dalam format Teks.

```
{
  "accountCounts": [
    {
      "businessUnitId": "6",
      "parentAccountNumber": "32001",
      "accountNumber": "42001",
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": "7",
      "parentAccountNumber": "32000",
      "accountNumber": "42000",
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
```

```
      }
    }
  ]
}
```

Salin tabel Amazon DynamoDB di seluruh akun menggunakan AWS Backup

Ramkumar Ramanujam, Amazon Web Services

Ringkasan

Saat bekerja dengan Amazon DynamoDB aktif AWS, kasus penggunaan umum adalah menyalin atau menyinkronkan tabel DynamoDB dalam lingkungan pengembangan, pengujian, atau pementasan dengan data tabel yang ada di lingkungan produksi. Sebagai praktik standar, setiap lingkungan menggunakan yang berbeda Akun AWS.

AWS Backup mendukung pencadangan dan pemulihan data lintas wilayah dan lintas akun untuk DynamoDB, Amazon Simple Storage Service (Amazon S3), dan lainnya. Layanan AWS Pola ini menyediakan langkah-langkah untuk menggunakan AWS Backup cross-account backup dan restore untuk menyalin tabel DynamoDB antara. Akun AWS

Prasyarat dan batasan

Prasyarat

- Dua aktif Akun AWS yang berasal dari organisasi yang sama di AWS Organizations
- Izin untuk membuat tabel DynamoDB di kedua akun
- AWS Identity and Access Management (IAM) izin untuk membuat dan menggunakan brankas AWS Backup

Batasan

- Sumber dan target Akun AWS harus menjadi bagian dari organisasi yang sama di AWS Organizations.

Arsitektur

Tumpukan teknologi target

- AWS Backup
- Amazon DynamoDB

Arsitektur target

1. Buat cadangan tabel DynamoDB di brankas cadangan AWS Backup di akun sumber.
2. Salin cadangan ke brankas cadangan di akun target.
3. Kembalikan tabel DynamoDB di akun target dengan menggunakan cadangan dari brankas cadangan di akun target.

Otomatisasi dan skala

Anda dapat menggunakan AWS Backup untuk menjadwalkan backup untuk berjalan pada interval tertentu.

Alat

- AWS Backupadalah layanan yang dikelola sepenuhnya untuk memusatkan dan mengotomatiskan perlindungan data di seluruh Layanan AWS, di cloud, dan di tempat. Dengan menggunakan layanan ini, Anda dapat mengonfigurasi kebijakan pencadangan dan memantau aktivitas AWS sumber daya Anda di satu tempat. Ini memungkinkan Anda untuk mengotomatiskan dan mengkonsolidasikan tugas pencadangan yang sebelumnya dilakukan layanan demi layanan, dan menghilangkan kebutuhan untuk membuat skrip khusus dan proses manual.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi dengan skalabilitas yang mulus.

Epik

Aktifkan AWS Backup fitur di akun sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan fitur lanjutan untuk DynamoDB dan cadangan lintas akun.	Baik di sumber maupun target Akun AWS, lakukan hal berikut: 1. Di Konsol Manajemen AWS, buka AWS Backup konsol.	AWS DevOps, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pilih Pengaturan. 3. Di bawah Fitur lanjutan untuk cadangan Amazon DynamoDB, konfirmas ikan bahwa fitur lanjutan diaktifkan, atau pilih Aktifkan. 4. Di bawah Manajemen lintas akun, untuk pencadangan lintas akun, pilih Aktifkan.	

Buat brankas cadangan di akun sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat brankas cadangan.	Baik di sumber maupun target Akun AWS, lakukan hal berikut: 1. Di AWS Backup konsol, pilih Backup vaults. 2. Pilih Buat brankas Cadangan. 3. Salin Nama Sumber Daya Amazon (ARN) dari brankas cadangan dan simpan. Vault cadangan sumber dan target akan diperlukan saat Anda menyalin cadangan tabel DynamoDB antara akun sumber dan target. ARNs	AWS DevOps, Insinyur migrasi

Lakukan pencadangan dan pemulihan menggunakan brankas cadangan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Di akun sumber, buat cadangan tabel DynamoDB.	Untuk membuat cadangan untuk tabel DynamoDB di akun sumber, lakukan hal berikut: 1. Pada halaman AWS Backup Dasbor, pilih Buat cadangan sesuai permintaan. 2. Di bagian Pengaturan, untuk jenis sumber daya, pilih DynamoDB, lalu pilih nama tabel. 3. Di daftar dropdown Backup vault, pilih brankas cadangan yang Anda buat di akun sumber. 4. Pilih periode Retensi yang Anda inginkan. 5. Pilih Buat cadangan sesuai permintaan. Pekerjaan cadangan baru dibuat. Untuk memantau status pekerjaan cadangan, pada halaman AWS Backup Pekerjaan, pilih tab Backup Jobs. Semua pekerjaan pencadangan yang aktif,	AWS DevOps, DBA, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sedang berlangsung, dan selesai tercantum di tab ini.	
Salin cadangan dari akun sumber ke akun target.	Setelah pekerjaan pencadangan selesai, salin cadangan tabel DynamoDB dari brankas cadangan di akun sumber ke brankas cadangan di akun target. Untuk menyalin brankas cadangan, di akun sumber, lakukan hal berikut: 1. Di AWS Backup konsol, pilih Backup vaults. 2. Di bawah Backups, pilih cadangan tabel DynamoDB. 3. Pilih Tindakan, Salin. 4. Masukkan Wilayah AWS akun target. 5. Untuk ARN vault eksternal, masukkan ARN dari brankas cadangan yang Anda buat di akun target. 6. Untuk menyalin cadangan dari akun sumber ke akun target, di brankas cadangan akun target, aktifkan akses dari akun lain.	AWS DevOps, Insinyur migrasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan cadangan di akun target.	Dalam target Akun AWS, lakukan hal berikut: 1. Di AWS Backup konsol, pilih Backup vaults. 2. Di bawah Cadangan, pilih cadangan yang Anda salin dari akun sumber. 3. Pilih Tindakan, Pulihkan. 4. Masukkan nama tabel DynamoDB target yang ingin Anda pulihkan.	AWS DevOps, DBA, Insinyur migrasi

Sumber daya terkait

- Menggunakan AWS Backup dengan DynamoDB
- Membuat salinan cadangan di seluruh Akun AWS
- AWS Backup harga

Salin tabel Amazon DynamoDB di seluruh akun menggunakan implementasi kustom

Ramkumar Ramanujam, Amazon Web Services

Ringkasan

Saat bekerja dengan Amazon DynamoDB di Amazon Web Services (AWS), kasus penggunaan umum adalah menyalin atau menyinkronkan tabel DynamoDB dalam lingkungan pengembangan, pengujian, atau pementasan dengan data tabel yang ada di lingkungan produksi. Sebagai praktik standar, setiap lingkungan menggunakan akun AWS yang berbeda.

DynamoDB sekarang mendukung pencadangan lintas akun menggunakan AWS Backup. Untuk informasi tentang biaya penyimpanan terkait saat menggunakan AWS Backup, lihat harga AWS Backup. Saat Anda menggunakan AWS Backup untuk menyalin seluruh akun, akun sumber dan target harus menjadi bagian dari organisasi AWS Organizations. Ada solusi lain untuk pencadangan dan pemulihan lintas akun menggunakan layanan AWS seperti AWS Glue. Namun, menggunakan solusi tersebut meningkatkan jejak aplikasi, karena ada lebih banyak layanan AWS untuk diterapkan dan dipelihara.

Anda juga dapat menggunakan Amazon DynamoDB Streams untuk menangkap perubahan tabel di akun sumber. Kemudian Anda dapat memulai fungsi AWS Lambda, dan membuat perubahan yang sesuai pada tabel target di akun target. Tetapi solusi itu berlaku untuk kasus penggunaan di mana tabel sumber dan target harus selalu disinkronkan. Ini mungkin tidak berlaku untuk pengembangan, pengujian, dan lingkungan pementasan di mana data sering diperbarui.

Pola ini menyediakan langkah-langkah untuk menerapkan solusi khusus untuk menyalin tabel Amazon DynamoDB dari satu akun ke akun lainnya. Pola ini dapat diimplementasikan menggunakan bahasa pemrograman umum seperti C #, Java, dan Python. Sebaiknya gunakan bahasa yang didukung oleh AWS SDK.

Prasyarat dan batasan

Prasyarat

- Dua akun AWS aktif
- Tabel DynamoDB di kedua akun
- Pengetahuan tentang peran dan kebijakan AWS Identity and Access Management (IAM)

- Pengetahuan tentang cara mengakses tabel Amazon DynamoDB menggunakan bahasa pemrograman umum, seperti C #, Java, atau Python

Batasan

Pola ini berlaku untuk tabel DynamoDB yang berukuran sekitar 2 GB atau lebih kecil. Dengan logika tambahan untuk menangani koneksi atau interupsi sesi, pelambatan, dan kegagalan dan percobaan ulang, ini dapat digunakan untuk tabel yang lebih besar.

Operasi pemindaian DynamoDB, yang membaca item dari tabel sumber, hanya dapat mengambil hingga 1 MB data dalam satu panggilan. Untuk tabel yang lebih besar, lebih besar dari 2 GB, batasan ini dapat meningkatkan total waktu untuk melakukan salinan tabel lengkap.

Arsitektur

Diagram berikut menunjukkan implementasi kustom antara sumber dan target akun AWS. Kebijakan IAM dan token keamanan digunakan dengan implementasi khusus. Data dibaca dari Amazon DynamoDB di akun sumber dan ditulis ke DynamoDB di akun target.

Otomatisasi dan skala

Pola ini berlaku untuk tabel DynamoDB yang ukurannya lebih kecil, sekitar 2 GB.

Untuk menerapkan pola ini untuk tabel yang lebih besar, atasi masalah berikut:

- Selama operasi penyalinan tabel, dua sesi aktif dipertahankan, menggunakan token keamanan yang berbeda. Jika operasi penyalinan tabel memakan waktu lebih lama dari waktu kedaluwarsa token, Anda harus menerapkan logika untuk menyegarkan token keamanan.
- Jika unit kapasitas baca (RCUs) dan unit kapasitas tulis (WCUs) tidak disediakan, pembacaan atau penulisan pada tabel sumber atau target mungkin terhambat. Pastikan untuk menangkap dan menangani pengecualian ini.
- Tangani kegagalan atau pengecualian lain dan letakkan mekanisme coba lagi untuk mencoba lagi atau melanjutkan dari tempat operasi penyalinan gagal.

Alat

Alat

- Amazon DynamoDB - Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi dengan skalabilitas yang mulus.
- Alat tambahan yang diperlukan akan berbeda berdasarkan bahasa pemrograman yang Anda pilih untuk implementasi. Misalnya, jika Anda menggunakan C #, Anda memerlukan Microsoft Visual Studio dan NuGet paket-paket berikut:
  - `AWSSDK`
  - `AWSSDK.DynamoDBv2`

Kode

Cuplikan kode Python berikut menghapus dan membuat ulang tabel DynamoDB menggunakan perpustakaan Boto3.

Jangan gunakan `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` pengguna IAM karena ini adalah kredensil jangka panjang, yang harus dihindari untuk akses terprogram ke layanan AWS. Untuk informasi selengkapnya tentang kredensil sementara, lihat bagian Praktik terbaik.

Cuplikan kode berikut `AWS_ACCESS_KEY_IDAWS_SECRET_ACCESS_KEY`,, dan `TEMPORARY_SESSION_TOKEN` digunakan dalam cuplikan kode berikut adalah kredensil sementara yang diambil dari AWS Security Token Service (AWS STS).

```
import boto3
import sys
import json

#args = input-parameters = GLOBAL_SEC_INDEXES_JSON_COLLECTION,
 ATTRIBUTES_JSON_COLLECTION, TARGET_DYNAMODB_NAME, TARGET_REGION, ...

#Input param: GLOBAL_SEC_INDEXES_JSON_COLLECTION
#[{"IndexName":"Test-index","KeySchema":[{"AttributeName":"AppId","KeyType":"HASH"},
{"AttributeName":"AppType","KeyType":"RANGE"}],"Projection":
{"ProjectionType":"INCLUDE","NonKeyAttributes":["PK","SK","OwnerName","AppVersion"]}}]

#Input param: ATTRIBUTES_JSON_COLLECTION
#[{"AttributeName":"PK","AttributeType":"S"},
{"AttributeName":"SK","AttributeType":"S"},
{"AttributeName":"AppId","AttributeType":"S"},
{"AttributeName":"AppType","AttributeType":"N"}]
```

```
region = args['TARGET_REGION']
target_ddb_name = args['TARGET_DYNAMODB_NAME']

global_secondary_indexes = json.loads(args['GLOBAL_SEC_INDEXES_JSON_COLLECTION'])
attribute_definitions = json.loads(args['ATTRIBUTES_JSON_COLLECTION'])

# Drop and create target DynamoDB table
dynamodb_client = boto3.Session(
        aws_access_key_id=args['AWS_ACCESS_KEY_ID'],
        aws_secret_access_key=args['AWS_SECRET_ACCESS_KEY'],
        aws_session_token=args['TEMPORARY_SESSION_TOKEN'],
    ).client('dynamodb')

# Delete table
print('Deleting table: ' + target_ddb_name + ' ...')

try:
    dynamodb_client.delete_table(TableName=target_ddb_name)

    #Wait for table deletion to complete
    waiter = dynamodb_client.get_waiter('table_not_exists')
    waiter.wait(TableName=target_ddb_name)
    print('Table deleted.')
except dynamodb_client.exceptions.ResourceNotFoundException:
    print('Table already deleted / does not exist.')
    pass

print('Creating table: ' + target_ddb_name + ' ...')

table = dynamodb_client.create_table(
    TableName=target_ddb_name,
    KeySchema=[
        {
            'AttributeName': 'PK',
            'KeyType': 'HASH'  # Partition key
        },
        {
            'AttributeName': 'SK',
            'KeyType': 'RANGE'  # Sort key
        }
    ],
    AttributeDefinitions=attribute_definitions,
    GlobalSecondaryIndexes=global_secondary_indexes,
    BillingMode='PAY_PER_REQUEST'
```

```
)

waiter = dynamodb_client.get_waiter('table_exists')
waiter.wait(TableName=target_ddb_name)

print('Table created.')
```

Praktik terbaik

Kredensial sementara

Sebagai praktik keamanan terbaik, saat mengakses layanan AWS secara terprogram, hindari menggunakan `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` pengguna IAM karena ini adalah kredensil jangka panjang. Selalu coba gunakan kredensil sementara untuk mengakses layanan AWS secara terprogram.

Sebagai contoh, pengembang hardcode `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` pengguna IAM dalam aplikasi selama pengembangan tetapi gagal menghapus nilai hardcode sebelum mendorong perubahan ke repositori kode. Kredensi yang terbuka ini dapat digunakan oleh pengguna yang tidak diinginkan atau jahat, yang dapat memiliki implikasi serius (terutama jika kredensil yang terekspos memiliki hak istimewa admin). Kredensi yang terbuka ini harus segera dinonaktifkan atau dihapus dengan menggunakan konsol IAM atau AWS Command Line Interface (AWS CLI).

Untuk mendapatkan kredensi sementara untuk akses terprogram ke layanan AWS, gunakan AWS STS. Kredensi sementara hanya berlaku untuk waktu yang ditentukan (dari 15 menit hingga 36 jam). Durasi maksimum yang diizinkan untuk kredensil sementara bervariasi tergantung pada faktor-faktor seperti pengaturan peran dan rantai peran. Untuk informasi selengkapnya tentang AWS STS, lihat dokumentasinya.

Epik

Siapkan tabel DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel DynamoDB.	Buat tabel DynamoDB, dengan indeks, di akun AWS sumber dan target.	Pengembang aplikasi, DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tetapkan penyediaan kapasitas sebagai mode sesuai permintaan, yang memungkinkan DynamoDB untuk read/write menskalakan kapasitas secara dinamis berdasarkan beban kerja. Atau, Anda dapat menggunakan kapasitas yang disediakan dengan 4000 RCUs dan 4000. WCUs	
Isi tabel sumber.	Isi tabel DynamoDB di akun sumber dengan data uji. Memiliki setidaknya 50 MB atau lebih data pengujian membantu Anda melihat puncak dan rata-rata yang RCUs dikonsumsi selama penyalinan tabel. Anda kemudian dapat mengubah penyediaan kapasitas sesuai kebutuhan.	Pengembang aplikasi, DBA, insinyur Migrasi

Siapkan kredensil untuk mengakses tabel DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk mengakses sumber dan target tabel DynamoDB.	Buat peran IAM di akun sumber dengan izin untuk mengakses (baca) tabel DynamoDB di akun sumber.	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tambahkan akun sumber sebagai entitas tepercaya untuk peran ini. Buat peran IAM di akun target dengan izin untuk mengakses (membuat, membaca, memperbarui, menghapus) tabel DynamoDB di akun target. Tambahkan akun target sebagai entitas tepercaya untuk peran ini.	

Salin data tabel dari satu akun ke akun lainnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan kredensi sementara untuk peran IAM.	Dapatkan kredensi sementara untuk peran IAM yang dibuat di akun sumber. Dapatkan kredensi sementara untuk peran IAM yang dibuat di akun target. Salah satu cara untuk mendapatkan kredensil sementara untuk peran IAM adalah dengan menggunakan AWS STS dari AWS CLI. `aws sts assume-role --role-arn arn:aws:iam::<account-id>:r`	Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ole/<role-name> --role-session-name <session-name> --profile <profile-name>` Gunakan profil AWS yang sesuai (sesuai dengan sumber atau akun target). Untuk informasi selengkap nya tentang berbagai cara untuk mendapatkan kredensil sementara, lihat berikut ini: • Referensi API AWS Security Token Service • Mendapatkan kredensi peran IAM untuk akses CLI	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi klien DynamoDB untuk sumber dan target akses DynamoDB.	Inisialisasi klien DynamoDB, yang disediakan oleh AWS SDK, untuk tabel DynamoDB sumber dan target. • Untuk klien DynamoDB sumber, gunakan kredenal sementara yang diambil dari akun sumber. • Untuk klien DynamoDB target, gunakan kredenal sementara yang diambil dari akun target. Untuk informasi selengkapnya tentang membuat permintaan dengan menggunakan kredensil sementara IAM, lihat dokumentasi AWS.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jatuhkan dan buat ulang tabel target.	Hapus dan buat ulang tabel DynamoDB target (bersama dengan indeks) di akun target, menggunakan klien DynamoDB akun target. Menghapus semua catatan dari tabel DynamoDB adalah operasi yang mahal karena mengkonsumsi provision ed. WCUs Menghapus dan membuat ulang tabel menghindari biaya tambahan tersebut. Anda dapat menambahk an indeks ke tabel setelah Anda membuatnya, tetapi ini membutuhkan waktu 2-5 menit lebih lama. Membuat indeks selama pembuatan tabel, dengan meneruskan koleksi indeks ke `createTab le` panggilan, lebih efisien.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan salinan tabel.	Ulangi langkah-langkah berikut hingga semua data disalin: • Lakukan pemindaian pada tabel di akun sumber, menggunakan klien DynamoDB sumber. Setiap pemindaian DynamoDB hanya mengambil 1 MB data dari tabel, jadi Anda harus mengulangi operasi ini sampai semua item, atau catatan, dibaca. • Untuk setiap set item yang dipindai, tulis item ke tabel di akun target, dengan klien DynamoDB target, menggunakan `BatchWriteItem` panggilan di AWS SDK untuk DynamoDB. Ini mengurangi jumlah `PutItem` permintaan yang dibuat ke DynamoDB. • `BatchWriteItem` memiliki batasan 25 menulis atau menempatkan, atau hingga 16 MB. Anda harus menambahkan logika untuk mengakumulasi item yang dipindai dalam hitungan 25 sebelum menelepon. `BatchWriteItem` `BatchWrit`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`eItem` mengembalikan daftar item yang tidak berhasil disalin. Menggunakan daftar ini, tambahkan logika coba lagi untuk melakukan `BatchWriteItem` panggilan lain hanya dengan item yang tidak berhasil. Untuk informasi selengkapnya, lihat implementasi referensi di C # (untuk menjatuhkan, membuat, dan mengisi tabel) di bagian Lampiran. Contoh tabel konfigurasi JavaScript Object Notation (JSON) file juga dilampirkan.	

Sumber daya terkait

- Dokumentasi Amazon DynamoDB
- Membuat pengguna IAM di akun AWS Anda
- AWS SDKs
- Menggunakan kredensil sementara dengan sumber daya AWS

Informasi tambahan

Pola ini diimplementasikan menggunakan C# untuk menyalin tabel DynamoDB dengan 200.000 item (ukuran item rata-rata 5 KB dan ukuran tabel 250 MB). Tabel DynamoDB target disiapkan dengan kapasitas yang disediakan 4000 dan 4000. RCUs WCUs

Operasi penyalinan tabel lengkap (dari akun sumber ke akun target), termasuk menjatuhkan dan membuat ulang tabel, membutuhkan waktu 5 menit. Total unit kapasitas yang dikonsumsi: 30.000 RCUs dan sekitar WCUs 400.000.

Untuk informasi selengkapnya tentang mode kapasitas DynamoDB, lihat Mode kapasitas Baca/Tulis dalam dokumentasi AWS.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat laporan biaya dan penggunaan terperinci untuk Amazon RDS dan Amazon Aurora

Lakshmanan Lakshmanan dan Sudarshan Narasimhan, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara melacak biaya penggunaan untuk Amazon Relational Database Service (Amazon RDS) atau Amazon Aurora cluster dengan mengonfigurasi tag alokasi biaya yang ditentukan pengguna. Anda dapat menggunakan tag ini untuk membuat laporan biaya dan penggunaan terperinci di AWS Cost Explorer untuk klaster di berbagai dimensi. Misalnya, Anda dapat melacak biaya penggunaan di tingkat tim, proyek, atau pusat biaya, lalu menganalisis data di Amazon Athena.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Satu atau lebih contoh Amazon RDS atau Amazon Aurora

Batasan

Untuk pembatasan penandaan, lihat Panduan Pengguna Penagihan AWS.

Arsitektur

Tumpukan teknologi target

- Amazon RDS atau Amazon Aurora
- Laporan Biaya dan Penggunaan AWS
- AWS Cost Explorer
- Amazon Athena

Alur kerja dan arsitektur

Alur kerja penandaan dan analisis terdiri dari langkah-langkah berikut:

1. Insinyur data, administrator database, atau administrator AWS membuat tag alokasi biaya yang ditentukan pengguna untuk klaster Amazon RDS atau Aurora.
2. Administrator AWS mengaktifkan tag.
3. Tag melaporkan metadata ke AWS Cost Explorer.
4. Insinyur data, administrator database, atau administrator AWS membuat laporan alokasi biaya bulanan.
5. Insinyur data, administrator database, atau administrator AWS menganalisis laporan alokasi biaya bulanan dengan menggunakan Amazon Athena.

Diagram berikut menunjukkan cara menerapkan tag untuk melacak biaya penggunaan untuk instans Amazon RDS atau Aurora.

Diagram arsitektur berikut menunjukkan bagaimana laporan alokasi biaya terintegrasi dengan Amazon Athena untuk analisis.

Laporan alokasi biaya bulanan disimpan dalam bucket Amazon S3 yang Anda tentukan. Saat Anda menyiapkan Athena dengan CloudFormation template AWS, seperti yang dijelaskan di bagian Epics, templat tersebut menyediakan beberapa sumber daya tambahan, termasuk crawler AWS Glue, database AWS Glue, peristiwa Amazon Simple Notification System (Amazon SNS), fungsi AWS Lambda, dan peran AWS Identity and Access Management (IAM) untuk fungsi Lambda. Saat file data biaya baru tiba di bucket S3, pemberitahuan acara digunakan untuk meneruskan file ini ke fungsi Lambda untuk diproses. Fungsi Lambda memulai pekerjaan crawler AWS Glue untuk membuat atau memperbarui tabel di AWS Glue Data Catalog. Tabel ini kemudian digunakan untuk query data di Athena.

Alat

- Amazon Athena adalah layanan kueri interaktif yang memudahkan untuk menganalisis data di Amazon S3 menggunakan SQL standar.
- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.

- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- AWS CloudFormation adalah layanan infrastruktur sebagai kode (IAc) yang memungkinkan Anda memodelkan, menyediakan, dan mengelola AWS dan sumber daya pihak ketiga dengan mudah.
- AWS Cost Explorer membantu Anda melihat dan menganalisis biaya dan penggunaan AWS Anda.

Epik

Buat dan aktifkan tag untuk klaster Amazon RDS atau Aurora Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tag alokasi biaya yang ditentukan pengguna untuk Amazon RDS atau klaster Aurora Anda.	Untuk menambahkan tag ke kluster Amazon RDS atau Aurora baru atau yang sudah ada, ikuti petunjuk di Menambahkan, mencantumkan, dan menghapus tag di Panduan Pengguna Amazon Aurora. **Note** Untuk informasi tentang cara menyiapkan klaster Amazon Aurora, lihat petunjuk untuk MySQL dan PostgreSQL di Panduan Pengguna Amazon Aurora.	Administrator AWS, Insinyur data, DBA
Aktifkan tag alokasi biaya yang ditentukan pengguna.	Ikuti petunjuk dalam Mengaktifkan tag alokasi biaya yang ditentukan pengguna	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di Panduan Pengguna Penagihan AWS.	

Membuat laporan biaya dan penggunaan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan laporan biaya dan penggunaan untuk cluster Anda.	1. Masuk ke AWS Management Console dan buka konsol AWS Billing. 2. Di panel navigasi kiri, pilih Laporan Biaya & Penggunaan. 3. Pilih Buat laporan. 4. Berikan nama laporan, simpan pengaturan default untuk opsi lain, lalu pilih Berikutnya. 5. Pilih Konfigurasi dan berikan detail bucket S3 yang ada. Anda juga dapat memilih untuk membuat bucket S3 baru dari layar ini. Pilih Berikutnya. 6. Verifikasi kebijakan default yang akan diterapkan ke bucket Anda, pilih kotak centang konfirmasi, lalu pilih Simpan. 7. Untuk awalan jalur Laporan, tentukan awalan yang ingin Anda tambahkan ke nama laporan.	Pemilik aplikasi, administrator AWS, DBA, AWS Umum, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. Untuk perincian Waktu, pilih Per Jam, Harian atau Bulanan, tergantung seberapa sering Anda ingin data dikumpulkan untuk laporan. 9. Untuk pembuatan versi Laporan, pilih apakah Anda ingin versi laporan baru dibuat secara terpisah atau timpa laporan yang ada dengan setiap versi. 10. Untuk Aktifkan integrasi data laporan, pilih Amazon Athena. Verifikasi bahwa jenis kompresi diatur ke Parket. 11. Pilih Berikutnya. 12. Tinjau pengaturan laporan, lalu pilih Tinjau dan Selesaikan. Data akan tersedia dalam 24 jam.	

Menganalisis data laporan biaya dan penggunaan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menganalisis data laporan biaya dan penggunaan.	1. Siapkan dan gunakan Athena untuk menganalisis data laporan. Untuk petunjuk, lihat Menanyaka	Pemilik aplikasi, administrator AWS, DBA, AWS Umum, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n Laporan Biaya dan Penggunaan menggunakan Amazon Athena di Panduan Pengguna Laporan Biaya dan Penggunaan AWS. Kami menyarankan Anda menggunakan CloudFormation template AWS yang disediakan oleh Athena. 2. Jalankan pertanyaan Athena. Misalnya, Anda dapat menggunakan kueri SQL berikut untuk memeriksa status penyegaran data. `select status from cost_and_usage_data_status` Untuk informasi selengkapnya, lihat Menjalankan kueri Amazon Athena di Panduan Pengguna Laporan Biaya dan Penggunaan AWS. **Note** Ketika Anda menjalankan query SQL Anda, pastikan bahwa database yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	benar dipilih dari daftar dropdown.	

Sumber daya terkait

Referensi

- Menyiapkan Athena menggunakan CloudFormation templat AWS (disarankan)
- Menyiapkan Athena secara manual
- Menjalankan kueri Amazon Athena
- Memuat data laporan ke sumber daya lain

Tutorial dan video

- Analisis Laporan Biaya dan Penggunaan menggunakan Amazon Athena (video) YouTube

Menerapkan instance cluster failover SQL Server di Amazon EC2 dan Amazon FSx dengan menggunakan Terraform

Mark Hudson dan Matt Burgess, Amazon Web Services

Ringkasan

Pola ini menggunakan Terraform untuk menyebarkan instance cluster failover SQL Server () di seluruh node Windows Server Failover Cluster (WSFCFCIs) di Amazon Elastic Compute Cloud (Amazon). EC2 Selain itu, pola menggunakan penyimpanan FSx bersama Amazon untuk data dan file log.

Ketika database SQL Server dimigrasikan ke AWS, pilihan pertama adalah Amazon RDS for SQL Server. Namun, terkadang Amazon RDS for SQL Server tidak cocok dan SQL Server harus digunakan di EC2 Amazon dalam arsitektur yang sangat tersedia. Dalam solusi ini, SQL Server FCIs diinstal di seluruh node WSFC.

Modul Terraform yang disertakan dengan pola ini menyediakan hingga dua instance Amazon EC2 SQL Server. Sistem file Amazon FSx untuk Windows File Server bertindak sebagai saksi kuorum dan menyimpan data bersama dan file log. Terlepas dari jumlah instance yang dikonfigurasi, node instance SQL Server akan selalu membuat dan bergabung dengan cluster FCI untuk memastikan paritas lingkungan. (Biasanya, satu instance dikonfigurasi untuk pengembangan dan dua instance untuk lingkungan produksi.) Untuk konfigurasi yang menggunakan dua node untuk ketersediaan tinggi, Network Load Balancer internal disediakan. Network Load Balancer menggunakan probe kesehatan yang dikonfigurasi pada cluster FCI untuk mengidentifikasi node mana yang utama.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Amazon Virtual Private Cloud (Amazon VPC) dengan dua subnet di Availability Zone terpisah.
- Set opsi Amazon VPC DHCP. Konfigurasikan nama domain untuk menyelesaikan nama domain Active Directory Anda dan domain dan server nama NetBIOS untuk menunjuk ke pengontrol domain Active Directory Anda. Untuk informasi selengkapnya, lihat Konfigurasi VPC di Informasi tambahan.
- AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD).

- Gambar Mesin Amazon Kustom (AMI). Untuk detail selengkapnya, lihat konfigurasi AMI di Informasi tambahan.
- Bucket Amazon Simple Storage Service (Amazon S3) yang berisi image ISO SQL Server. Prasyarat ini hanya diperlukan jika menggunakan EC2 Image Builder dengan `component.yaml` file yang disediakan untuk membuat AMI kustom.
- AWS Key Management Service (AWS KMS) kunci enkripsi.
- Secara default, SQL Server diinstal menggunakan kunci produk edisi pengembang. Sistem produksi diharapkan menggunakan kunci produk yang valid yang diteruskan ke modul oleh variabel yang relevan.

Batasan

- Solusi ini membutuhkan AWS Managed Microsoft AD. Namun, jika Anda mau, Anda dapat menggunakan implementasi Active Directory yang dikelola sendiri sebagai gantinya. Untuk melakukannya, ubah modul Amazon FSx Terraform yang disertakan untuk menghapus atribut. `active_directory_id` Kemudian, tambahkan empat atribut yang diperlukan untuk Active Directory yang dikelola sendiri seperti yang ditunjukkan dalam dokumentasi Terraform.
- SQL Server diatur untuk menggunakan otentikasi mode campuran. Jika mau, Anda dapat menggunakan otentikasi khusus Windows. Untuk melakukannya, dalam skrip data pengguna yang disediakan, hapus `/SECURITYMODE` dan `/SAPWD` parameter yang disediakan ke `setup.exe` perintah. Anda dapat menghapus `sql_accounts.tf` file, dan Anda dapat memodifikasi `instances.tf` file untuk menghapus `sql_sa_password` entri.
- Saat menghapus cluster yang digunakan, Anda harus menghapus objek komputer virtual yang sesuai dan objek komputer individual di Active Directory. Untuk menghapus objek, gunakan alat administratif Direktori Aktif.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

Solusi ini diuji dengan versi berikut:

- Windows Server 2019
- SQL Server 2019

- Terraform v0.13.0

Arsitektur

Tumpukan teknologi sumber

- SQL Server

Tumpukan teknologi target

- SQL Server FCI pada node WSFC menggunakan Amazon EC2
- Amazon FSx untuk Server File Windows
- Buket Amazon S3
- AWS Secrets Manager
- AWS Managed Microsoft AD
- AWS KMS
- AWS Identity and Access Management (IAM)

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk solusi ini.

Diagram menunjukkan yang berikut:

- Peran IAM yang menyediakan akses EC2 instans ke AWS KMS dan Secrets Manager
- Dua node SQL Server digunakan pada EC2 instans Amazon dalam subnet pribadi di dua Availability Zone
- Network Load Balancer untuk memfasilitasi koneksi ke instance SQL Server aktif (tidak digunakan saat menyiapkan cluster node tunggal)
- Amazon FSx untuk sistem file Windows File Server digunakan di kedua subnet pribadi untuk penyimpanan bersama oleh node SQL Server
- Secrets Manager untuk menyimpan kredensil dan konfigurasi Active Directory dan SQL Server
- Bucket Amazon S3 untuk menyimpan gambar instalasi SQL Server
- AWS Managed Microsoft AD untuk otentikasi Windows

- AWS KMS untuk membuat kunci enkripsi

Otomatisasi dan skala

Anda dapat mengotomatiskan penerapan arsitektur target dengan menggunakan modul Terraform yang ada di repositori. GitHub Anda harus memodifikasi `terraform.tfvars` file untuk menyertakan nilai variabel yang spesifik untuk lingkungan Anda. Bucket Amazon S3, AWS Managed Microsoft AD komponen, kunci AWS KMS enkripsi, dan beberapa rahasia adalah prasyarat untuk penerapan ini dan tidak termasuk dalam kode Terraform.

Alat

Layanan AWS

- AWS Directory Service for Microsoft Active Directorymemungkinkan beban kerja dan sumber daya yang sadar direktori Anda AWS untuk menggunakan Microsoft Active Directory di file. AWS Cloud Dalam pola ini, AWS Managed Microsoft AD digunakan untuk otentikasi Windows Server dan SQL Server dan untuk DNS.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah. Dalam pola ini, instance cluster failover SQL Server diinstal pada instans Amazon. EC2
- EC2 Image Builder membantu Anda mengotomatiskan pembuatan, pengelolaan, dan penyebaran gambar server yang disesuaikan.
- Amazon FSx untuk Windows File Server menyediakan penyimpanan bersama yang dikelola sepenuhnya di Windows Server. Dalam pola ini, FSx untuk Windows File Server menyediakan penyimpanan bersama untuk data SQL Server dan file log dan saksi kuorum.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda. Dalam pola ini, ini digunakan untuk mengenkripsi rahasia Secrets Manager, penyimpanan SQL Server pada volume Amazon Elastic Block Store (Amazon EBS), dan FSx sistem file untuk Windows File Server.
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara

terprogram. Dalam pola ini, kredensi Active Directory untuk menginstal dan menjalankan SQL Server, kredensi `sa` pengguna, dan informasi koneksi database disimpan di Secrets Manager.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Pola ini menggunakan bucket Amazon S3 untuk menyimpan image instalasi SQL Server.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat-alat lainnya

- Microsoft SQL Server FCIs diinstal di seluruh node cluster Windows Server. Selain itu, mereka dapat diinstal di beberapa subnet. Dalam pola ini, instance SQL Server FCI diinstal di seluruh node WSFC.
- Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud. Dalam pola ini, Terraform digunakan untuk membuat sumber daya dan mengkonfigurasi instance SQL Server FCI.
- Windows Server Failover Clustering menyediakan fitur infrastruktur yang mendukung ketersediaan tinggi aplikasi server yang dihosting seperti SQL Server. Dalam pola ini, node FCI menggunakan fungsionalitas WSFC untuk menyediakan ketersediaan tinggi lokal melalui redundansi di tingkat instans.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub cluster-amazon-elastic-compute- cloud-amazon-fsx-microsoft -sql-server. Sumber daya berikut tersedia di repositori:

- `README.md`File yang memberikan gambaran umum tentang solusi dan informasi instalasi dan penggunaan tambahan
- Kumpulan dasar file konfigurasi Terraform dan modul FSx khusus Amazon untuk menyediakan komponen untuk pola ini
- Skrip penyiapan instans yang dijalankan sebagai skrip data EC2 pengguna Amazon
- File `component.yaml` yang dapat digunakan Image Builder untuk membuat AMI kustom

Praktik terbaik

Keamanan dan penambalan

- Instalasi dan konfigurasi prasyarat AMI adalah persyaratan minimum untuk menerapkan cluster SQL Server FCI. Perangkat lunak dan konfigurasi tambahan mungkin diperlukan untuk mematuhi standar dan persyaratan keamanan organisasi Anda.
- Setelah penerapan, tambal Windows secara berkelanjutan. Baik langsung menambal instance yang sedang berjalan, atau buat AMI baru dengan tambalan Windows terbaru dan ganti instance (satu per satu) menggunakan AMI baru. AWS merilis Windows baru AMIs setiap bulan yang berisi patch sistem operasi terbaru, driver, dan agen peluncuran. Kami menyarankan Anda memeriksa AMI terbaru saat meluncurkan instans baru, atau saat Anda membuat gambar kustom Anda sendiri.
- EC2 Instans Amazon dikonfigurasi untuk memungkinkan semua lalu lintas keluar. Ketika diterapkan di lingkungan produksi, aturan keluar dalam kelompok keamanan harus diberlakukan untuk membatasi lalu lintas ini ke tujuan yang diperlukan.
- Sistem file FSx untuk Windows File Server dapat secara otomatis merekam log audit untuk berbagi file dan akses file dan folder dan mengirimkannya ke tujuan yang Anda inginkan jika ini adalah persyaratan di lingkungan Anda.
- Putar rahasia Secrets Manager secara otomatis secara teratur. Untuk key pair EC2 instans Amazon, pertimbangkan solusi rotasi otomatis seperti yang dijelaskan dalam Cara menggunakan AWS Secrets Manager untuk menyimpan dan memutar pasangan kunci SSH dengan aman. Untuk kredensi Active Directory dan `sa` rahasia kredensi SQL Server, siapkan rotasi otomatis sesuai dengan kebijakan Anda untuk pengelolaan kata sandi.

Manajemen Direktori Aktif

- Sebagai bagian dari cluster FCI, Windows menghasilkan Computer Name Object (CNO) di Active Directory. CNO merespons permintaan DNS dan meneruskan lalu lintas ke node SQL aktif. Kami tidak menyarankan menggunakan DNS yang disediakan Direktori Aktif ini. TTL terlalu tinggi untuk memberikan waktu failover yang masuk akal dan seringkali membutuhkan waktu lebih dari 5 menit untuk mencerminkan alamat IP utama yang baru. Sebaliknya, untuk instalasi yang sangat tersedia, Network Load Balancer internal dikonfigurasi untuk failover dalam 30 detik.
- Administrator domain Active Directory diperlukan untuk membuat cluster. Persyaratan ini karena izin tinggi yang diperlukan untuk membuat objek cluster dan memodifikasi izin di Active Directory. Namun, layanan SQL Server tidak perlu dijalankan sebagai administrator domain. Oleh karena

itu, kami menyarankan Anda membuat pengguna Active Directory kedua untuk tujuan ini. Namun, Anda dapat menghilangkan pengguna ini jika layanan akan berjalan sebagai pengguna administrator domain. Dalam hal ini, pengguna administrator domain harus ditambahkan ke grup administrator Direktori Aktif yang dibuat sebagai bagian dari pola ini.

Epik

Siapkan kredenal cluster

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup Active Directory.	Di AWS Managed Microsoft AD, buat grup berikut: • Grup administrator cluster - Grup ini akan ditambahkan ke grup Administrator lokal pada setiap node cluster. • Grup desktop jarak jauh cluster - Grup ini akan ditambahkan ke grup Pengguna Desktop Jarak Jauh lokal pada setiap node cluster. Untuk detail selengkapnya, lihat Membuat AWS Managed Microsoft AD grup dalam AWS dokumentasi.	Administrator AD
Buat pengguna Active Directory.	Di AWS Managed Microsoft AD, buat pengguna berikut • Pengguna administrator domain — Gunakan akun ini untuk membuat cluster.	Administrator AD

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Pengguna domain — Layanan SQL Server akan menggunakan akun ini untuk menjalankan. Tambahkan pengguna ini ke grup administrator cluster yang Anda buat di tugas sebelumnya. Untuk detail selengkapnya, lihat Membuat AWS Managed Microsoft AD pengguna dalam AWS dokumentasi.	
Tambahkan kredensil Active Directory ke rahasia.	Gunakan Secrets Manager untuk membuat empat rahasia untuk menyimpan informasi berikut: • Nama pengguna administrator domain • Kata sandi pengguna administrator domain • Nama pengguna domain • Kata sandi pengguna domain Untuk detail selengkapnya, lihat Membuat AWS Secrets Manager rahasia di AWS dokumentasi.	Administrator AWS

Siapkan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AMI Windows.	Buat AMI Windows kustom yang mencakup perangkat lunak dan konfigurasi prasyarat. Untuk detail selengkapnya, lihat Informasi Tambahan.	Administrator AWS, AWS DevOps
Instal Terraform.	Untuk menginstal Terraform , ikuti instruksi di situs web Terraform.	AWS DevOps
Kloning repositori.	Kloning repositori pola ini. Untuk detail selengkapnya, lihat Mengkloning repositori di situs web. GitHub	AWS DevOps

Instal cluster

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah variabel Terraform.	Perbarui `terraform .tfvars` file yang disediakan untuk mengatur semua variabel ke nilai yang sesuai untuk lingkungan Anda. Misalnya, perbarui `domain_group_rdp_users` variabel `domain_group_administrators` dan untuk menggunakan nama domain Active Directory dan nama grup	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Active Directory yang dibuat sebelumnya.	
Inisialisasi Terraform.	Untuk melihat penerapan yang diusulkan, arahkan ke root repositori. Gunakan antarmuka baris perintah Terraform (CLI) untuk menjalankan dan kemudian menjalankan`terraform init`. `terraform plan`	AWS DevOps
Menyebarkan sumber daya.	Untuk menerapkan cluster SQL dan sumber daya terkait, gunakan Terraform CLI untuk menjalankannya. `terraform apply`	AWS DevOps, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan.	Untuk memvalidasi penerapan , gunakan langkah-langkah berikut: 1. Connect ke salah satu EC2 instans Windows Amazon yang digunakan menggunakan Remote Desktop. 2. Untuk detail selengkapnya, lihat Connect to Windows Anda menggunakan RDP dalam dokumentasi. AWS 3. Buka Windows Failover Cluster Manager. Validasi bahwa cluster telah dibuat, dan bahwa peran SQL Server dibuat dan sedang berjalan. 4. Untuk menguji konektivi tas dan otentikasi dengan SQL Server, gunakan alat database seperti SQL Server Managemen t Studio untuk terhubung ke titik akhir SQL Server. (Nilai endpoint disimpan di Secrets Manager.) Untuk detail selengkapnya, lihat Connect to Microsoft SQL Server di Amazon EC2 dalam AWS dokumentasi.	DBA, administrator sistem AWS

Pemecahan masalah

Isu	Solusi
Penyediaan Terraform selesai tetapi Windows Failover Cluster Manager tidak menunjukkan cluster telah dibuat atau cluster dalam keadaan tidak dapat dioperasikan.	Seluruh instalasi sumber daya dan konfigurasi cluster dapat memakan waktu 45-60 menit. Setelah Terraform selesai, skrip data pengguna harus berjalan hingga selesai, yang memerlukan beberapa reboot. Untuk memantau kemajuan, Anda dapat menggunakan `Checkpoints` direktori di `C:\` drive dan log instalasi SQL Server. `C:\Program Data\Microsoft SQL Server\150\Log` Setelah selesai, pesan Instalasi Lengkap tersedia dalam `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\UserdataExecution.log` file.
Setelah menyediakan cluster yang berfungsi, menggunakan Terraform untuk menghapus dan membuat ulang cluster tidak berhasil. Terraform selesai, tetapi cluster tidak diatur dengan benar.	Bagian dari proses penyediaan melibatkan mendaftarkan mesin dan objek virtual ke Active Directory dan Active Directory DNS. Ketika nama komputer ada untuk node EC2 cluster Amazon dan node cluster, FCI tidak dapat menginisialisasi dengan benar dan akan gagal penyediaan. Untuk memperbaiki masalah ini, lakukan langkah-langkah berikut: 1. Hapus EC2 node Amazon, komputer virtual cluster, dan id cluster di pengguna dan komputer Active Directory. 2. Hapus entri DNS untuk komputer virtual cluster di Active Directory DNS. 3. Jalankan perintah berikut untuk menghapus cluster id string acak sumber daya Terraform. `terragrunt destroy -target=r`

Isu	Solusi
	`andom_string.cluster_id` Tindakan ini akan menghapus EC2 instans Amazon yang ada. 4. Jalankan`terraform apply`, dan harapkan tiga sumber daya baru berikut: Dua EC2 instans Amazon FCI dan 1 id cluster string acak.

Sumber daya terkait

AWS dokumentasi

- Buat gambar kustom dengan Image Builder
- Buat kunci KMS
- Membuat ember tujuan umum
- Membuat kebijakan utama
- Menciptakan AWS Managed Microsoft AD

Informasi tambahan

Informasi modul Terraform

Modul ini menggunakan campuran konfigurasi AMI dan konfigurasi data pengguna untuk mendapatkan perpaduan yang baik antara waktu penyediaan dan stabilitas. Selama penyediaan, Windows memerlukan beberapa restart dan menunggu. Metode pos pemeriksaan telah diterapkan untuk melindungi terhadap loop tak terbatas selama restart data pengguna persisten. Data pengguna dikonfigurasi agar persisten. Oleh karena itu, skrip konfigurasi data pengguna memiliki, dan harus melanjutkan, untuk dikembangkan menjadi idempoten. Idempotency merampingkan proses pembaruan, memungkinkan instance ditukar selama siklus pembaruan tanpa konfigurasi manual untuk bergabung kembali atau membuat ulang cluster FCI.

String koneksi SQL Server dan pengelompokan failover

Modul akan menerbitkan rahasia yang berisi alamat titik akhir yang harus digunakan dalam string koneksi untuk database ini. Nama rahasia mengikuti format`{environment_name}/sqlserver/`

`{cluster_name}/endpoint`. Untuk instalasi di mana hanya satu node yang digunakan, Anda dapat mengharapkan ini menjadi alamat IP dari antarmuka SQL Server Amazon EC2 instance. Untuk instalasi ketersediaan tinggi (dua contoh), Anda dapat mengharapkan ini menjadi nama DNS dari Network Load Balancer internal.

Virtual pengelompokan failover IPs tidak didukung dalam modul ini. IP virtual harus tetap berada di subnet yang sama agar dapat berfungsi. Di AWS, satu subnet tidak dapat menjangkau beberapa Availability Zone. Oleh karena itu, penggunaan virtual IPs akan menghilangkan kemampuan modul ini untuk dianggap sangat tersedia.

Setiap EC2 instans Amazon diberikan tiga alamat IP pribadi. USAI mereka sebagai berikut:

- IP utama untuk lalu lintas jaringan — IP sumber untuk lalu lintas keluar.
- Komunikasi FCI — Digunakan untuk mempertahankan status dan sinkronisasi cluster failover.
- SQL Server (port TCP 1433) - Pendengar dan juga mendengarkan lalu lintas detak jantung untuk menentukan instance mana yang utama.

Konfigurasi VPC

Prasyarat mencantumkan set opsi DHCP yang dikonfigurasi untuk menggunakan Active Directory untuk resolusi DNS. Namun, prasyarat ini bukanlah persyaratan yang sulit. Persyaratan yang sulit adalah bahwa EC2 instance harus dapat menyelesaikan nama domain Active Directory Anda. Memenuhi persyaratan ini dapat dicapai dengan cara lain, seperti dengan menggunakan titik Amazon Route 53 Resolver akhir. Untuk informasi selengkapnya, lihat Mengintegrasikan resolusi DNS Layanan Direktori Anda dengan Amazon Route 53 Resolvers (AWS posting Blog).

Konfigurasi AMI

AMI yang digunakan dalam pola ini harus berisi perangkat lunak dan konfigurasi prasyarat berikut:

1. Unduh dan perluas file instalasi SQL Server 2019 ke dalam`C:\SQL_Install_media`.
2. Instal fitur Windows berikut:
   - `Install-WindowsFeature Failover-Clustering`
   - `Install-WindowsFeature RSAT-AD-PowerShell`
   - `Install-WindowsFeature RSAT-AD-Tools`
   - `Install-WindowsFeature RSAT-Clustering-Mgmt`
   - `Install-WindowsFeature RSAT-Clustering-PowerShell`

- `Install-WindowsFeature RSAT-Clustering-CmdInterface`

3. Nonaktifkan firewall Windows sebagai berikut:
   - `Get-NetFirewallProfile | Set-NetFirewallProfile -Enabled False`
4. Aktifkan metode otentikasi CredSSP (ganti <domain> dengan nama domain Windows organisasi Anda) sebagai berikut:
   - `Enable-WSManCredSSP -Role "Server" -Force`
   - `Enable-WSManCredSSP -Role "Client" -DelegateComputer *.<domain>.com -Force`
5. Mengatur kunci registri berikut:
   - Izinkan kredensi otentikasi NTLM:
     - `HKLM:\Software\Policies\Microsoft\Windows\CredentialsDelegation`
       - Nama: `AllowFreshCredentialsWhenNTLMOnly`
       - Nilai: 1
       - Tipe: `REG_DWORD`
   - Izinkan komputer domain lokal menggunakan NTLM dari: PowerShell
     - Jalan: `HKLM:\Software\Policies\Microsoft\Windows\CredentialsDelegation\AllowFreshCredentialsWhenNTLMOnly`
       - Nama: 1
       - Nilai: `wsman/*.<domain>.com`
       - Tipe: `REG_SZ`
6. Siapkan PowerShell Galeri sebagai berikut:
   - `[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12`
   - `Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force`
   - `Set-PSRepository -Name PSGallery -InstallationPolicy Trusted`
7. Instal PowerShell modul Windows berikut*:
   - `Install-Module -Name ComputerManagementDsc`
   - `Install-Module -Name FailOverClusterDsc`
   - `Install-Module -Name PSDscResources`
   - `Install-Module -Name xSmbShare`
   - `Install-Module -Name xActiveDirectory`

- `Install-Module -Name SqlServer`

Untuk menggunakan Image Builder untuk membuat AMI, ikuti petunjuk di Membuat pipeline gambar menggunakan wizard EC2 Image Builderconsole dalam dokumentasi Image Builder. Untuk membuat komponen resep dengan prasyarat sebelumnya, gunakan langkah-langkah berikut:

1. Unduh file component.yaml dari `ami` folder repositori. GitHub
2. Salin konten ke komponen Image Builder baru.
3. Perbarui placeholder berikut dengan informasi Anda:
   - `<domain>`— Nama domain Direktori Aktif Anda
   - `<bucket_name>`— Nama bucket Amazon S3 yang berisi gambar SQL Server

Meniru array PL/SQL asosiatif Oracle di Amazon Aurora PostgreSQL dan Amazon RDS untuk PostgreSQL

Rajkumar Raghuwanshi, Bhanu Ganesh Gudivada, dan Sachin Khanna, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara meniru array PL/SQL asosiatif Oracle dengan posisi indeks kosong di Amazon Aurora PostgreSQL dan Amazon RDS untuk lingkungan PostgreSQL. Ini juga menjelaskan beberapa perbedaan antara array PL/SQL asosiatif Oracle dan array PostgreSQL sehubungan dengan bagaimana masing-masing menangani posisi indeks kosong selama migrasi.

Kami menyediakan alternatif PostgreSQL untuk `aws_oracle_ext` menggunakan fungsi untuk menangani posisi indeks kosong saat memigrasi database Oracle. Pola ini menggunakan kolom tambahan untuk menyimpan posisi indeks, dan mempertahankan penanganan Oracle terhadap array jarang sambil menggabungkan kemampuan PostgreSQL asli.

Oracle

Di Oracle, koleksi dapat diinisialisasi sebagai kosong dan diisi menggunakan metode `EXTEND` pengumpulan, yang menambahkan `NULL` elemen ke array. Saat bekerja dengan array PL/SQL asosiatif yang diindeks oleh`PLS_INTEGER`, `EXTEND` metode ini menambahkan `NULL` elemen secara berurutan, tetapi elemen juga dapat diinisialisasi pada posisi indeks nonsequential. Posisi indeks apa pun yang tidak diinisialisasi secara eksplisit tetap kosong.

Fleksibilitas ini memungkinkan struktur array jarang di mana elemen dapat diisi pada posisi arbitrer. Saat melakukan iterasi melalui koleksi menggunakan `FOR LOOP` with `FIRST` dan `LAST` bounds, hanya elemen yang diinisialisasi (baik `NULL` atau dengan nilai yang ditentukan) yang diproses, sementara posisi kosong dilewati.

PostgreSQL (Amazon Aurora dan Amazon RDS)

PostgreSQL menangani nilai kosong secara berbeda dari nilai. `NULL` Ini menyimpan nilai kosong sebagai entitas berbeda yang menggunakan satu byte penyimpanan. Ketika sebuah array memiliki nilai kosong, PostgreSQL menetapkan posisi indeks sekuensial seperti nilai non-kosong. Tetapi pengindeksan sekuensial membutuhkan pemrosesan tambahan karena sistem harus mengulangi semua posisi yang diindeks, termasuk yang kosong. Ini membuat pembuatan array tradisional tidak efisien untuk kumpulan data yang jarang.

AWS Schema Conversion Tool

The AWS Schema Conversion Tool (AWS SCT) biasanya menangani Oracle-to-PostgreSQL migrasi menggunakan `aws_oracle_ext` fungsi. Dalam pola ini, kami mengusulkan pendekatan alternatif yang menggunakan kemampuan PostgreSQL asli, yang menggabungkan tipe array PostgreSQL dengan kolom tambahan untuk menyimpan posisi indeks. Sistem kemudian dapat mengulangi melalui array hanya menggunakan kolom indeks.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Pengguna AWS Identity and Access Management (IAM) dengan izin administrator.
- Instance yang kompatibel dengan Amazon RDS atau Aurora PostgreSQL.
- Pemahaman dasar tentang database relasional.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

Pola ini diuji dengan versi berikut:

- Amazon Aurora PostgreSQL 13.3
- Amazon RDS untuk PostgreSQL 13.3
- AWS SCT 1.0.674
- Oracle 12c EE 12.2

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal

Tumpukan teknologi target

- Amazon Aurora PostgreSQL
- Amazon RDS for PostgreSQL

Arsitektur target

Diagram menunjukkan yang berikut:

- Sumber Amazon RDS for Oracle database instance
- EC2 Instans Amazon dengan AWS SCT untuk mengonversi fungsi Oracle ke PostgreSQL yang setara
- Database target yang kompatibel dengan Amazon Aurora PostgreSQL

Alat

Layanan AWS

- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.
- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud
- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di file. AWS Cloud
- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di file. AWS Cloud
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Alat lainnya

- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi yang menyederhanakan pengembangan dan pengelolaan database Oracle baik dalam penerapan tradisional maupun berbasis cloud.
- pgAdmin adalah alat manajemen open source untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database. Dalam pola ini, pgAdmin terhubung ke RDS untuk instance database PostgreSQL dan menanyakan data. Atau, Anda dapat menggunakan klien baris perintah psql.

Praktik terbaik

- Uji batas set data dan skenario tepi.
- Pertimbangkan untuk menerapkan penanganan kesalahan untuk kondisi out-of-bounds indeks.
- Optimalkan kueri untuk menghindari pemindaian kumpulan data yang jarang.

Epik

Perilaku array asosiatif Oracle (sumber)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat PL/SQL blok sumber di Oracle.	Buat PL/SQL blok sumber di Oracle yang menggunakan array asosiatif berikut: `DECLARE TYPE country_codes IS TABLE OF VARCHAR2(100) INDEX BY pls_integer; cc country_codes; cc_idx NUMBER := NULL; BEGIN cc(7) := 'India'; cc(3) := 'UK'; cc(5) := 'USA';`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> cc(0) := 'China';
	    cc(-2) := 'Invalid'
	;
	    dbms_output.put_li
	ne('cc_length:' ||
	 cc.COUNT);
	    IF (cc.COUNT > 0)
	 THEN
	        cc_idx :=
	 cc.FIRST;
	        FOR i IN
	 1..cc.COUNT LOOP
	            dbms_outp
	ut.put_line('cc_id
	x:' || cc_idx || '
	 country:' || cc(cc_idx
	));
	            cc_idx :=
	 cc.next(cc_idx);
	        END LOOP;
	    END IF;
	END;</pre>	

Jalankan PL/SQL blok.	Jalankan PL/SQL blok sumber di Oracle. Jika ada kesenjang an antara nilai indeks dari array asosiatif, tidak ada data yang disimpan dalam celah tersebut. Hal ini memungkin kan loop Oracle untuk iterasi hanya melalui posisi indeks.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tinjau output.	Lima elemen dimasukkan ke dalam array (cc) pada interval yang tidak berurutan. Jumlah array ditunjukkan pada output berikut:
	`cc_length:5`
	`cc_idx:-2 country:I`
	`nvalid`
	`cc_idx:0 country:China`
	`cc_idx:3 country:UK`
	`cc_idx:5 country:USA`
	`cc_idx:7 country:India`
		DBA

Perilaku array asosiatif PostgreSQL (target)

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat PL/pgSQL blok target di PostgreSQL.	Buat PL/pgSQL blok target di PostgreSQL yang menggunakan array asosiatif berikut:
	`DO $$`
	`DECLARE`
	`    cc character`
	` varying(100)[];`
	`    cc_idx integer :=`
	` NULL;`
	`BEGIN`
	`    cc[7] := 'India';`
	`    cc[3] := 'UK';`
	`    cc[5] := 'USA';`
	`    cc[0] := 'China';`
	`    cc[-2] := 'Invalid'`
	`;`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> RAISE NOTICE
	 'cc_length: %',
	 ARRAY_LENGTH(cc, 1);
	    IF (ARRAY_LE
	NGTH(cc, 1) > 0) THEN
	        FOR i IN
	 ARRAY_LOWER(cc,
	 1)..ARRAY_UPPER(cc,
	 1)
	        LOOP
	            RAISE NOTICE
	 'cc_idx:% country:%',
	 i, cc[i];
	        END LOOP;
	    END IF;
	END;
	$$;</pre>	

Jalankan PL/pgSQL blok.	Jalankan PL/pgSQL blok target di PostgreSQL. Jika ada kesenjangan antara nilai indeks dari array asosiatif, tidak ada data yang disimpan dalam celah tersebut. Hal ini memungkinkan loop Oracle untuk iterasi hanya melalui posisi indeks.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tinjau output.	Panjang array lebih besar dari 5 karena NULL disimpan dalam celah antara posisi indeks. Seperti yang ditunjukkan pada output berikut, loop menyelesaikan 10 iterasi untuk mengambil 5 nilai dalam array.
	`cc_length:10`
	`cc_idx:-2 country:Invalid`
	`cc_idx:-1 country:<NULL>`
	`cc_idx:0 country:China`
	`cc_idx:1 country:<NULL>`
	`cc_idx:2 country:<NULL>`
	`cc_idx:3 country:UK`
	`cc_idx:4 country:<NULL>`
	`cc_idx:5 country:USA`
	`cc_idx:6 country:<NULL>`
	`cc_idx:7 country:India`
		DBA

Meniru perilaku array asosiatif Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat PL/pgSQL blok target dengan array dan tipe yang ditentukan pengguna.	Untuk mengoptimalkan kinerja dan mencocokkan fungsionalitas Oracle, kita dapat membuat tipe yang ditentukan pengguna yang	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menyimpan posisi indeks dan data yang sesuai. Pendekatan ini mengurangi iterasi yang tidak perlu dengan mempertahankan asosiasi langsung antara indeks dan nilai. `DO $$ DECLARE cc country_codes[]; cc_append country_codes := NULL; i record; BEGIN cc_append.idx = 7; cc_append.val = 'India'; cc := array_append(cc, cc_append); cc_append.idx = 3; cc_append.val = 'UK'; cc := array_append(cc, cc_append); cc_append.idx = 5; cc_append.val = 'USA'; cc := array_append(cc, cc_append); cc_append.idx = 0; cc_append.val = 'China'; cc := array_append(cc, cc_append); cc_append.idx = -2; cc_append.val = 'Invalid'; cc := array_append(cc, cc_append);`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>RAISE NOTICE 'cc_length: %', ARRAY_LENGTH(cc, 1);
	IF (ARRAY_LENGTH(cc, 1) > 0) THEN
	FOR i IN (
	SELECT
	*
	FROM
	unnest(cc)
	ORDER BY
	idx)
	LOOP
	RAISE NOTICE 'cc_idx: % country:%', i.idx, i.val;
	END LOOP;
	END IF;
	END;
	$$;</pre>	

Jalankan PL/pgSQL blok.	Jalankan PL/pgSQL blok target. Jika ada kesenjangan antara nilai indeks dari array asosiatif, tidak ada data yang disimpan dalam celah tersebut. Hal ini memungkinkan loop Oracle untuk iterasi hanya melalui posisi indeks.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tinjau output.	Seperti yang ditunjukkan pada output berikut, tipe yang ditentukan pengguna hanya menyimpan elemen data yang terisi, yang berarti bahwa panjang array cocok dengan jumlah nilai. Akibatnya , L00P iterasi dioptimalkan untuk memproses hanya data yang ada, menghilangkan kebutuhan untuk melacak posisi kosong.
	`cc_length:5`
	`cc_idx:-2 country:I`
	`nvalid`
	`cc_idx:0 country:China`
	`cc_idx:3 country:UK`
	`cc_idx:5 country:USA`
	`cc_idx:7 country:India`
		DBA

Sumber daya terkait

AWS dokumentasi

- AWS blog basis data
- Buku pedoman migrasi Oracle ke Aurora PostgreSQL

Dokumentasi lainnya

- Array asosiatif Oracle
- Fungsi dan operator array PostgreSQL
- PostgreSQL tipe yang ditentukan pengguna

Aktifkan koneksi terenkripsi untuk instans PostgreSQL DB di Amazon RDS

Rohit Kapoor, Amazon Web Services

Ringkasan

Amazon Relational Database Service (Amazon RDS) mendukung enkripsi SSL untuk instans PostgreSQL DB. Menggunakan SSL, Anda dapat mengenkripsi koneksi PostgreSQL antara aplikasi dan Amazon RDS for PostgreSQL DB instans. Secara default, Amazon RDS SSL/TLS for PostgreSQL menggunakan dan mengharapkan semua klien terhubung dengan menggunakan enkripsi. SSL/TLS Amazon RDS for PostgreSQL mendukung TLS versi 1.1 dan 1.2.

Pola ini menjelaskan bagaimana Anda dapat mengaktifkan koneksi terenkripsi untuk instans Amazon RDS for PostgreSQL DB. Anda dapat menggunakan proses yang sama untuk mengaktifkan koneksi terenkripsi untuk Amazon Aurora PostgreSQL Edisi yang kompatibel.

Prasyarat dan batasan

- Akun AWS yang aktif
- Sebuah Amazon RDS untuk instans DB Amazon RDS for PostgreSQL
- Bundel SSL

Arsitektur

Alat

- pgAdmin adalah platform administrasi dan pengembangan open-source untuk PostgreSQL. Anda dapat menggunakan pgAdmin di Linux, Unix, macOS, dan Windows untuk mengelola objek database Anda di PostgreSQL 10 dan yang lebih baru.
- Editor PostgreSQL menyediakan antarmuka yang lebih ramah pengguna untuk membantu Anda membuat, mengembangkan, dan menjalankan kueri, dan mengedit kode sesuai dengan kebutuhan Anda.

Praktik terbaik

- Pantau koneksi database yang tidak aman.

- Audit hak akses basis data.
- Pastikan bahwa backup dan snapshot dienkripsi saat istirahat.
- Pantau akses basis data.
- Hindari grup akses yang tidak terbatas.
- Tingkatkan notifikasi Anda dengan Amazon GuardDuty.
- Pantau kepatuhan kebijakan secara teratur.

Epik

Unduh sertifikat tepercaya dan impor ke toko kepercayaan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Muat sertifikat tepercaya ke komputer Anda.	Untuk menambahkan sertifikat ke toko Otoritas Sertifikasi Root Tepercaya untuk komputer Anda, ikuti langkah-langkah berikut. (Instruksi ini menggunakan Window Server sebagai contoh.) 1. Di Windows Server, pilih Mulai, Jalankan, lalu ketik mmc. 2. Di konsol, pilih File, Tambah/Hapus Snap-in. 3. Di bawah Snap-in yang tersedia, pilih Sertifikat, lalu pilih Tambah. 4. Di bawah Snap-in ini akan selalu mengelola sertifikat untuk, pilih Akun komputer, Berikutnya. 5. Pilih Komputer lokal, Selesai.	DevOps insinyur, insinyur migrasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Jika Anda tidak memiliki snap-in lagi untuk ditambahkan ke konsol, pilih OK. 7. Di pohon konsol, klik dua kali Sertifikat. 8. Klik kanan Otoritas Sertifikasi Root Tepercaya. 9. Pilih Semua Tugas, Impor untuk mengimpor sertifikat yang diunduh. 10. Ikuti langkah-langkah di Wisaya Impor Sertifikat.	

Paksa koneksi SSL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup parameter dan atur parameter rds.force_ssl.	Jika instance PostgreSQL DB memiliki grup parameter kustom, edit grup parameter dan ubah ke 1. `rds.force_ssl` Jika instans DB menggunakan grup parameter default yang belum `rds.force_ssl` diaktifkan, buat grup parameter baru. Anda dapat memodifikasi grup parameter baru dengan menggunakan Amazon RDS API atau secara manual seperti pada petunjuk berikut.	DevOps insinyur, insinyur migrasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk membuat grup parameter baru: 1. Masuk ke konsol AWS Management dan buka konsol Amazon RDS untuk Wilayah AWS yang menghosting instans DB. 2. Di panel navigasi, pilih Grup parameter. 3. Pilih Buat grup parameter, dan atur nilai berikut: • Untuk keluarga grup Parameter, pilih postgres14. • <database_instance >Untuk nama Grup, ketik pgsql- -ssl. • Untuk Deskripsi, masukkan deskripsi bentuk bebas untuk grup parameter yang Anda tambahkan. • Pilih Buat. 4. Pilih grup parameter yang Anda buat. 5. Dari Tindakan grup parameter, pilih Edit. 6. Temukan rds.force_ssl dan ubah pengaturannya menjadi 1.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Lakukan pengujian sisi klien sebelum mengubah parameter ini. 7. Pilih Simpan perubahan. Untuk mengaitkan grup parameter dengan instance PostgreSQL DB Anda: 1. Di konsol Amazon RDS, di panel navigasi, pilih Databases, lalu pilih instans PostgreSQL DB. 2. Pilih Ubah. 3. Di bawah Konfigurasi tambahan, pilih grup parameter baru, lalu pilih Lanjutkan. 4. Di bawah Jadwalkan modifikasi, pilih Terapkan segera. 5. Pilih Ubah instans DB. Untuk informasi selengkapnya, lihat dokumentasi Amazon RDS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memaksa koneksi SSL.	Connect ke instans Amazon RDS for PostgreSQL DB. Upaya koneksi yang tidak menggunakan SSL ditolak dengan pesan kesalahan. Untuk informasi selengkapnya, lihat dokumentasi Amazon RDS.	DevOps insinyur, insinyur migrasi, DBA

Instal ekstensi SSL

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal ekstensi SSL.	1. Luncurkan koneksi psql atau pgAdmin sebagai DBA.
	2. Panggil fungsi ssl_is_used () untuk menentukan apakah SSL sedang digunakan.
	`select ssl_is_used();`
	Fungsi kembali `t` jika koneksi menggunakan SSL; jika tidak, ia kembali`f`.
	3. Instal ekstensi SSL.
	`create extension sslinfo;`
	`show ssl;`
	`select ssl_cipher();`
		DevOps insinyur, insinyur migrasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya, lihat dokumentasi Amazon RDS.	

Konfigurasikan klien PostgreSQL Anda untuk SSL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan klien untuk SSL.	Dengan menggunakan SSL, Anda dapat memulai server PostgreSQL dengan dukungan untuk koneksi terenkripsi yang menggunakan protokol TLS. Server mendengarkan koneksi standar dan SSL pada port TCP yang sama, dan bernegosiasi dengan klien penghubung apakah akan menggunakan SSL. Secara default, ini adalah opsi klien. Jika Anda menggunakan klien psql: 1. Pastikan sertifikat Amazon RDS telah dimuat ke komputer lokal Anda. 2. Luncurkan koneksi klien SSL dengan menambahkan yang berikut ini: `psql postgres -h SOMEHOST.amazonaws.com -p 8192 -U`	DevOps insinyur, insinyur migrasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>someuser sslmode=verify-full sslrootcert=rds-ssl-ca-cert.pem
	select ssl_cipher();</pre>
	Untuk klien PostgreSQL lainnya:
	• Ubah parameter kunci publik aplikasi masing-masing. Ini mungkin tersedia sebagai opsi, sebagai bagian dari string koneksi Anda, atau sebagai properti di halaman koneksi di alat GUI.
	
	Tinjau halaman berikut untuk klien ini:
	• Dokumentasi pgAdmin
	• Dokumentasi JDBC	

Pemecahan masalah

Isu	Solusi
Tidak dapat mengunduh sertifikat SSL.	Periksa koneksi Anda ke situs web, dan coba lagi unduh sertifikat ke komputer lokal Anda.

Sumber daya terkait

- Amazon RDS untuk dokumentasi PostgreSQL

- Menggunakan SSL dengan instance PostgreSQL DB (dokumentasi Amazon RDS)
- TCP/IP Koneksi Aman dengan SSL (dokumentasi PostgreSQL)
- Menggunakan SSL (dokumentasi JDBC)

Enkripsi Amazon RDS yang ada untuk instans DB Amazon RDS for PostgreSQL

Piyush Goyal, Shobana Raghu, dan Yaser Raja, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara mengenkripsi Amazon Relational Database Service (Amazon RDS) yang ada untuk instans PostgreSQL DB di AWS Cloud dengan waktu henti minimal. Proses ini juga berfungsi untuk Amazon RDS for MySQL DB instans.

Anda dapat mengaktifkan enkripsi untuk instans Amazon RDS DB saat Anda membuatnya, tetapi tidak setelah dibuat. Namun, Anda dapat menambahkan enkripsi ke instans DB yang tidak terenkripsi dengan membuat snapshot dari instans DB Anda, dan kemudian membuat salinan terenkripsi dari snapshot itu. Anda kemudian dapat memulihkan instans DB dari snapshot terenkripsi untuk mendapatkan salinan terenkripsi dari instans DB asli Anda. Jika proyek Anda memungkinkan downtime (setidaknya untuk transaksi tulis) selama aktivitas ini, hanya ini yang perlu Anda lakukan. Ketika salinan instans DB yang baru dan terenkripsi tersedia, Anda dapat mengarahkan aplikasi Anda ke database baru. Namun, jika proyek Anda tidak mengizinkan waktu henti yang signifikan untuk aktivitas ini, Anda memerlukan pendekatan alternatif yang membantu meminimalkan waktu henti. Pola ini menggunakan AWS Database Migration Service (AWS DMS) untuk bermigrasi dan terus mereplikasi data sehingga cutover ke database baru yang terenkripsi dapat dilakukan dengan waktu henti minimal.

Instans DB terenkripsi Amazon RDS menggunakan algoritme enkripsi AES-256 standar industri untuk mengenkripsi data Anda di server yang menghosting instans Amazon RDS DB Anda. Setelah data Anda dienkripsi, Amazon RDS menangani otentikasi akses dan dekripsi data Anda secara transparan, dengan dampak minimal pada kinerja. Anda tidak perlu memodifikasi aplikasi klien basis data untuk menggunakan enkripsi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Amazon RDS yang tidak terenkripsi untuk instans DB Amazon RDS for PostgreSQL
- Pengalaman bekerja dengan (membuat, memodifikasi, atau menghentikan) tugas AWS DMS (lihat Bekerja dengan tugas AWS DMS dalam dokumentasi AWS DMS)

- Keakraban dengan AWS Key Management Service (AWS KMS) untuk mengenkripsi database (lihat dokumentasi AWS KMS)

Batasan

- Anda dapat mengaktifkan enkripsi untuk instans Amazon RDS DB hanya ketika Anda membuatnya, bukan setelah instans DB dibuat.
- Data dalam tabel yang tidak tercatat tidak akan dipulihkan menggunakan snapshot. Untuk informasi lebih lanjut, tinjau Praktik terbaik untuk bekerja dengan PostgreSQL.
- Anda tidak dapat memiliki replika baca terenkripsi dari instans DB yang tidak dienkripsi atau replika baca yang tidak dienkripsi dari instans DB terenkripsi.
- Anda tidak dapat memulihkan cadangan atau snapshot yang tidak terenkripsi ke instans DB terenkripsi.
- AWS DMS tidak secara otomatis mentransfer Urutan sehingga diperlukan langkah-langkah tambahan untuk menangani hal ini.

Untuk informasi selengkapnya, lihat Batasan instans DB terenkripsi Amazon RDS dalam dokumentasi Amazon RDS.

Arsitektur

Arsitektur sumber

- Instans RDS DB yang tidak terenkripsi

Arsitektur target

- Instans RDS DB terenkripsi
  - Instans RDS DB tujuan dibuat dengan memulihkan salinan snapshot DB dari instans RDS DB sumber.
  - Kunci AWS KMS digunakan untuk enkripsi saat memulihkan snapshot.
  - Tugas replikasi AWS DMS digunakan untuk memigrasikan data.

Alat

Alat yang digunakan untuk mengaktifkan enkripsi:

- Kunci AWS KMS untuk enkripsi — Saat membuat instans DB terenkripsi, Anda dapat memilih kunci yang dikelola pelanggan atau kunci terkelola AWS untuk Amazon RDS untuk mengenkripsi instans DB Anda. Jika Anda tidak menentukan pengenal kunci untuk kunci yang dikelola pelanggan, Amazon RDS menggunakan kunci terkelola AWS untuk instans DB baru Anda. Amazon RDS membuat kunci terkelola AWS untuk Amazon RDS untuk akun AWS Anda. Akun AWS Anda memiliki kunci terkelola AWS yang berbeda untuk Amazon RDS untuk setiap Wilayah AWS. Untuk informasi selengkapnya tentang menggunakan kunci KMS untuk enkripsi Amazon RDS, lihat Mengenkripsi Sumber Daya Amazon RDS.

Alat yang digunakan untuk replikasi berkelanjutan:

- AWS DMS — Anda dapat menggunakan AWS Database Migration Service (AWS DMS) untuk mereplikasi perubahan dari DB sumber ke DB target. Penting untuk menjaga sumber dan target DB tetap sinkron untuk meminimalkan waktu henti. Untuk informasi tentang menyiapkan AWS DMS dan membuat tugas, lihat dokumentasi AWS DMS.

Epik

Buat snapshot dari instance DB sumber dan enkripsi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa detail untuk sumber PostgreSQL DB instance.	Di konsol Amazon RDS, pilih instans DB PostgreSQL sumber. Pada tab Konfigura si, pastikan enkripsi tidak diaktifkan untuk instance. Untuk ilustrasi layar, lihat bagian Informasi tambahan.	DBA
Buat snapshot DB.	Buat snapshot DB dari instance yang ingin Anda enkripsi. Jumlah waktu yang diperlukan untuk membuat	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	snapshot tergantung pada ukuran database Anda. Untuk petunjuknya, lihat Membuat snapshot DB di dokumentasi Amazon RDS.	
Enkripsi snapshot.	Di panel navigasi konsol Amazon RDS, pilih Snapshots , dan pilih snapshot DB yang Anda buat. Untuk Tindakan, pilih Salin Snapshot. Berikan Wilayah AWS tujuan dan nama salinan snapshot DB di bidang yang sesuai. Pilih kotak centang Aktifkan Enkripsi. Untuk Master Key, tentukan pengidentifikasi kunci KMS yang akan digunakan untuk mengenkripsi salinan snapshot DB. Pilih Salin Snapshot. Untuk informasi selengkapnya, lihat Menyalin snapshot di dokumentasi Amazon RDS.	DBA

Siapkan instans DB target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan snapshot DB.	Di konsol Amazon RDS, pilih Snapshots. Pilih snapshot terenkripsi yang Anda buat. Untuk Tindakan, pilih Pulihkan Snapshot. Untuk DB Instance Identifier, berikan nama	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	unik untuk instans DB baru. Tinjau detail instans, lalu pilih Restore DB Instance. Instans DB baru yang terenkripsi akan dibuat dari snapshot Anda. Untuk informasi selengkap nya, lihat Memulihkan dari snapshot DB di dokumentasi Amazon RDS.	
Migrasikan data dengan menggunakan AWS DMS.	Di konsol AWS DMS, buat tugas AWS DMS. Untuk Jenis Migrasi, pilih Migrasika n data yang ada dan replikasi perubahan yang sedang berlangsung. Di Pengaturan Tugas, untuk mode persiapan tabel Target, pilih Potong. Untuk informasi selengkapnya, lihat Membuat tugas dalam dokumentasi AWS DMS.	DBA
Aktifkan validasi data.	Di Pengaturan Tugas, pilih Aktifkan validasi. Ini memungkinkan Anda membandingkan data sumber dengan data target untuk memverifikasi bahwa data telah dimigrasi secara akurat.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Nonaktifkan kendala pada instans DB target.	Nonaktifkan pemicu dan batasan kunci asing apa pun pada instans DB target, lalu mulai tugas AWS DMS. Untuk informasi selengkapnya tentang menonaktifkan pemicu dan batasan kunci asing, lihat dokumentasi AWS DMS.	DBA
Verifikasi data.	Setelah beban penuh selesai, verifikasi data pada instans DB target untuk melihat apakah itu cocok dengan data sumber. Untuk informasi selengkapnya, lihat validasi data AWS DMS dalam dokumentasi AWS DMS.	DBA

Potong ke instans DB target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan operasi tulis pada instans DB sumber.	Hentikan operasi tulis pada instans DB sumber sehingga downtime aplikasi dapat dimulai. Verifikasi bahwa AWS DMS telah menyelesaikan replikasi untuk data dalam pipeline. Aktifkan pemicu dan kunci asing pada instans DB target.	DBA
Perbarui urutan basis data	Jika database sumber berisi nomor urut, verifikas	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	i dan perbarui urutan dalam database target.	
Konfigurasikan titik akhir aplikasi.	Konfigurasikan koneksi aplikasi Anda untuk menggunakan titik akhir instans Amazon RDS DB yang baru. Instans DB sekarang dienkripsi.	DBA, Pemilik aplikasi

Sumber daya terkait

- Membuat tugas AWS DMS
- Memantau tugas replikasi menggunakan Amazon CloudWatch
- Memantau tugas AWS DMS
- Memperbarui kunci enkripsi Amazon RDS

Informasi tambahan

Memeriksa enkripsi untuk sumber PostgreSQL DB instance:

Catatan tambahan untuk pola ini:

- Aktifkan replikasi pada PostgreSQL dengan mengatur parameter ke 1. `rds.logical_replication`

Catatan penting: Slot replikasi menyimpan file write ahead log (WAL) sampai file dikonsumsi secara eksternal—misalnya, oleh`pg_recvlogical`; dengan mengekstrak, mengubah, dan memuat (ETL) pekerjaan; atau oleh AWS DMS. Saat Anda menetapkan nilai `rds.logical_replication` parameter ke 1, AWS DMS menetapkan`wal_level`,, `max_wal_sendersmax_replication_slots`, dan `max_connections` parameter. Jika slot replikasi logis hadir tetapi tidak ada konsumen untuk file WAL yang disimpan oleh slot replikasi, Anda mungkin melihat peningkatan penggunaan disk log transaksi dan penurunan konstan dalam ruang

penyimpanan gratis. Untuk informasi lebih lanjut dan langkah-langkah untuk mengatasi masalah ini, lihat artikel Bagaimana saya bisa mengidentifikasi apa yang menyebabkan kesalahan “Tidak ada ruang tersisa di perangkat” atau "DiskFull" di Amazon RDS for PostgreSQL? di Pusat Pengetahuan AWS Support.

- Perubahan skema apa pun yang Anda buat pada instans DB sumber setelah Anda membuat snapshot DB tidak akan ada pada instans DB target.
- Setelah Anda membuat instans DB terenkripsi, Anda tidak dapat mengubah kunci KMS yang digunakan oleh instans DB tersebut. Pastikan untuk menentukan persyaratan kunci KMS Anda sebelum Anda membuat instans DB terenkripsi Anda.
- Anda harus menonaktifkan pemicu dan kunci asing pada instans DB target sebelum menjalankan tugas AWS DMS. Anda dapat mengaktifkan kembali ini ketika tugas selesai.

Menerapkan penandaan otomatis database Amazon RDS saat diluncurkan

Susanne Kangnoh dan Archit Mathur, Amazon Web Services

Ringkasan

Amazon Relational Database Service (Amazon RDS) adalah layanan web yang memudahkan penyiapan, pengoperasian, dan skala database relasional di Amazon Web Services (AWS) Cloud. Layanan ini menyediakan kapasitas yang hemat biaya dan dapat diubah ukurannya untuk basis data relasional standar industri serta mengelola tugas administrasi basis data umum.

Anda dapat menggunakan penandaan untuk mengkategorikan sumber daya AWS Anda dengan berbagai cara. Penandaan basis data relasional berguna ketika Anda memiliki banyak sumber daya di akun Anda dan Anda ingin dengan cepat mengidentifikasi sumber daya tertentu berdasarkan tag. Anda dapat menggunakan tag Amazon RDS untuk menambahkan metadata kustom ke instans RDS DB Anda. Tag terdiri dari kunci dan nilai yang ditentukan pengguna. Kami menyarankan Anda membuat kumpulan tag yang konsisten untuk memenuhi persyaratan organisasi Anda.

Pola ini menyediakan CloudFormation template AWS untuk membantu Anda memantau dan menandai instans RDS DB. Template membuat CloudWatch acara Amazon Events yang mengawasi acara AWS CloudTrail Create DBInstance. (CloudTrail menangkap panggilan API untuk Amazon RDS sebagai acara.) Saat mendeteksi peristiwa ini, ia memanggil fungsi AWS Lambda yang secara otomatis menerapkan kunci tag dan nilai yang Anda tentukan. Template juga mengirimkan pemberitahuan bahwa instance telah diberi tag, dengan menggunakan Amazon Simple Notification Service (Amazon SNS).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Bucket Amazon Simple Storage Service (Amazon S3) untuk mengunggah kode Lambda.
- Alamat email tempat Anda ingin menerima pemberitahuan penandaan.

Batasan

- Solusinya mendukung CloudTrail Buat DBInstance acara. Itu tidak membuat pemberitahuan untuk acara lainnya.

Arsitektur

Arsitektur alur kerja

Otomatisasi dan skala

- Anda dapat menggunakan CloudFormation template AWS beberapa kali untuk Wilayah dan akun AWS yang berbeda. Anda perlu menjalankan template hanya sekali di setiap Wilayah atau akun.

Alat

Layanan AWS

- AWS CloudTrail — AWS CloudTrail adalah layanan AWS yang membantu Anda dalam tata kelola, kepatuhan, serta audit operasional dan risiko akun AWS Anda. Tindakan yang diambil oleh pengguna, peran, atau layanan AWS dicatat sebagai peristiwa di CloudTrail.
- Amazon CloudWatch Events — Amazon CloudWatch Events menghadirkan aliran peristiwa sistem yang mendekati real-time yang menjelaskan perubahan sumber daya AWS. CloudWatch Peristiwa menjadi sadar akan perubahan operasional saat terjadi dan mengambil tindakan korektif seperlunya, dengan mengirim pesan untuk merespons lingkungan, mengaktifkan fungsi, membuat perubahan, dan menangkap informasi negara.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung kode berjalan tanpa perlu menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel yang dapat digunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan danau data.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) adalah layanan web yang memungkinkan aplikasi, pengguna akhir, dan perangkat untuk langsung mengirim dan menerima pemberitahuan dari cloud.

Kode

Pola ini mencakup lampiran dengan dua file:

- `index.zip`adalah file terkompresi yang menyertakan kode Lambda untuk pola ini.
- `rds.yaml`adalah CloudFormation template yang menyebarkan kode Lambda.

Lihat bagian Epics untuk informasi tentang cara menggunakan file-file ini.

Epik

Terapkan kode Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode ke bucket S3.	Buat bucket S3 baru atau gunakan bucket S3 yang sudah ada untuk mengunggah `index.zip` file terlampir (kode Lambda). Bucket ini harus berada di Wilayah AWS yang sama dengan sumber daya (instans RDS DB) yang ingin Anda pantau.	Arsitek awan
Menyebarkan CloudFormation template.	Buka konsol Cloudformation di Wilayah AWS yang sama dengan bucket S3, dan terapkan `rds.yaml` file yang disediakan di lampiran. Dalam epik berikutnya, berikan nilai untuk parameter template.	Arsitek awan

Lengkapi parameter dalam CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan nama bucket S3.	Masukkan nama bucket S3 yang Anda buat atau pilih di epik pertama. Bucket S3	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ini berisi file.zip untuk kode Lambda dan harus berada di Wilayah AWS yang sama dengan CloudFormation template dan instans RDS DB yang ingin Anda pantau.	
Berikan kunci S3.	Berikan lokasi file kode Lambda .zip di bucket S3 Anda, tanpa garis miring di depan (misalnya, atau). `index.zip controls/index.zip`	Arsitek awan
Berikan alamat email.	Berikan alamat email aktif tempat Anda ingin menerima pemberitahuan pelanggaran.	Arsitek awan
Tentukan tingkat logging.	Tentukan tingkat logging dan verbositas. `Info`menunjuk pesan informasi rinci tentang kemajuan aplikasi dan harus digunakan hanya untuk debugging. `Error`menunjuk peristiwa kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. `Warning`menunjuk situasi yang berpotensi berbahaya.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masukkan kunci tag dan nilai untuk instans RDS DB Anda.	Masukkan kunci tag dan nilai yang diperlukan yang ingin Anda terapkan secara otomatis ke instance RDS. Untuk informasi selengkap nya, lihat Menandai resource Amazon RDS dalam dokumentasi AWS.	Arsitek awan

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan langganan email.	Ketika CloudFormation template berhasil digunakan , ia mengirimkan pesan email berlangganan ke alamat email yang Anda berikan. Untuk menerima pemberitahuan saat instans Anda ditandai, Anda harus mengonfirmasi langganan email ini.	Arsitek awan

Sumber daya terkait

- Membuat ember (dokumentasi Amazon S3)
- Menandai sumber daya Amazon RDS (dokumentasi Amazon Aurora)
- Mengunggah objek (dokumentasi Amazon S3)
- Membuat aturan CloudWatch Peristiwa yang memicu panggilan AWS API menggunakan AWS CloudTrail (CloudWatch dokumentasi Amazon)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Perkirakan biaya tabel DynamoDB untuk kapasitas sesuai permintaan

Moinul Al-Mamun, Amazon Web Services

Ringkasan

Amazon DynamoDB adalah database transaksional NoSQL yang menyediakan latensi milidetik satu digit bahkan pada skala petabyte. Penawaran tanpa server Amazon Web Services (AWS) ini semakin populer karena kinerja dan skalabilitasnya yang konsisten. Anda tidak perlu menyediakan infrastruktur yang mendasarinya. Meja tunggal Anda bisa tumbuh hingga petabyte.

Dengan mode kapasitas sesuai permintaan, Anda membayar per permintaan untuk data yang dibaca dan ditulis bahwa aplikasi Anda bekerja di tabel. Biaya AWS didasarkan pada akumulasi unit permintaan baca (RRUs) dan unit permintaan tulis (WRUs) dalam sebulan. DynamoDB memantau ukuran meja Anda terus menerus sepanjang bulan untuk menentukan biaya penyimpanan Anda. Ini mendukung pencadangan berkelanjutan dengan point-in-time-recovery (PITR). DynamoDB memantau ukuran tabel berkemampuan PITR Anda secara terus menerus sepanjang bulan untuk menentukan biaya cadangan Anda.

Untuk memperkirakan biaya DynamoDB untuk sebuah proyek, penting untuk menghitung berapa banyak RRU, WRU, dan penyimpanan yang akan dikonsumsi pada berbagai tahap siklus hidup produk Anda. Untuk estimasi biaya kasar, Anda dapat menggunakan Kalkulator Harga AWS, tetapi Anda harus memberikan perkiraan jumlah RRUs, WRUs, dan persyaratan penyimpanan untuk tabel Anda. Ini bisa sulit untuk diperkirakan pada awal proyek. AWS Pricing Calculator tidak mempertimbangkan tingkat pertumbuhan data atau ukuran item, dan tidak mempertimbangkan jumlah pembacaan dan penulisan untuk tabel dasar dan indeks sekunder global (GSIs) secara terpisah. Untuk menggunakan Kalkulator Harga AWS, Anda harus memperkirakan semua aspek tersebut untuk mengasumsikan angka rata-rata untuk WRU, RRU, dan ukuran penyimpanan untuk mendapatkan estimasi biaya Anda.

Pola ini menyediakan mekanisme dan template Microsoft Excel yang dapat digunakan kembali untuk memperkirakan faktor biaya DynamoDB dasar, seperti biaya tulis, baca, penyimpanan, cadangan dan pemulihan, untuk mode kapasitas sesuai permintaan. Ini lebih terperinci daripada AWS Pricing Calculator, dan mempertimbangkan tabel dasar dan GSIs persyaratan secara independen. Ini juga mempertimbangkan tingkat pertumbuhan data item bulanan dan perkiraan biaya selama tiga tahun.

Prasyarat dan batasan

Prasyarat

- Pengetahuan dasar tentang desain model data DynamoDB dan DynamoDB
- Pengetahuan dasar tentang harga DynamoDB, WRU, RRU, penyimpanan, dan pencadangan dan pemulihan DynamoDB (untuk informasi lebih lanjut, lihat Harga untuk Kapasitas Sesuai Permintaan)
- Pengetahuan tentang data, model data, dan ukuran item Anda di DynamoDB
- Pengetahuan tentang DynamoDB GSIs

Batasan

- Template memberi Anda perkiraan perhitungan, tetapi tidak sesuai untuk semua konfigurasi. Untuk mendapatkan perkiraan yang lebih akurat, Anda harus mengukur ukuran item individual untuk setiap item di tabel dasar dan GSIs.
- Untuk perkiraan yang lebih akurat, Anda harus mempertimbangkan jumlah penulisan yang diharapkan (menyisipkan, memperbarui, dan menghapus) dan membaca untuk setiap item dalam satu bulan rata-rata.
- Pola ini mendukung estimasi hanya biaya tulis, baca, penyimpanan, dan cadangan dan pemulihan untuk beberapa tahun ke depan berdasarkan asumsi pertumbuhan data tetap.

Alat

Layanan AWS

- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.

Alat lainnya

- AWS Pricing Calculator adalah alat perencanaan berbasis web yang dapat Anda gunakan untuk membuat perkiraan kasus penggunaan AWS Anda.

Praktik terbaik

Untuk membantu menjaga biaya tetap rendah, pertimbangkan praktik terbaik desain DynamoDB berikut.

- Desain kunci partisi - Gunakan kunci partisi kardinalitas tinggi untuk mendistribusikan beban secara merata.
- Pola desain daftar kedekatan - Gunakan pola desain ini untuk mengelola one-to-many dan many-to-many hubungan.
- Indeks jarang — Gunakan indeks jarang untuk Anda. GSIs Saat Anda membuat GSI, Anda menentukan kunci partisi dan secara opsional kunci pengurutan. Hanya item di tabel dasar yang berisi kunci partisi GSI yang sesuai yang muncul di indeks jarang. Ini membantu untuk tetap GSIs lebih kecil.
- Index overloading — Gunakan GSI yang sama untuk mengindeks berbagai jenis item.
- GSI menulis sharding — Shard dengan bijak untuk mendistribusikan data di seluruh partisi untuk kueri yang efisien dan lebih cepat.
- Item besar — Simpan hanya metadata di dalam tabel, simpan gumpalan di Amazon S3, dan simpan referensi di DynamoDB. Pecahkan item besar menjadi beberapa item, dan indeks secara efisien dengan menggunakan tombol sortir.

Untuk praktik terbaik desain lainnya, lihat Panduan Pengembang Amazon DynamoDB.

Epik

Ekstrak informasi item dari model data DynamoDB Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan ukuran item.	1. Periksa berapa banyak jenis barang yang akan Anda simpan di meja Anda. 2. Untuk menghitung ukuran setiap item dalam kilobyte, tambahkan ukuran Kunci dan Nilai dari setiap atribut. 3. Hitung ukuran item untuk tabel dasar dan untuk setiap GSI.	Insinyur data
Perkirakan biaya tulis.	Untuk memperkirakan biaya tulis dalam mode kapasitas	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	sesuai permintaan, pertama-tama Anda harus mengukur berapa banyak yang WRUs akan dikonsumsi dalam sebulan. Untuk itu, Anda perlu mempertimbangkan faktor-faktor berikut:
	• Jumlah operasi buat, perbarui, dan hapus untuk setiap item dalam sebulan.
	• Jumlah yang tersedia GSIs. Pertimbangkan setiap indeks secara independen.
	  • Ukuran rata-rata item indeks
	  • Jumlah waktu sinkronisasi pada indeks
	• Berapa banyak hal baru (misalnya, komponen atau produk) yang akan ditambahkan dalam tabel setiap bulan? Jumlah hal yang ditambahkan bisa berbeda setiap bulan, tetapi Anda dapat mengasumsikan tingkat pertumbuhan rata-rata berdasarkan kasus bisnis Anda.
	
	Untuk informasi selengkapnya, lihat bagian Informasi tambahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perkirakan biaya baca.	Untuk memperkirakan biaya baca dalam mode sesuai permintaan, pertama-tama Anda harus mengukur berapa banyak yang RRUs akan dikonsumsi dalam sebulan. Untuk itu, Anda perlu mempertimbangkan faktor-faktor berikut: • Jumlah yang tersedia GSIs. Pertimbangkan setiap indeks secara independen. • Ukuran rata-rata item indeks • Jumlah rata-rata pembacaan per produk per bulan. • Jumlah total barang yang tersedia (komponen atau produk) dalam tabel DynamoDB.	Insinyur data, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perkirakan ukuran dan biaya penyimpanan.	Pertama, perkirakan kebutuhan penyimpanan bulanan rata-rata berdasarkan ukuran item Anda di tabel. Kemudian hitung biaya penyimpanan dengan mengalikan ukuran penyimpanan dengan harga penyimpanan per GB untuk Wilayah AWS Anda. Jika Anda sudah memasukkan data untuk memperkirakan biaya penulisan, Anda tidak perlu memasukkannya lagi untuk menghitung ukuran penyimpanan. Jika tidak, untuk memperkirakan ukuran penyimpanan, Anda perlu mempertimbangkan faktor-faktor berikut: • Jumlah item data dalam modul (produk) berdasarkan desain tabel Anda. • Ukuran item rata-rata dalam kilobyte. • Jumlah yang tersedia GSIs. Pertimbangkan setiap indeks secara independen. • Ukuran rata-rata item indeks • Berapa banyak produk baru yang akan ditambahkan	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dalam tabel setiap bulan? Jumlah produk baru bisa berbeda setiap bulan, tetapi Anda dapat mengasumsikan tingkat pertumbuhan rata-rata berdasarkan kasus bisnis Anda. Contoh ini menggunakan rata-rata 10 juta produk baru setiap bulan.	

Masukkan informasi item dan objek di template Excel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh templat Excel dari bagian Lampiran, dan sesuaikan dengan tabel kasus penggunaan Anda.	1. Unduh templat Excel. 2. Sesuaikan modul bisnis dan GSIs, berdasarkan desain meja Anda.	Insinyur data
Masukkan informasi dalam template Excel.	1. Perbarui informasi item di lembar. Perbarui data hanya dalam sel oranye. 2. Sesuaikan nomor objek: Berapa banyak yang bisa ditambahkan ke dalam tabel setiap bulan? 3. Perbarui harga WRU dan RRU per juta untuk Wilayah AWS Anda. 4. Perbarui harga penyimpanan dan pencadangan per GB-bulan untuk Wilayah AWS Anda.	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Perbarui harga pemulihan per GB untuk Wilayah AWS Anda. Dalam template, ada tiga item, atau entitas: informasi , metadata, dan hubungan. Ada dua GSIs. Untuk kasus penggunaan Anda, jika Anda membutuhkan lebih banyak item, buat baris baru. Jika Anda membutuhkan lebih banyak GSIs, salin blok GSI yang ada, dan tempel untuk membuat blok GSI sebanyak yang Anda butuhkan. Kemudian sesuaikan perhitungan kolom SUM dan TOTAL.	

Sumber daya terkait

Referensi

- Harga Amazon DynamoDB untuk kapasitas sesuai permintaan
- Kalkulator Harga AWS untuk DynamoDB
- Praktik terbaik untuk merancang dan membuat arsitektur dengan DynamoDB
- Mulai menggunakan DynamoDB

Panduan dan pola

- Pemodelan data dengan Amazon DynamoDB
- Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB

Informasi tambahan

Tulis contoh perhitungan biaya

Desain model data DynamoDB menunjukkan tiga item untuk suatu produk, dan ukuran item rata-rata 4 KB. Ketika Anda menambahkan produk baru ke dalam tabel dasar DynamoDB, ia mengkonsumsi jumlah item* (ukuran item/1 KB unit tulis) = 3 * (4/1) = 12 WRU. Dalam contoh ini, untuk menulis 1 KB, produk mengkonsumsi 1 WRU.

Baca contoh perhitungan biaya

Untuk mendapatkan estimasi RRU, pertimbangkan rata-rata berapa kali setiap item akan dibaca dalam sebulan. Misalnya, item Informasi akan dibaca, rata-rata, 10 kali dalam sebulan, dan item metadata akan dibaca dua kali, dan item hubungan akan dibaca lima kali. Dalam contoh template, total RRU untuk semua komponen = jumlah komponen baru yang dibuat setiap bulan* RRU per komponen per bulan = 10 juta * 17 RRU = 170 juta RRU setiap bulan.

Setiap bulan, hal-hal baru (komponen atau produk akan ditambahkan, dan jumlah total produk akan bertambah seiring waktu. Jadi, persyaratan RRU juga akan tumbuh seiring waktu.

- Untuk RRU bulan pertama, konsumsi akan menjadi 170 juta.
- Untuk bulan kedua, konsumsi RRU akan menjadi 2 * 170 juta = 340 juta.
- Untuk bulan ketiga konsumsi RRU akan menjadi 3 * 170 juta = 510 juta.

Grafik berikut menunjukkan konsumsi RRU bulanan dan perkiraan biaya.

Perhatikan bahwa harga dalam grafik hanya untuk ilustrasi. Untuk membuat prakiraan akurat untuk kasus penggunaan Anda, periksa halaman harga AWS, dan gunakan harga tersebut di lembar Excel.

Contoh perhitungan biaya penyimpanan, pencadangan, dan pemulihan

DynamoDB penyimpanan, backup dan restore semuanya terhubung satu sama lain. Backup terhubung langsung dengan penyimpanan, dan pemulihan terhubung langsung dengan ukuran cadangan. Ketika ukuran tabel meningkat, biaya penyimpanan, cadangan, dan pemulihan yang sesuai akan meningkat secara proporsional.

Ukuran dan biaya penyimpanan

Biaya penyimpanan akan meningkat seiring waktu berdasarkan tingkat pertumbuhan data Anda. Misalnya, asumsikan bahwa ukuran rata-rata komponen atau produk dalam tabel dasar dan 11 KB,

dan GSIs 10 juta produk baru akan ditambahkan setiap bulan ke dalam tabel database Anda. Dalam hal ini, ukuran tabel DynamoDB Anda akan bertambah (11 KB* 10 juta) /1024/1024 = 105 GB per bulan. Pada bulan pertama, ukuran penyimpanan meja Anda akan menjadi 105 GB, pada bulan kedua akan menjadi 105+105 = 210 GBs, dan seterusnya.

- Untuk bulan pertama, biaya penyimpanan akan menjadi 105 GB* harga penyimpanan per GB untuk Wilayah AWS Anda.
- Untuk bulan kedua, biaya penyimpanan akan menjadi 210 GB * harga penyimpanan per GB untuk Wilayah Anda.
- Untuk bulan ketiga, biaya penyimpanan akan menjadi 315 GB* harga penyimpanan per GB untuk Wilayah Anda.

Untuk ukuran dan biaya penyimpanan selama tiga tahun ke depan, lihat bagian Ukuran dan perkiraan penyimpanan.

Biaya Backup

Biaya Backup akan meningkat seiring waktu berdasarkan tingkat pertumbuhan data Anda. Saat Anda mengaktifkan pencadangan berkelanjutan dengan point-in-time-recovery (PITR), biaya pencadangan berkelanjutan didasarkan pada penyimpanan rata-rata GB-bulan. Dalam satu bulan kalender, ukuran cadangan rata-rata akan sama dengan ukuran penyimpanan tabel Anda, meskipun ukuran sebenarnya bisa sedikit berbeda. Karena produk baru akan ditambahkan setiap bulan, ukuran penyimpanan total dan ukuran cadangan akan bertambah seiring waktu. Misalnya, untuk bulan pertama, ukuran cadangan rata-rata 105 GB bisa tumbuh menjadi 210 GB untuk bulan kedua.

- Untuk bulan pertama, biaya pencadangan akan menjadi 105 GB-bulan* harga pencadangan berkelanjutan per GB untuk Wilayah AWS Anda.
- Untuk bulan kedua, biaya cadangan akan menjadi 210 GB-bulan* harga cadangan berkelanjutan per GB untuk Wilayah Anda.
- Untuk bulan ketiga, biaya cadangan akan menjadi 315 GB-bulan* harga cadangan berkelanjutan per GB untuk Wilayah Anda.
- dan, seterusnya

Biaya cadangan termasuk dalam grafik di bagian Ukuran penyimpanan dan perkiraan biaya.

Biaya pemulihan

Saat Anda mengambil cadangan terus menerus dengan PITR diaktifkan, biaya operasi pemulihan didasarkan pada ukuran pemulihan. Setiap kali Anda memulihkan, Anda membayar berdasarkan gigabyte data yang dipulihkan. Jika ukuran meja Anda besar dan Anda melakukan pemulihan beberapa kali dalam sebulan, itu akan mahal.

Untuk memperkirakan biaya pemulihan, contoh ini mengasumsikan bahwa Anda melakukan pemulihan PITR satu kali setiap bulan pada akhir bulan. Contoh menggunakan ukuran cadangan rata-rata bulanan sebagai ukuran data pemulihan untuk bulan itu. Untuk bulan pertama, ukuran cadangan rata-rata adalah 105 GB, dan untuk pemulihan pada akhir bulan, ukuran data pemulihan akan menjadi 105 GB. Untuk bulan kedua, itu akan menjadi 210 GBs, dan seterusnya.

Biaya pemulihan akan meningkat seiring waktu berdasarkan tingkat pertumbuhan data Anda.

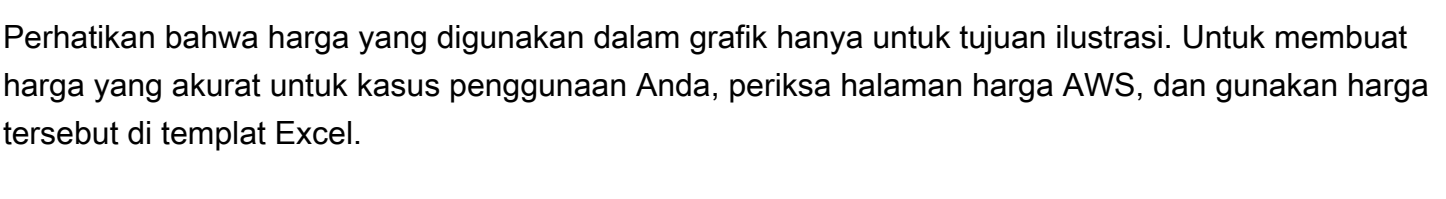
- Untuk bulan pertama, biaya pemulihan akan menjadi 105 GB* harga pemulihan per GB untuk Wilayah AWS Anda.
- Untuk bulan kedua, biaya pemulihan akan menjadi 210 GB * harga pemulihan per GB untuk Wilayah Anda.
- Untuk bulan ketiga, biaya pemulihan akan menjadi 315 GB* harga pemulihan per GB untuk Wilayah Anda.

Untuk informasi selengkapnya, lihat tab Penyimpanan, pencadangan, dan pemulihan di templat Excel dan grafik di bagian berikut.

Ukuran penyimpanan dan perkiraan biaya

Dalam template, ukuran penyimpanan aktual yang dapat ditagih dihitung dengan mengurangi tingkat gratis 25 GB per bulan untuk kelas tabel Standar. Di lembar, Anda akan mendapatkan grafik peramalan yang dipecah menjadi nilai bulanan.

Bagan contoh berikut memperkirakan ukuran penyimpanan bulanan dalam GB, biaya penyimpanan yang dapat ditagih, biaya cadangan sesuai permintaan, dan biaya pemulihan untuk 36 bulan kalender ke depan. Semua biaya dalam USD. Dari grafik, jelas bahwa biaya penyimpanan, cadangan, dan pemulihan meningkat secara proporsional dengan peningkatan ukuran penyimpanan.



Perhatikan bahwa harga yang digunakan dalam grafik hanya untuk tujuan ilustrasi. Untuk membuat harga yang akurat untuk kasus penggunaan Anda, periksa halaman harga AWS, dan gunakan harga tersebut di templat Excel.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB

Moinul Al-Mamun, Amazon Web Services

Ringkasan

Amazon DynamoDB adalah database transaksional NoSQL yang menyediakan latensi milidetik satu digit bahkan pada skala petabyte. Penawaran tanpa server yang populer ini AWS memberikan kinerja dan skalabilitas yang konsisten. Anda tidak perlu menyediakan penyimpanan, dan tabel tunggal Anda dapat tumbuh hingga petabyte.

DynamoDB memantau ukuran meja Anda terus menerus sepanjang bulan untuk menentukan biaya penyimpanan Anda. AWS kemudian menagih Anda untuk ukuran penyimpanan rata-rata dalam gigabyte. Semakin banyak meja Anda tumbuh dari waktu ke waktu, semakin banyak biaya penyimpanan Anda akan tumbuh. Untuk menghitung biaya penyimpanan, Anda dapat menggunakan Kalkulator AWS Harga, tetapi Anda perlu memberikan perkiraan ukuran tabel Anda, termasuk indeks sekunder global (GSIs), yang sangat sulit diperkirakan pada awal proyek. Juga, Kalkulator AWS Harga tidak mempertimbangkan tingkat pertumbuhan data.

Pola ini menyediakan mekanisme dan template Microsoft Excel yang dapat digunakan kembali untuk menghitung ukuran dan biaya penyimpanan DynamoDB. Ini mempertimbangkan persyaratan penyimpanan untuk tabel dasar dan secara GSIs independen. Ini menghitung ukuran penyimpanan dengan mempertimbangkan ukuran item individual Anda dan tingkat pertumbuhan data dari waktu ke waktu.

Untuk mendapatkan perkiraan, masukkan dua informasi ke dalam template:

- Ukuran item individu dalam kilobyte untuk tabel dasar dan GSIs
- Berapa banyak objek atau produk baru yang dapat ditambahkan ke tabel, rata-rata, dalam sebulan, (misalnya, 10 juta)

Template menghasilkan grafik penyimpanan dan perkiraan biaya untuk tiga tahun ke depan, yang ditunjukkan pada contoh berikut.

Prasyarat dan batasan

Prasyarat

- Pengetahuan dasar tentang DynamoDB, termasuk penyimpanan dan harga DynamoDB
- Pengetahuan tentang data, model data, dan ukuran item Anda di DynamoDB
- Pengetahuan tentang indeks sekunder global DynamoDB () GSIs

Batasan

- Template memberi Anda perkiraan perhitungan, tetapi tidak sesuai untuk semua konfigurasi. Untuk mendapatkan perkiraan yang lebih akurat, Anda harus mengukur ukuran item individual untuk setiap item di tabel dasar dan GSIs.
- Pola ini mendukung estimasi hanya ukuran penyimpanan dan biaya untuk beberapa tahun ke depan berdasarkan asumsi pertumbuhan data tetap.

Alat

Layanan AWS

- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.

Alat lainnya

- AWS Kalkulator Harga adalah alat perencanaan berbasis web yang dapat Anda gunakan untuk membuat perkiraan untuk kasus AWS penggunaan Anda.

Epik

Ekstrak informasi item dari model data DynamoDB Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan ukuran item.	1. Tentukan berapa banyak jenis barang yang berbeda yang akan Anda simpan di meja Anda. 2. Untuk menghitung ukuran setiap item dalam kilobyte,	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tambahkan `Key` dan `Value` ukuran setiap atribut. 3. Hitung ukuran item untuk tabel dasar dan untuk setiap GSI.	
Dapatkan jumlah objek yang ditambahkan dalam sebulan.	Perkirakan berapa banyak komponen, atau objek, yang akan ditambahkan ke dalam tabel DynamoDB, rata-rata, dalam satu bulan.	Insinyur data

Masukkan informasi item dan objek di template Excel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh dan sesuaikan spreadsheet Excel.	1. Unduh templat Excel dari dokumen terlampir. 2. Sesuaikan modul bisnis dan GSIs, berdasarkan desain meja Anda.	Insinyur data
Masukkan informasi dalam template Excel.	1. Perbarui informasi item di lembar. 2. Sesuaikan nomor objek: Berapa banyak yang bisa ditambahkan ke dalam tabel setiap bulan? 3. Perbarui harga penyimpan an per GB-bulan untuk Anda. Wilayah AWS	Insinyur data

Sumber daya terkait

- Harga untuk Kapasitas Sesuai Permintaan untuk Amazon DynamoDB
- AWS Kalkulator Harga untuk DynamoDB

Informasi tambahan

Perhatikan bahwa template terlampir hanya memperkirakan ukuran penyimpanan dan biaya untuk kelas tabel penyimpanan Standar. Berdasarkan perkiraan biaya penyimpanan, dan mempertimbangkan ukuran item individu dan tingkat pertumbuhan produk atau objek, Anda dapat memperkirakan hal berikut:

- Biaya ekspor data
- Biaya Backup dan Pemulihan
- Persyaratan penyimpanan data.

Biaya penyimpanan data Amazon DynamoDB

DynamoDB memantau ukuran tabel Anda terus menerus untuk menentukan biaya penyimpanan Anda. DynamoDB mengukur ukuran data yang dapat ditagih dengan menambahkan ukuran byte mentah data Anda ditambah overhead penyimpanan per item yang bergantung pada fitur yang telah Anda aktifkan. Untuk informasi selengkapnya, lihat Panduan Pengembang DynamoDB.

Harga untuk penyimpanan data tergantung pada kelas tabel Anda. 25 GB pertama yang disimpan setiap bulan gratis jika Anda menggunakan kelas tabel Standar DynamoDB. Untuk informasi selengkapnya tentang biaya penyimpanan untuk kelas tabel Akses Standar dan Jarang Jarang Standar Wilayah AWS, lihat Harga untuk Kapasitas Sesuai Permintaan.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Perkirakan ukuran mesin Amazon RDS untuk database Oracle dengan menggunakan laporan AWR

Abhishek Verma dan Eduardo Valentim, Amazon Web Services

Ringkasan

Saat Anda memigrasikan database Oracle ke Amazon Relational Database Service (Amazon RDS) atau Amazon Aurora, menghitung CPU, memori, dan disk I/O untuk database target adalah persyaratan utama. Anda dapat memperkirakan kapasitas database target yang diperlukan dengan menganalisis laporan Oracle Automatic Workload Repository (AWR). Pola ini menjelaskan cara menggunakan laporan AWR untuk memperkirakan nilai-nilai ini.

Basis data Oracle sumber dapat berada di lokasi atau di-host di instans Amazon Elastic Compute Cloud EC2 (Amazon), atau bisa juga berupa Amazon RDS for Oracle DB instans. Basis data target bisa berupa database Amazon RDS atau Aurora.

> **Note**
>
> Perkiraan kapasitas akan lebih tepat jika mesin database target Anda adalah Oracle. Untuk database Amazon RDS lainnya, ukuran mesin dapat bervariasi karena perbedaan arsitektur database.

Kami menyarankan Anda menjalankan uji kinerja sebelum memigrasikan database Oracle Anda.

Prasyarat dan batasan

Prasyarat

- Lisensi Oracle Database Enterprise Edition dan lisensi Oracle Diagnostics Pack untuk mengunduh laporan AWR.

Versi produk

- Semua edisi Oracle Database untuk versi 11g (versi 11.2.0.3.v1 dan yang lebih baru) dan hingga 12.2, dan 18c, 19c.
- Pola ini tidak mencakup Oracle Engineered Systems atau Oracle Cloud Infrastructure (OCI).

Arsitektur

Tumpukan teknologi sumber

Salah satu dari yang berikut:

- Database Oracle lokal
- Database Oracle pada sebuah instance EC2
- Instans Amazon RDS for Oracle DB

Tumpukan teknologi target

- Basis data Amazon RDS atau Amazon Aurora apa pun

Arsitektur target

Untuk informasi tentang proses migrasi lengkap, lihat pola Memigrasikan database Oracle ke Aurora PostgreSQL menggunakan AWS DMS dan AWS SCT.

Otomatisasi dan skala

Jika Anda memiliki beberapa database Oracle untuk dimigrasi dan Anda ingin menggunakan metrik kinerja tambahan, Anda dapat mengotomatiskan proses dengan mengikuti langkah-langkah yang dijelaskan dalam posting blog Instans Amazon RDS ukuran kanan pada skala berdasarkan metrik kinerja Oracle.

Alat

- Oracle Automatic Workload Repository (AWR) adalah repositori yang dibangun ke dalam database Oracle. Ini secara berkala mengumpulkan dan menyimpan aktivitas sistem dan data beban kerja, yang kemudian dianalisis oleh Automatic Database Diagnostic Monitor (ADDM). AWR mengambil snapshot data kinerja sistem secara berkala (secara default, setiap 60 menit) dan menyimpan informasi (secara default, hingga 8 hari). Anda dapat menggunakan tampilan dan laporan AWR untuk menganalisis data ini.

Praktik terbaik

- Untuk menghitung persyaratan sumber daya untuk database target, Anda dapat menggunakan satu laporan AWR, beberapa laporan AWR, atau tampilan AWR dinamis. Kami menyarankan

Anda menggunakan beberapa laporan AWR selama periode beban puncak untuk memperkirakan sumber daya yang diperlukan untuk menangani beban puncak tersebut. Selain itu, tampilan dinamis memberikan lebih banyak titik data yang membantu Anda menghitung kebutuhan sumber daya dengan lebih tepat.

- Anda harus memperkirakan IOPS hanya untuk database yang Anda rencanakan untuk dimigrasi, bukan untuk database dan proses lain yang menggunakan disk.
- Untuk menghitung berapa banyak yang I/O digunakan oleh database, jangan gunakan informasi di bagian Load Profile pada laporan AWR. Gunakan bagian I/O Profil sebagai gantinya, jika tersedia, atau lewati ke bagian Statistik Aktivitas Instance dan lihat nilai total untuk operasi baca dan tulis fisik.
- Saat Anda memperkirakan pemanfaatan CPU, sebaiknya gunakan metode metrik basis data alih-alih statistik sistem operasi (OS), karena didasarkan pada CPU yang hanya digunakan oleh database. (Statistik OS juga mencakup penggunaan CPU oleh proses lain.) Anda juga harus memeriksa rekomendasi terkait CPU dalam laporan ADDM untuk meningkatkan kinerja setelah migrasi.
- Pertimbangkan batas I/O throughput–Amazon Elastic Block Store (Amazon EBS) throughput dan throughput jaringan–untuk ukuran instans tertentu saat Anda menentukan jenis instans yang tepat.
- Jalankan uji kinerja sebelum migrasi untuk memvalidasi ukuran mesin.

Epik

Buat laporan AWR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan laporan AWR.	Untuk mengaktifkan laporan, ikuti instruksi dalam dokumentasi Oracle.	DBA
Periksa periode retensi.	Untuk memeriksa periode retensi laporan AWR, gunakan kueri berikut. `SQL> SELECT snap_interval,retention FROM dba_hist_wr_control;`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan snapshot.	Jika interval snapshot AWR tidak cukup terperinci untuk menangkap lonjakan beban kerja puncak, Anda dapat membuat laporan AWR secara manual. Untuk menghasil kan snapshot AWR manual, gunakan kueri berikut. `SQL> EXEC dbms_work load_repository.cr eate_snapshot;`	DBA
Periksa snapshot terbaru.	Untuk memeriksa snapshot AWR terbaru, gunakan kueri berikut. `SQL> SELECT snap_id, to_char(begin_inte rval_time,'dd/MON/ yy hh24:mi') Begin_Int erval, to_char(end_interv al_time,'dd/MON/yy hh24:mi') End_Interval FROM dba_hist_snapshot ORDER BY 1;`	DBA

Perkirakan I/O persyaratan disk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih metode.	IOPS adalah ukuran standar operasi input dan output per detik pada perangkat	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	penyimpanan, dan mencakup operasi baca dan tulis. Jika Anda memigrasikan database lokal ke AWS, Anda perlu menentukan disk puncak yang I/O digunakan oleh database. Anda dapat menggunakan metode berikut untuk memperkirakan disk I/O untuk database target Anda: • Muat bagian Profil dari laporan AWR • Bagian Statistik Aktivitas Instance dari laporan AWR (gunakan bagian ini untuk Oracle Database 12c atau yang lebih baru) • Bagian Profil I/O dari laporan AWR (gunakan bagian ini untuk versi Oracle Database sebelum 12c) • Tampilan AWR Langkah-langkah berikut menjelaskan keempat metode ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 1: Gunakan profil beban.	Tabel berikut menunjukkan contoh bagian Load Profile dari laporan AWR. **Important** Untuk informasi yang lebih akurat, kami sarankan Anda menggunakan opsi 2 (profil I/O) atau opsi 3 (statistik aktivitas instance) alih-alih profil pemuatan. Per Detik / Per Trans / Per Exec / Per Pangg Wak DB: 26.6 0,2 0.00 0,02 CPU DB: 18.0 0,1 0.00 0,01 Lata Bela CPU 0,2 0.0 0.00 0.00 Ulan ukur (byte 2,45 .9 17,0	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Baca 3,37 23,4 logis ,5 (blok Peru 21,6 150, blok: Baca 13,5 94.4 fisik (blok Tulis 3,46 24.1 fisik (blok Baca 3,58 24.9 perm n IO: Tulis 574, 4.0 perm n IO: Baca 106. 0,7 IO (MB) Tulis 27.1 0,2 IO (MB) Baris 0.0 0.0 pemi n IM:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Sesi Logis Baca IM: Pang 1,245 8.7 peng Pars 4,62(32.2 (SQL Pars 8.9 0,1 keras (SQL Area 824.9 5.7 Kerja SQL (MB) Logc 1.7 0.0 Meng 136,(950,4 usi (SQL Rollb 22.9 0,2 Tran 143,8 : Berdasarkan informasi ini, Anda dapat menghitung IOPs dan throughput sebagai berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	IOPS = Baca I/O permintaan: + I/O Permintaan tulis = 3,586,8 + 574,7 = 4134,5 Throughput = Bacaan fisik (blok) +Tulis fisik (blok) = 13,575,1 + 3,467,3 = 17,042,4 Karena ukuran blok di Oracle adalah 8 KB, Anda dapat menghitung total throughput sebagai berikut: Total throughput dalam MB adalah 17042.4 * 8 * 1024/1024/1024 = 133,2 MB **Warning** Jangan gunakan profil pemuatan untuk memperkirakan ukuran instans. Ini tidak setepat statistik aktivitas instance atau I/O profil.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 2: Gunakan statistik aktivitas instance.	Jika Anda menggunakan versi Oracle Database sebelum 12c, Anda dapat menggunakan bagian Statistik Aktivitas Instance dari laporan AWR untuk memperkirakan IOPS dan throughput. Tabel berikut menunjukkan contoh bagian ini. Statist / Total / per detik / per Trans permir n IO total baca fisik / 2,547, ,217 / 3,610. / 25.11 total byte baca fisik / 80,776 6,124, / 114,48 26,26 / 796,149 8 tulis fisik total permir n IO / 534,19 08 / 757.1 / 5.27 fisik menul total byte / 25.517 8,849, / 36,165 1.84 / 251,508 8	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Berdasarkan informasi ini, Anda dapat menghitung total IOPS dan throughput sebagai berikut: Total IOPS = 3,610.28 + 757.11 = 4367 Total Mbps = 114,482,426.26 + 36,165,631.84 = 150648058 .1/1024/1024 = 143 Mbps	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 3: Gunakan I/O profil.	Dalam Oracle Database 12c, laporan AWR mencakup bagian I/O Profil yang menyajikan semua informasi dalam satu tabel dan memberikan data yang lebih akurat tentang kinerja database. Tabel berikut menunjukkan contoh bagian ini. Baca +Tulis Per Detik / Baca per Detik / Menulis Per Detik Jumlah Permin: 4,367, / 3,610. / 757.1 Permin Database: 4,161, / 3,586. / 574,7 Permin yang Dioptimalkan: 0.0 / 0.0 / 0.0 Permin Ulangi: 179.3 / 2.8 / 176.6	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jumlal 143,7 109.2 34,5 (MB): Datab 133.1 106.1 27.1 (MB): Total 0.0 0.0 0.0 yang Dioptir kan (MB): Ulangi 7.6 2.7 4.9 (MB): Datab 17,042 13,575 3,467,3 (blok): Melalu 5,898, 5,360, 537,6 Buffer Cache (blok): Langs 11,143 8,214, 2,929,7 (blok): Tabel ini memberikan nilai-nilai berikut untuk throughput dan total IOPS: Throughput = 143 MBPS (dari baris kelima, berlabel Total, kolom kedua)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	IOPS = 4.367.4 (dari baris pertama, berlabel Total Permintaan, kolom kedua)	
Opsi 4: Gunakan tampilan AWR.	Anda dapat melihat IOPS dan informasi throughput yang sama dengan menggunak an tampilan AWR. Untuk mendapatkan informasi ini, gunakan kueri berikut: `break on report compute sum of Value on report select METRIC_NAME,avg(AVERAGE) as "Value" from dba_hist_sysmetric_summary where METRIC_NAME in ('Physical Read Total IO Requests Per Sec','Physical Write Total IO Requests Per Sec') group by metric_name;`	DBA

Perkirakan persyaratan CPU

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih metode.	Anda dapat memperkirakan CPU yang diperlukan untuk database target dengan tiga cara:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Dengan menggunakan inti prosesor yang tersedia sebenarnya • Dengan menggunakan core yang digunakan berdasarkan statistik OS • Dengan menggunakan core yang digunakan berdasarkan statistik database Jika Anda melihat inti yang digunakan, kami sarankan Anda menggunakan metode metrik database alih-alih statistik OS, karena didasarkan pada CPU yang hanya digunakan oleh database yang Anda rencanakan untuk dimigrasikan. (Statistik OS juga mencakup penggunaan CPU oleh proses lain.) Anda juga harus memeriksa rekomendasi terkait CPU dalam laporan ADDM untuk meningkatkan kinerja setelah migrasi. Anda juga dapat memperkirakan persyaratan berdasarkan generasi CPU. Jika Anda menggunakan generasi CPU yang berbeda, Anda dapat memperkirakan CPU yang diperlukan dari database	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	target dengan mengikuti instruksi di whitepaper Mengungkap Jumlah v CPUs untuk Kinerja Beban Kerja Optimal.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 1: Perkirakan persyaratan berdasarkan inti yang tersedia.	Dalam laporan AWR:
	• CPUs mengacu pada logis dan virtual CPUs.
	• Core adalah jumlah prosesor dalam chipset CPU fisik.
	• Soket adalah perangkat fisik yang menghubungkan chip ke papan. Prosesor multi-core memiliki soket dengan beberapa core CPU.
	
	Anda dapat memperkirakan core yang tersedia dengan dua cara:
	• Dengan menggunakan perintah OS
	• Dengan menggunakan laporan AWR
	
	Untuk memperkirakan core yang tersedia dengan menggunakan perintah OS
	
	Gunakan perintah berikut untuk menghitung inti dalam prosesor.
	
	`$ cat /proc/cpuinfo | grep "cpu cores"|uniq`
	`cpu cores : 4`
	`cat /proc/cpuinfo |`
	`egrep "core id|physic`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`al id" | tr -d "\n" | sed s/physical/\\nphysical/g | grep -v ^$ | sort | uniq | wc -l`
	
	Gunakan perintah berikut untuk menghitung soket di prosesor.
	
	`grep "physical id" /proc/cpuinfo | sort -u`
	`physical id : 0`
	`physical id : 1`
	
	Note
	Kami tidak menyarankan menggunakan perintah OS seperti nmon dan sar untuk mengekstrak pemanfaatan CPU. Ini karena perhitungan tersebut mencakup pemanfaatan CPU oleh proses lain dan mungkin tidak mencerminkan CPU aktual yang digunakan oleh database.
	
	Untuk memperkirakan core yang tersedia dengan menggunakan laporan AWR	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda juga dapat memperoleh pemanfaatan CPU dari bagian pertama laporan AWR. Berikut kutipan dari laporan tersebut. N Id Inst Inst Wal Rilis RA D DB num Sta X <DB XX> 1 05- 12.1 TI Sep 0 23:(Nar Plat CPU Inti Sok Mem Hos (GB) <ho Linu 80 80 2 441.7 e> x86 64- bit Dalam contoh ini, CPUs hitungannya adalah 80, yang menunjukkan bahwa ini logis (virtual) CPUs. Anda juga dapat melihat bahwa konfigurasi ini memiliki dua soket, satu prosesor fisik pada setiap soket (untuk total dua prosesor fisik), dan 40 core untuk setiap prosesor fisik atau soket.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 2: Perkirakan pemanfaatan CPU dengan menggunakan statistik OS.	Anda dapat memeriksa statistik penggunaan CPU OS baik secara langsung di OS (menggunakan sar atau utilitas OS host lain) atau dengan meninjau nilai IDLE/(IDLE+BUSY) dari bagian Statistik Sistem Operasi dari laporan AWR. Anda dapat melihat detik-detik CPU yang dikonsumsi langsung dari v $osstat. Laporan AWR dan Statspack juga menunjukkan data ini di bagian Statistik Sistem Operasi. Jika ada beberapa database pada kotak yang sama, semuanya memiliki nilai v $osstat yang sama untuk BUSY_TIME. Statistik / Nilai / Nilai Akhir FREE_MEMORY_BYT / 6,810,67,248 / 12,280,799,232 INACTIVE_MEMORYTES / 175,627,33,632 / 160,380,653,568 SWAP_F_BYTES / 17,145,64,336 / 17,145,872,384	DBA

Tugas	Deskripsi		Keterampilan yang dibutuhkan
	SIBUK_V TU	1,305,56 ,937	
	IDLE_TI	4,312,71 ,839	
	IOWAIT_ KTU	53,417,1 4	
	BAGUS_ TU	29,815	
	SYS_WA	148,567, 70	
	USER_T	1,146,91 .783	
	BEBAN	25	29
	VM_IN_E ES	593.920	
	VM_OUT TES	327.680	
	PHYSIC/ MEMOR TES	474,362, 17.152	
	NUM_CF	80	
	NUM_CF ORES	80	
	NUM_CF OCKETS	2	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	GLOBAL_SEND_SOCKET_BUFFER_SIZE_MAX 4,194,304 GLOBAL_RECEIVE_SOCKET_BUFFER_SIZE_MAX 2,097,152 TCP_RECEIVE_SIZE_DEFAULT 87,380 TCP_RECEIVE_SIZE_MAX 6,291,456 TCP_RECEIVE_SIZE_MIN 4,096 TCP_SEND_SIZE_DEFAULT 16,384 TCP_SEND_SIZE_MAX 4,194,304 TCP_SEND_SIZE_MIN 4,096 Jika tidak ada konsumen CPU utama lainnya dalam sistem, gunakan rumus berikut untuk menghitung persentase pemanfaatan CPU:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pemanfaatan = Waktu sibuk/ Total waktu Waktu sibuk = persyaratan = v $osstat.Busy_time C = Total waktu (Sibuk+Me nganggur) C = kapasitas = v$ostat.B usy_time + v$ostat.idle_time Pemanfaatan = BUSY_TIME/ (BUSY_TIME + IDLE_TIME) = -1,305,569,937/(1,305,569,9 37 + 4,312,718,839) = 23% digunakan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 3: Perkirakan pemanfaatan CPU dengan menggunakan metrik database.	Jika beberapa database berjalan di sistem, Anda dapat menggunakan metrik database yang muncul di awal laporan. Snap Id / Wakt Jepre / Sesi / Kursor Se si Mula Snap 1846 28-Sep-09:0(1226 35.8 Akhi Snap 1854 06-Okt-2 13:0(1876 41.1 Berla 11.7! (mer Wak DB: 312.(0 (mer Untuk mendapatkan metrik pemanfaatan CPU, gunakan rumus ini: Penggunaan CPU basis data (% dari daya CPU yang tersedia) = Waktu CPU/ NUM_CPUS/waktu berlalu	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di mana penggunaan CPU dijelaskan oleh waktu CPU dan mewakili waktu yang dihabiskan untuk CPU, bukan waktu menunggu CPU. Perhitungan ini menghasilkan: = 312.625.40/ 11,759.64/80 = 33% CPU sedang digunakan Jumlah core (33%) * 80 = 26,4 core Total inti = 26,4 * (120%) = 31,68 inti Anda dapat menggunakan yang lebih besar dari dua nilai ini untuk menghitung pemanfaatan CPU dari instans Amazon RDS atau Aurora DB. **Note** Di IBM AIX, pemanfaatan yang dihitung tidak sesuai dengan nilai dari sistem operasi atau database. Nilai-nilai ini memang cocok pada sistem operasi lain.	

Perkirakan kebutuhan memori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perkirakan kebutuhan memori dengan menggunakan statistik memori.	Anda dapat menggunak an laporan AWR untuk menghitung memori database sumber dan mencocokk annya di database target. Anda juga harus memeriksa kinerja database yang ada dan mengurangi kebutuhan memori Anda untuk menghemat biaya, atau meningkatkan kebutuhan Anda untuk meningkatkan kinerja. Itu membutuhkan analisis rinci tentang waktu respons AWR dan perjanjian tingkat layanan (SLA) aplikasi. Gunakan jumlah penggunaa n Oracle system global area (SGA) dan program global area (PGA) sebagai estimasi pemanfaatan memori untuk Oracle. Tambahkan tambahan 20 persen untuk OS untuk menentukan persyaratan ukuran memori target. Untuk Oracle RAC, gunakan jumlah estimasi pemanfaatan memori pada semua node RAC dan kurangi total memori, karena disimpan di blok umum. 1. Periksa metrik dalam tabel Persentase Efisiensi	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Instans. Tabel menggunakan istilah-istilah berikut: • Buffer Hit% adalah persentase kali blok tertentu ditemukan di cache buffer alih-alih melakukan I/O fisik. Untuk kinerja yang lebih baik, targetkan 100 persen. • Buffer Nowait% harus mendekati 100 persen. • Latch Hit% harus mendekati 100 persen. • % Non-Parse CPU adalah persentase waktu CPU yang dihabiskan dalam aktivitas non-parsing. Nilai ini harus mendekati 100 persen.. Persentase Efisiensi Instance (target 100%) Penya Nowa %: 99,99 Ulang NoWa %: 100.00 Penya Hit %: 99,84 Uruta % dalam memo 100.00	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pusta Hit %: 748.7 Soft Parse %: 99,81 Jalan ke Parse %: 96,61 Kait Hit %: 100.00 Parse CPU untuk Parse Elaps %: 72,73 % CPU Non-Parse :: 99,21 Flash Cach Hit %: 0.00 Dalam contoh ini, semua metrik terlihat bagus, sehingga Anda dapat menggunakan SGA dan PGA untuk database yang ada sebagai persyaratan perencanaan kapasitas. 2. Periksa bagian statistik memori dan hitung SGA/ PGA. Mulailah Akhiri	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tuan Rumah Mem (MB): 452,387 452,387.3 Penggun SGA (MB): 220,544 220,544,0 Penggun PGA (MB): 36,874,9 45,270,0 Total memori instance yang digunakan = SGA+PGA = 220 GB+45 GB = 265 GB Tambahkan 20 persen buffer: Total memori instance = 1,2 * 265 GB = 318 GB Karena SGA dan PGA menyumbang 70 persen dari memori host, kebutuhan memori total adalah: Total memori host = 318/0,7 = 464 GB **Note** Saat Anda bermigrasi ke Amazon RDS for	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Oracle, PGA dan SGA dihitung sebelumny a berdasarkan rumus yang telah ditentukan sebelumnya. Pastikan nilai yang telah dihitung sebelumnya mendekati perkiraan Anda.	

Tentukan jenis instans DB dari database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan jenis instans DB berdasarkan disk I/O, CPU, dan estimasi memori.	Berdasarkan perkiraan pada langkah sebelumnya, kapasitas target Amazon RDS atau database Aurora harus: • 68 core dari CPU • 143 MBPS throughput • 4367 IOPS untuk disk I/O • Memori 464 GB Dalam database Amazon RDS atau Aurora target, Anda dapat memetakan nilai-nilai ini ke tipe instans db.r5.16x large, yang memiliki kapasitas 32 core, 512 GB RAM, dan 13.600 Mbps throughput. Untuk informasi selengkapnya, lihat posting blog AWS Instans	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon RDS ukuran kanan pada skala berdasarkan metrik kinerja Oracle.	

Sumber daya terkait

- Kelas instans Aurora DB (dokumentasi Amazon Aurora)
- Penyimpanan instans Amazon RDS DB (dokumentasi Amazon RDS)
- Alat AWS Miner (GitHub repositori)

Ekspor Amazon RDS untuk tabel SQL Server ke bucket S3 dengan menggunakan AWS DMS

Subhani Shaik, Amazon Web Services

Ringkasan

Amazon Relational Database Service (Amazon RDS) untuk SQL Server tidak mendukung pemuatan data ke server terkait mesin DB lainnya di Amazon Web Services (AWS) Cloud. Sebagai gantinya, Anda dapat menggunakan AWS Database Migration Service (AWS DMS) untuk mengekspor tabel Amazon RDS for SQL Server ke bucket Amazon Simple Storage Service (Amazon S3), tempat data tersedia untuk engine DB lainnya.

AWS DMS membantu Anda memigrasikan database ke AWS dengan cepat dan aman. Database sumber tetap beroperasi penuh selama migrasi, meminimalkan waktu henti ke aplikasi yang bergantung pada database. AWS DMS dapat memigrasikan data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan.

Pola ini menggunakan AWS Secrets Manager saat mengonfigurasi titik akhir AWS DMS. Secrets Manager membantu Anda melindungi rahasia yang diperlukan untuk mengakses aplikasi, layanan, dan sumber daya TI Anda. Anda dapat menggunakan layanan untuk memutar, mengelola, dan mengambil kredenal database, kunci API, dan rahasia lainnya sepanjang siklus hidupnya. Pengguna dan aplikasi mengambil rahasia dengan panggilan ke Secrets Manager, mengurangi kebutuhan untuk hardcode informasi sensitif. Secrets Manager menawarkan rotasi rahasia dengan integrasi bawaan untuk Amazon RDS, Amazon Redshift, dan Amazon DocumentDB. Selain itu, layanan ini dapat diperluas ke jenis rahasia lainnya, termasuk kunci API dan OAuth token. Dengan Secrets Manager, Anda dapat mengontrol akses ke rahasia dengan menggunakan izin halus dan mengaudit rotasi rahasia secara terpusat untuk sumber daya di AWS Cloud, layanan pihak ketiga, dan di tempat.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Ember S3
- Awan pribadi virtual (VPC)
- Sebuah subnet DB

- Amazon RDS for SQL Server
- Peran AWS Identity and Access Management (IAM) dengan akses (daftar, dapatkan, dan letakkan objek) ke bucket S3 atas nama instans Amazon RDS.
- Secrets Manager untuk menyimpan kredensil instans RDS.

Arsitektur

Tumpukan teknologi

- Amazon RDS for SQL Server
- AWS DMS
- Amazon S3
- AWS Secrets Manager

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk mengimpor data dari instans Amazon RDS ke bucket S3 dengan bantuan AWS DMS.

1. Tugas migrasi AWS DMS menghubungkan ke instans Amazon RDS sumber melalui titik akhir sumber
2. Menyalin data dari sumber instans Amazon RDS
3. Tugas migrasi AWS DMS terhubung ke bucket S3 target melalui titik akhir target
4. Mengekspor data yang disalin ke bucket S3 dalam format nilai yang dipisahkan koma (CSV)

Alat

Layanan AWS

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.

Layanan lainnya

- Microsoft SQL Server Management Studio (SSMS) adalah alat untuk mengelola SQL Server, termasuk mengakses, mengkonfigurasi, dan mengelola komponen SQL Server.

Epik

Konfigurasikan instans Amazon RDS for SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh Amazon RDS for SQL Server.	1. Buka AWS Management Console, pilih RDS, dan gunakan opsi Standard create untuk membuat instans Amazon RDS dengan edisi yang diperlukan, seperti SQL Server Express Edition, SQL Server Standard Edition, atau SQL Server Enterprise Edition. Untuk versi, pilih 2016 atau yang lebih baru. 2. Di bawah Template, pilih Dev/Test.	DBA, DevOps insinyur
Siapkan kredensil untuk instance.	1. Masukkan nama untuk contoh.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Berikan nama pengguna dan kata sandi untuk instans Amazon RDS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kelas instance, penyimpanan, penskalaan otomatis, dan ketersediaan.	1. Pilih kelas instans DB dari daftar: Kelas Standard, Memory Optimized, dan Burstable. Pilih jenis instans DB yang mengalokasikan kapasitas komputasi, jaringan, dan memori yang dibutuhkan oleh beban kerja yang direncanakan untuk instans DB ini. Untuk informasi selengkapnya, lihat dokumentasi AWS. 2. Pilih Jenis Penyimpanan dari daftar: General Purpose SSD, Provisioned IOPS SSD, atau Magnetic. Alokasikan ukuran penyimpanan default sesuai kebutuhan. 3. Pilih Aktifkan penskalaan otomatis penyimpanan untuk meningkatkan penyimpanan Amazon RDS berdasarkan perencanaan kapasitas Anda. 4. Penerapan multi-AZ dengan instans replikasi didukung oleh AWS DMS. Jika terjadi pemadaman di Availability Zone, perangkat keras internal, atau jaringan, AWS DMS akan membuat instance siaga dan menyediakan	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ketersediaan tinggi (HA) melalui failover otomatis ke replika siaga. Tergantung pada ukuran impor Anda, pilih opsi yang sesuai.	
Tentukan VPC, grup subnet, akses publik, dan grup keamanan.	Pilih grup keamanan VPC, subnet DB, dan VPC sesuai kebutuhan untuk membuat instans Amazon RDS. Ikuti praktik terbaik, misalnya: • Jangan aktifkan akses publik ke instans RDS DB. • Jangan gunakan CIDR 0.0.0.0/0 di grup keamanan. • Gunakan hanya alamat IP dan detail port yang diperlukan untuk mengakses instans RDS.	DBA, DevOps insinyur
Konfigurasikan pemantauan, pencadangan, dan pemeliharaan.	1. Tentukan opsi cadangan yang Anda inginkan. Secara default, pencadangan otomatis diaktifkan dengan periode retensi 7 hari. 2. Pilih pengaturan jendela pemutakhiran dan Pemeliharaan versi minor otomatis yang sesuai untuk menerapkan modifikasi atau pemeliharaan yang tertunda ke database oleh Amazon RDS. 3. Pilih Buat basis data.	DBA, DevOps insinyur

Mengatur database dan contoh data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel dan muat contoh data.	Dalam database baru, buat tabel. Gunakan kode contoh di bagian Informasi tambahan untuk memuat data ke dalam tabel.	DBA, DevOps insinyur

Siapkan kredensial

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat secret.	1. Di konsol, pilih Secrets Manager, dan pilih Simpan rahasia baru. 2. Masukkan nama pengguna dan kata sandi untuk database Amazon RDS for SQL Server. Rahasia ini akan digunakan untuk titik akhir sumber AWS DMS.	DBA, DevOps insinyur

Siapkan akses antara database dan bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk akses ke Amazon RDS.	1. Di konsol, pilih IAM, dan buat peran IAM yang memberikan akses bucket S3 read/write ke Amazon RDS.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di bawah Fitur, pilih Integrasi S3.	

Buat ember S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Untuk menyimpan data dari Amazon RDS for SQL Server, di konsol, pilih S3, lalu pilih Buat bucket. Pastikan bucket S3 tidak tersedia untuk umum.	DBA, DevOps insinyur

Siapkan akses antara AWS DMS dan bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk AWS DMS untuk mengakses Amazon S3.	Buat peran IAM yang memungkinkan AWS DMS membuat daftar, mendapatk an, dan meletakkan objek dari bucket S3.	DBA, DevOps insinyur

Konfigurasikan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir sumber AWS DMS.	1. Di konsol, pilih Database Migration Service, dan pilih Endpoints. Buat titik akhir Sumber, pilih kotak centang Select RDS DB instance.	DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Untuk mesin Sumber, pilih Microsoft SQL Server. 3. Di bawah Access to endpoint database, pilih AWS Secrets Manager, dan masukkan peran rahasia dan IAM yang Anda buat sebelumnya, dan nama database. 4. Uji titik akhir sumber.	
Buat titik akhir target AWS DMS.	Buat titik akhir Target, pilih Amazon S3 sebagai mesin Target. Berikan nama bucket S3 dan nama folder untuk peran IAM yang Anda buat sebelumnya.	DBA, DevOps insinyur
Buat instance replikasi AWS DMS.	Di VPC, subnet, dan grup keamanan yang sama, buat instance replikasi AWS DMS. Untuk informasi selengkapnya tentang memilih kelas instans, lihat dokumentasi AWS.	DBA, DevOps insinyur
Buat tugas migrasi AWS DMS.	Untuk mengekspor data dari Amazon RDS for SQL Server ke bucket S3, buat tugas migrasi database. Untuk jenis migrasi, pilih Migrasi data yang ada. Pilih titik akhir AWS DMS dan instance replikasi yang Anda buat.	DBA, DevOps insinyur

Ekspor data ke bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan tugas migrasi database.	Untuk mengekspor data tabel SQL Server, mulai tugas migrasi database. Tugas akan mengekspor data dari Amazon RDS for SQL Server ke bucket S3 dalam format CSV.	DBA, DevOps insinyur

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya.	Untuk menghindari biaya tambahan, gunakan konsol untuk menghapus sumber daya dalam urutan berikut: 1. Tugas migrasi 2. Instans replikasi 3. Titik akhir 4. Bucket S3 5. Contoh basis data	DBA, DevOps insinyur

Sumber daya terkait

- AWS DMS
- Amazon S3
- Amazon RDS for SQL Server
- Integrasi Amazon S3

Informasi tambahan

Untuk membuat database dan tabel, dan untuk memuat contoh data, gunakan kode berikut.

```
--Step1: Database creation in RDS SQL Server
CREATE DATABASE [Test_DB]
 ON  PRIMARY
( NAME = N'Test_DB', FILENAME = N'D:\rdsdbdata\DATA\Test_DB.mdf' , SIZE = 5120KB ,
 FILEGROWTH = 10%)
 LOG ON
( NAME = N'Test_DB_log', FILENAME = N'D:\rdsdbdata\DATA\Test_DB_log.ldf' , SIZE =
 1024KB , FILEGROWTH = 10%)
GO

--Step2: Create Table
USE Test_DB
GO
Create Table Test_Table(ID int, Company Varchar(30), Location Varchar(20))

--Step3: Load sample data.
USE Test_DB
GO
Insert into Test_Table values(1,'AnyCompany','India')
Insert into Test_Table values(2,'AnyCompany','USA')
Insert into Test_Table values(3,'AnyCompany','UK')
Insert into Test_Table values(4,'AnyCompany','Hyderabad')
Insert into Test_Table values(5,'AnyCompany','Banglore')
```

Menangani blok anonim dalam pernyataan SQL Dinamis di Aurora PostgreSQL

Situs web, Amazon Web Services

Ringkasan

Catatan: Amazon Cloud Directory tidak lagi terbuka untuk pelanggan baru. Untuk alternatif Cloud Directory, jelajahi Amazon DynamoDB dan Amazon Neptune. Jika Anda memerlukan bantuan untuk memilih alternatif yang tepat untuk kasus penggunaan Anda, atau untuk pertanyaan lain, hubungi AWS Dukungan.

Pola ini menunjukkan cara menghindari kesalahan yang Anda dapatkan saat menangani blok anonim dalam pernyataan Dynamic SQL. Anda menerima pesan galat saat menggunakan AWS Schema Conversion Tool untuk mengonversi database Oracle ke database Edisi yang kompatibel dengan Aurora PostgreSQL. Untuk menghindari kesalahan, Anda harus mengetahui nilai variabel 0UT bind, tetapi Anda tidak dapat mengetahui nilai variabel 0UT bind sampai setelah Anda menjalankan pernyataan SQL. Kesalahan dihasilkan dari AWS Schema Conversion Tool (AWS SCT) yang tidak memahami logika di dalam pernyataan Dynamic SQL. AWS SCT tidak dapat mengonversi pernyataan SQL dinamis dalam PL/SQL kode (yaitu, fungsi, prosedur, dan paket).

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif
- Contoh database Aurora PostgreSQL (DB)
- Amazon Relational Database Service (Amazon RDS) untuk instans Oracle DB
- Terminal interaktif PostgreSQL (psql)
- SQL* Ditambah
- `AWS_ORACLE_EXT`skema (bagian dari paket ekstensi AWS SCT) di database target Anda
- Versi terbaru AWS Schema Conversion Tool (AWS SCT) dan driver yang diperlukan

Arsitektur

Tumpukan teknologi sumber

- Oracle Database 10g lokal dan versi yang lebih baru

Tumpukan teknologi target

- Amazon Aurora PostgreSQL
- Amazon RDS for PostgreSQL
- Alat Konversi Skema AWS (AWS SCT)

Arsitektur migrasi

Diagram berikut menunjukkan cara menggunakan variabel 0UT pengikat AWS SCT dan Oracle untuk memindai kode aplikasi Anda untuk pernyataan SQL yang disematkan dan mengonversi kode ke format yang kompatibel yang dapat digunakan database Aurora.

Diagram menunjukkan alur kerja berikut:

1. Buat laporan AWS SCT untuk database sumber dengan menggunakan Aurora PostgreSQL sebagai basis data target.
2. Identifikasi blok anonim di blok kode SQL Dinamis (di mana AWS SCT memunculkan kesalahannya).
3. Konversi blok kode secara manual dan terapkan kode pada database target.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di AWS Cloud.
- AWS Schema Conversion Tool (AWS SCT) membantu Anda membuat migrasi database heterogen dapat diprediksi dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar objek kode database ke format yang kompatibel dengan database target.

Alat lainnya

- pgAdmin memungkinkan Anda untuk terhubung dan berinteraksi dengan server database Anda.

- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi yang dapat Anda gunakan untuk mengembangkan dan mengelola database di Oracle Database. Anda dapat menggunakan SQL* Plus atau Oracle SQL Developer untuk pola ini.

Epik

Konfigurasikan database sumber Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance Oracle di Amazon RDS atau Amazon. EC2	Untuk membuat instans Oracle DB di Amazon RDS, lihat Membuat instans Oracle DB dan menghubungkan ke database pada instans Oracle DB dalam dokumentasi Amazon RDS. Untuk membuat instans Oracle DB di Amazon Elastic Compute Cloud (Amazon EC2), lihat Amazon EC2 for Oracle di dokumentasi AWS Prescriptive Guidance.	DBA
Buat skema database dan objek untuk migrasi.	Anda dapat menggunakan Amazon Cloud Directory untuk membuat skema database. Untuk informasi selengkap nya, lihat Membuat Skema di dokumentasi Cloud Directory.	DBA
Konfigurasikan grup keamanan masuk dan keluar.	Untuk membuat dan mengonfigurasi grup keamanan, lihat Mengontrol akses dengan grup keamanan di dokumentasi Amazon RDS.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan bahwa database sedang berjalan.	Untuk memeriksa status database Anda, lihat Melihat peristiwa Amazon RDS di dokumentasi Amazon RDS.	DBA

Konfigurasikan basis data PostgreSQL Aurora target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance Aurora PostgreSQL di Amazon RDS.	Untuk membuat instance PostgreSQL Aurora, lihat Membuat cluster DB dan menghubungkan ke database pada klaster DB PostgreSQL Aurora dalam dokumentasi Amazon RDS.	DBA
Konfigurasikan grup keamanan masuk dan keluar.	Untuk membuat dan mengonfigurasi grup keamanan, lihat Menyediakan akses ke cluster DB di VPC dengan membuat grup keamanan dalam dokumentasi Aurora.	DBA
Konfirmasikan bahwa database Aurora PostgreSQL sedang berjalan.	Untuk memeriksa status database Anda, lihat Melihat peristiwa Amazon RDS di dokumentasi Aurora.	DBA

Siapkan AWS SCT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect AWS SCT ke database sumber.	Untuk menghubungkan AWS SCT ke database sumber Anda, lihat Menyambung ke PostgreSQL sebagai sumber dalam dokumentasi AWS SCT.	DBA
Connect AWS SCT ke database target.	Untuk menghubungkan AWS SCT ke database target Anda, lihat Apa itu AWS Schema Conversion Tool? dalam Panduan Pengguna AWS Schema Conversion Tool.	DBA
Konversikan skema database di AWS SCT dan simpan kode otomatis yang dikonversi sebagai file SQL.	Untuk menyimpan file yang dikonversi AWS SCT, lihat Menyimpan dan menerapkan skema yang dikonversi di AWS SCT di Panduan Pengguna Alat Konversi Skema AWS.	DBA

Migrasikan kode

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan file SQL untuk konversi manual.	Dalam file yang dikonversi AWS SCT, tarik file SQL yang memerlukan konversi manual.	DBA
Perbarui skrip.	Perbarui file SQL secara manual.	DBA

Sumber daya terkait

- Amazon RDS
- Fitur Amazon Aurora

Informasi tambahan

Kode contoh berikut menunjukkan cara mengkonfigurasi database sumber Oracle:

```
CREATE or replace PROCEDURE calc_stats_new1 (
  a NUMBER,
  b NUMBER,
  result out NUMBER)
IS
BEGIN
result:=a+b;
END;
/
```

```
set serveroutput on ;

DECLARE
  a NUMBER := 4;
  b NUMBER := 7;
  plsql_block VARCHAR2(100);
  output number;
BEGIN
  plsql_block := 'BEGIN calc_stats_new1(:a, :b,:output); END;';
  EXECUTE IMMEDIATE plsql_block USING a, b,out output;
  DBMS_OUTPUT.PUT_LINE('output:'||output);

END;
```

Contoh kode berikut menunjukkan cara mengkonfigurasi target Aurora PostgreSQL database:

```
 w integer,
 x integer)
RETURNS integer
AS
$BODY$
DECLARE
```

```
begin
return w + x ;
end;
$BODY$
LANGUAGE  plpgsql;


CREATE OR REPLACE FUNCTION test_pg.init()
RETURNS void
AS
$BODY$
BEGIN
if aws_oracle_ext.is_package_initialized
      ('test_pg' ) then
      return;
    end if;
    perform aws_oracle_ext.set_package_initialized
      ('test_pg' );

PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', NULL::INTEGER);
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_status', NULL::text);
END;
$BODY$
LANGUAGE  plpgsql;


DO $$
declare
v_sql text;
v_output_loc int;
a integer :=1;
b integer :=2;
BEGIN
perform  test_pg.init();
--raise notice 'v_sql %',v_sql;
execute 'do $a$ declare v_output_l int; begin select * from test_pg.calc_stats_new1('||
a||','||b||') into v_output_l;
PERFORM aws_oracle_ext.set_package_variable(''test_pg'', ''v_output'', v_output_l) ;
 end; $a$'  ;
v_output_loc := aws_oracle_ext.get_package_variable('test_pg', 'v_output');
raise notice 'v_output_loc %',v_output_loc;
END ;
$$
```

Membantu menegakkan penandaan DynamoDB

Mansi Suratwala, Amazon Web Services

Ringkasan

Pola ini mengatur notifikasi otomatis ketika tag Amazon DynamoDB yang telah ditentukan sebelumnya hilang atau dihapus dari sumber daya DynamoDB di Amazon Web Services (AWS) Cloud.

DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat dan dapat diprediksi dengan skalabilitas. DynamoDB memungkinkan Anda menurunkan beban administratif pengoperasian dan penskalaan database terdistribusi. Saat Anda menggunakan DynamoDB, Anda tidak perlu khawatir tentang penyediaan perangkat keras, penyiapan dan konfigurasi, replikasi, penambalan perangkat lunak, atau penskalaan klaster.

Pola ini menggunakan CloudFormation template AWS, yang membuat CloudWatch acara Amazon Events dan fungsi AWS Lambda. Acara ini mengamati informasi penandaan DynamoDB baru atau yang sudah ada dengan menggunakan AWS. CloudTrail Jika tag yang telah ditentukan hilang atau dihapus, akan CloudWatch memicu fungsi Lambda, yang mengirimi Anda notifikasi Amazon Simple Notification Service (Amazon SNS) yang memberi tahu Anda tentang pelanggaran tersebut.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Bucket Amazon Simple Storage Service (Amazon S3) untuk file.zip Lambda yang berisi skrip Python untuk menjalankan fungsi Lambda

Batasan

- Solusinya hanya berfungsi ketika `UntagResource` CloudTrail peristiwa `TagResource` atau terjadi. Itu tidak membuat pemberitahuan untuk acara lainnya.

Arsitektur

Tumpukan teknologi target

- Amazon DynamoDB
- AWS CloudTrail
- Amazon CloudWatch
- AWS Lambda
- Amazon S3
- Amazon SNS

Arsitektur target

Otomatisasi dan skala

Anda dapat menggunakan CloudFormation template AWS beberapa kali untuk Wilayah dan akun AWS yang berbeda. Anda perlu menjalankan template hanya sekali di setiap Wilayah atau akun.

Alat

Alat

- Amazon DynamoDB - DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat dan dapat diprediksi dengan skalabilitas.
- AWS CloudTrail — CloudTrail adalah layanan AWS yang membantu Anda dalam tata kelola, kepatuhan, serta audit operasional dan risiko akun AWS Anda. Tindakan yang diambil oleh pengguna, peran, atau layanan AWS dicatat sebagai peristiwa di CloudTrail.
- Amazon CloudWatch Events — Amazon CloudWatch Events menghadirkan aliran peristiwa sistem yang mendekati waktu nyata yang menjelaskan perubahan sumber daya AWS.
- AWS Lambda — Lambda adalah layanan komputasi yang mendukung kode berjalan tanpa perlu menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel yang dapat digunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan danau data.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) adalah layanan web yang memungkinkan aplikasi, pengguna akhir, dan perangkat untuk langsung mengirim dan menerima pemberitahuan dari cloud.

Kode

- File.zip proyek tersedia sebagai lampiran.

Epik

Tentukan bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan bucket S3.	Di konsol Amazon S3, pilih atau buat bucket S3 dengan nama unik yang tidak mengandung garis miring di depan. Bucket S3 ini akan meng-host file kode Lambda .zip. Bucket S3 Anda harus berada di Wilayah AWS yang sama dengan sumber daya DynamoDB yang sedang dipantau.	Arsitek Awan

Unggah kode Lambda ke bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode Lambda ke bucket S3.	Unggah file kode Lambda .zip yang disediakan di bagian Lampiran ke bucket S3. Bucket S3 harus berada di Wilayah yang sama dengan sumber daya DynamoDB yang sedang dipantau.	Arsitek Awan

Menerapkan template AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan CloudFormation template AWS.	Di CloudFormation konsol AWS, terapkan CloudFormation template AWS yang disediakan di bagian Lampiran. Dalam epik berikutnya, berikan nilai untuk parameter.	Arsitek Awan

Lengkapi parameter dalam CloudFormation template AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Beri nama ember S3.	Masukkan nama bucket S3 yang Anda buat atau pilih di epik pertama.	Arsitek Awan
Berikan kunci Amazon S3.	Berikan lokasi file kode Lambda .zip di bucket S3 Anda, tanpa garis miring di depan (misalnya,). `<folder>/<file-name>.zip`	Arsitek Awan
Berikan alamat email	Berikan alamat email aktif untuk menerima notifikasi Amazon SNS.	Arsitek Awan
Tentukan tingkat logging.	Tentukan tingkat logging dan frekuensi untuk fungsi Lambda Anda. `Info`menunjuk pesan informasi rinci tentang kemajuan aplikasi. `Error`menunjuk peristiwa	Arsitek Awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. `Warning`menunjuk situasi yang berpotensi berbahaya.	
Masukkan kunci tag DynamoDB yang diperlukan.	Pastikan tag dipisahka n dengan koma, tanpa spasi di antara mereka (misalnya,`Applicati onId,CreatedBy,Env ironment,Organizat ion`). CloudWatch Acara Acara mencari tag ini dan mengirimkan pemberitahuan jika tidak ditemukan.	Arsitek Awan

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Ketika template berhasil digunakan, ia mengirimk an email berlangganan ke alamat email yang Anda berikan. Untuk menerima pemberitahuan pelanggaran, Anda harus mengonfirmasi langganan email ini.	Arsitek Awan

Sumber daya terkait

- Membuat bucket S3
- Mengunggah file ke bucket S3

- Menandai sumber daya di DynamoDB
- Membuat aturan CloudWatch Acara yang dipicu pada panggilan AWS API menggunakan AWS CloudTrail

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menerapkan pemulihan bencana lintas wilayah dengan AWS DMS dan Amazon Aurora

Mark Hudson, Amazon Web Services

Ringkasan

Bencana alam atau yang disebabkan oleh manusia dapat terjadi kapan saja dan dapat memengaruhi ketersediaan layanan dan beban kerja yang berjalan di Wilayah AWS tertentu. Untuk mengurangi risiko, Anda harus mengembangkan rencana pemulihan bencana (DR) yang menggabungkan kemampuan lintas wilayah bawaan layanan AWS. Untuk layanan AWS yang secara inheren tidak menyediakan fungsionalitas Lintas wilayah, paket DR juga harus memberikan solusi untuk menangani failover mereka di seluruh Wilayah AWS.

Pola ini memandu Anda melalui penyiapan pemulihan bencana yang melibatkan dua cluster database Amazon Aurora MySQL Edition yang kompatibel dalam satu Wilayah. Untuk memenuhi persyaratan DR, kluster database dikonfigurasi untuk menggunakan fitur database global Amazon Aurora, dengan database tunggal yang mencakup beberapa Wilayah AWS. Tugas AWS Database Migration Service (AWS DMS) mereplikasi data antar cluster di Wilayah lokal. AWS DMS, bagaimanapun, saat ini tidak mendukung failover tugas antar Wilayah. Pola ini mencakup langkah-langkah yang diperlukan untuk mengatasi batasan tersebut dan mengonfigurasi AWS DMS secara independen di kedua Wilayah.

Prasyarat dan batasan

Prasyarat

- Wilayah AWS primer dan sekunder terpilih yang mendukung basis data global Amazon Aurora.
- Dua cluster database Edisi yang kompatibel dengan Amazon Aurora MySQL independen dalam satu akun di Wilayah utama.
- Database instance kelas db.r5 atau lebih tinggi (disarankan).
- Tugas AWS DMS di Wilayah utama yang melakukan replikasi berkelanjutan antara kluster database yang ada.
- Sumber daya Wilayah DR tersedia untuk memenuhi persyaratan pembuatan instance basis data. Untuk informasi selengkapnya, lihat Bekerja dengan instans DB di VPC.

Batasan

- Untuk daftar lengkap batasan basis data global Amazon Aurora, lihat Batasan database global Amazon Aurora.

Versi produk

- Amazon Aurora MySQL Edisi 5.7 atau 8.0 yang kompatibel dengan Amazon Aurora. Untuk informasi selengkapnya, lihat versi Amazon Aurora.

Arsitektur

Tumpukan teknologi target

- Kluster basis data global Edisi yang kompatibel dengan Amazon Aurora MySQL
- AWS DMS

Arsitektur target

Diagram berikut menunjukkan database global untuk dua Wilayah AWS, satu dengan basis data utama dan reporter utama dan replikasi AWS DMS, dan satu dengan basis data utama dan reporter sekunder.

Otomatisasi dan skala

Anda dapat menggunakan AWS CloudFormation untuk membuat infrastruktur prasyarat di Wilayah sekunder, seperti virtual private cloud (VPC), subnet, dan grup parameter. Anda juga dapat menggunakan AWS CloudFormation untuk membuat cluster sekunder di Wilayah DR dan menambahkannya ke database global. Jika Anda menggunakan CloudFormation templat untuk membuat kluster database di Wilayah utama, Anda dapat memperbarui atau menambahkannya dengan templat tambahan untuk membuat sumber daya database global. Untuk informasi selengkapnya, lihat Membuat klaster DB Amazon Aurora dengan dua instans DB dan Membuat cluster database global untuk Aurora MySQL.

Terakhir, Anda dapat membuat tugas AWS DMS di Wilayah primer dan sekunder menggunakan CloudFormation setelah peristiwa failover dan failback terjadi. Untuk informasi selengkapnya, lihat AWS::DMS::ReplicationTask.

Alat

- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang kompatibel dengan MySQL dan PostgreSQL. Pola ini menggunakan Amazon Aurora MySQL-Compatible Edition.
- Basis data global Amazon Aurora dirancang untuk aplikasi yang didistribusikan secara global. Satu database global Amazon Aurora dapat menjangkau beberapa Wilayah AWS. Ini mereplikasi data Anda tanpa berdampak pada kinerja database. Ini juga memungkinkan pembacaan lokal cepat dengan latensi rendah di setiap Wilayah, dan memberikan pemulihan bencana dari pemadaman di seluruh Wilayah.
- AWS DMS menyediakan migrasi satu kali atau replikasi yang sedang berlangsung. Tugas replikasi yang sedang berlangsung membuat database sumber dan target Anda tetap sinkron. Setelah diatur, tugas replikasi yang sedang berlangsung terus menerapkan perubahan sumber ke target dengan latensi minimal. Semua fitur AWS DMS, seperti validasi dan transformasi data, tersedia untuk tugas replikasi apa pun.

Epik

Siapkan cluster database yang ada di Region primer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memodifikasi kelompok parameter cluster database.	Dalam kelompok parameter cluster database yang ada, aktifkan pencatatan biner tingkat baris dengan mengatur **`binlog_format`** parameter ke nilai baris. AWS DMS memerlukan pencatatan biner tingkat baris untuk database yang kompatibel dengan MySQL saat melakukan replikasi atau pengambilan data perubahan (CDC) yang sedang berlangsung. Untuk	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi selengkapnya, lihat Menggunakan database AWS yang kompatibel dengan MySQL terkelola sebagai sumber AWS DMS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui periode retensi log biner database.	Menggunakan klien MySQL yang diinstal pada perangkat pengguna akhir atau instans Amazon Elastic Compute Cloud (EC2Amazon), jalankan prosedur tersimpan berikut yang disediakan oleh Amazon Relational Database Service (Amazon RDS) pada node penulis cluster database utamaXX, di mana jumlah jam untuk menyimpan log.
	`call mysql.rds_set_configuration('binlog retention hours', XX)`
	Konfirmasikan pengaturan dengan menjalankan perintah berikut.
	`call mysql.rds_show_configuration;`
	Database yang kompatibel dengan MySQL yang dikelola oleh AWS membersihkan log biner sesegera mungkin. Oleh karena itu, periode retensi harus cukup lama untuk memastikan bahwa log tidak dibersihkan sebelum tugas AWS DMS berjalan. Nilai 24 jam biasanya cukup, tetapi nilainya harus didasarkan
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pada waktu yang diperlukan untuk mengatur tugas AWS DMS di Wilayah DR.	

Perbarui tugas AWS DMS yang ada di Wilayah utama

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Rekam tugas AWS DMS ARN.	Gunakan Nama Sumber Daya Amazon (ARN) untuk mendapatkan nama tugas AWS DMS untuk digunakan nanti. Untuk mengambil ARN tugas AWS DMS, lihat tugas di konsol atau jalankan perintah berikut.
	`aws dms describe-replication-tasks`
	ARN terlihat seperti berikut ini.
	`arn:aws:dms:us-east-1:<accountid>:task:AN6HFFMPM246XOZVEUHCNSOVF7MQCLTOZUIRAMY`
	Karakter setelah titik dua terakhir sesuai dengan nama tugas yang digunakan pada langkah selanjutnya.
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah tugas AWS DMS yang ada untuk merekam pos pemeriksaan.	AWS DMS membuat pos pemeriksaan yang berisi informasi sehingga mesin replikasi mengetahui titik pemulihan untuk aliran perubahan. Untuk merekam informasi pos pemeriksaan, lakukan langkah-langkah berikut di konsol: 1. Hentikan tugas AWS DMS. 2. Gunakan editor JSON dalam tugas untuk mengatur TaskRecov eryTableEnabled parameter ke true. 3. Mulai tugas AWS DMS.	Administrator AWS
Validasi informasi pos pemeriksaan.	Menggunakan klien MySQL yang terhubung ke titik akhir penulis untuk cluster, kueri tabel metadata baru di kluster database reporter untuk memverifikasi bahwa itu ada dan berisi informasi status replikasi. Jalankan perintah berikut. `select * from awsdms_co ntrol.awsdms_txn_s tate;` Nama tugas dari ARN harus ditemukan di tabel ini di kolom. `Task_Name`	DBA

Perluas kedua cluster Amazon Aurora ke Wilayah DR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat infrastruktur dasar di Wilayah DR.	Buat komponen dasar yang diperlukan untuk pembuatan dan akses ke cluster Amazon Aurora: • Cloud privat virtual (VPC) • Subnet • Grup keamanan • Daftar kontrol akses jaringan • Grup subnet • Grup parameter DB • Grup parameter klaster basis data Pastikan bahwa konfigurasi kedua grup parameter cocok dengan konfigurasi di Wilayah utama.	Administrator AWS
Tambahkan Wilayah DR ke kedua cluster Amazon Aurora.	Tambahkan Wilayah sekunder (Wilayah DR) ke cluster Amazon Aurora utama dan reporter. Untuk informasi selengkapnya, lihat Menambahkan Wilayah AWS ke database global Amazon Aurora.	Administrator AWS

Lakukan failover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan tugas AWS DMS.	Tugas AWS DMS di Wilayah utama tidak akan berfungsi dengan baik setelah failover terjadi dan harus dihentikan untuk menghindari kesalahan.	Administrator AWS
Lakukan failover terkelola.	Lakukan failover terkelola dari cluster database utama ke Wilayah DR. Untuk petunjukn ya, lihat Melakukan failover terencana terkelola untuk database global Amazon Aurora. Setelah failover pada cluster database utama selesai, lakukan aktivitas yang sama pada kluster database reporter.	Administrator AWS, DBA
Memuat data ke dalam database utama.	Masukkan data uji ke node penulis dari database utama di cluster database DR. Data ini akan digunakan untuk memvalidasi bahwa replikasi berfungsi dengan baik.	DBA
Buat instance replikasi AWS DMS.	Untuk membuat instans replikasi AWS DMS di Wilayah DR, lihat Membuat instance replikasi.	Administrator AWS, DBA
Buat sumber AWS DMS dan titik akhir target.	Untuk membuat sumber AWS DMS dan titik akhir target di Wilayah DR, lihat Membuat titik akhir sumber dan target.	Administrator AWS, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Sumber harus menunjuk ke instance penulis dari cluster database utama. Target harus menunjuk ke instance penulis dari cluster database reporter.	
Dapatkan pos pemeriksaan replikasi.	Untuk mendapatkan pos pemeriksaan replikasi, gunakan klien MySQL untuk menanyakan tabel metadata dengan menjalankan yang berikut terhadap node penulis di kluster database reporter di Wilayah DR. `select * from awsdms_control.awsdms_txn_state;` Dalam tabel, temukan nilai task_name yang sesuai dengan ARN tugas AWS DMS yang ada di Wilayah utama yang Anda peroleh di epik kedua.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas AWS DMS.	Menggunakan konsol, buat tugas AWS DMS di Wilayah DR. Dalam tugas, tentukan metode migrasi Replikasi perubahan data saja. Untuk informasi selengkapnya, lihat Membuat tugas. 1. Dalam pengaturan tugas, gunakan wizard untuk menentukan yang berikut: • Mode mulai CDC untuk transaksi sumber - Aktifkan mode mulai CDC khusus • Titik awal CDC khusus untuk transaksi sumber - Tentukan pos pemeriksaan pemulihan 2. Di kotak pos pemeriksaan Pemulihan, masukkan nilai pos pemeriksaan replikasi yang sebelumnya diperoleh melalui kueri database di atas meja. `awsdms_txn_state` 3. Di bagian pengaturan tugas, pilih editor JSON, dan atur TaskRecoveryTableEnabledparameternya ke true.	Administrator AWS, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Setel pengaturan tugas AWS DMS Mulai tugas migrasi ke Otomatis saat membuat.	
Rekam tugas AWS DMS ARN.	Gunakan ARN untuk mendapatkan nama tugas AWS DMS untuk digunakan nanti. Untuk mengambil ARN tugas AWS DMS, jalankan perintah berikut. `aws dms describe-replication-tasks`	Administrator AWS, DBA
Validasi data yang direplikasi.	Kueri kluster database reporter di Wilayah DR untuk mengonfirmasi bahwa data pengujian yang Anda muat ke kluster basis data utama telah direplikasi.	DBA

Lakukan failback

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan tugas AWS DMS.	Tugas AWS DMS di Wilayah DR tidak akan berfungsi dengan baik setelah kegagalan terjadi dan harus dihentikan untuk menghindari kesalahan.	Administrator AWS
Lakukan failback terkelola.	Gagal kembali cluster database utama ke Region primer. Untuk petunjukn	Administrator AWS, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ya, lihat Melakukan failover terencana terkelola untuk database global Amazon Aurora. Setelah failback pada cluster database utama selesai, lakukan aktivitas yang sama pada kluster database reporter.	
Dapatkan pos pemeriksaan replikasi.	Untuk mendapatkan pos pemeriksaan replikasi, gunakan klien MySQL untuk menanyakan tabel metadata dengan menjalankan yang berikut terhadap node penulis di kluster database reporter di Wilayah DR. `select * from awsdms_control.awsdms_txn_state;` Dalam tabel, temukan `task_name` nilai yang sesuai dengan ARN tugas AWS DMS yang ada di Wilayah DR yang Anda peroleh di epik keempat.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui sumber AWS DMS dan titik akhir target.	Setelah cluster database gagal kembali, periksa cluster di Region primer untuk menentukan node mana yang merupakan instance penulis. Kemudian verifikasi sumber AWS DMS yang ada dan titik akhir target di Wilayah utama menunjuk ke instance penulis. Jika tidak, perbarui titik akhir dengan nama contoh penulis Domain Name System (DNS).	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas AWS DMS.	Menggunakan konsol, buat tugas AWS DMS di Wilayah utama. Dalam tugas, tentukan metode migrasi Replikasi perubahan data saja. Untuk informasi selengkapnya, lihat Membuat tugas. 1. Dalam pengaturan tugas, gunakan wizard dan tentukan yang berikut ini: • Mode mulai CDC untuk transaksi sumber - Aktifkan mode mulai CDC khusus • Titik awal CDC khusus untuk transaksi sumber - Tentukan pos pemeriksaan pemulihan 2. Di kotak pos pemeriksaan Pemulihan, masukkan nilai pos pemeriksaan replikasi yang sebelumnya diperoleh melalui kueri database di atas meja. `awsdms_txn_state` 3. Juga di dalam bagian pengaturan tugas, pilih editor JSON dan atur TaskRecoveryTableEnabledparameter ke true. 4. Terakhir, atur pengaturan tugas AWS DMS Start	Administrator AWS, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	migration task ke Automatic ally on create.	
Rekam tugas AWS DMS Nama Sumber Daya Amazon (ARN).	Gunakan ARN untuk mendapatkan nama tugas AWS DMS untuk digunakan nanti. Untuk mengambil ARN tugas AWS DMS, jalankan perintah berikut: `aws dms describe-replication-tasks` Nama tugas akan diperluka n saat melakukan failover terkelola lain atau selama skenario DR.	Administrator AWS, DBA
Hapus tugas AWS DMS.	Hapus tugas AWS DMS asli (saat ini dihentikan) di Wilayah utama dan tugas AWS DMS yang ada (saat ini dihentikan) di Wilayah sekunder.	Administrator AWS

Sumber daya terkait

- Mengonfigurasi cluster DB Amazon Aurora Anda
- Menggunakan basis data global Amazon Aurora
- Bekerja dengan Amazon Aurora MySQL
- Bekerja dengan instans replikasi AWS DMS
- Bekerja dengan titik akhir AWS DMS
- Bekerja dengan tugas AWS DMS
- Apa itu AWS CloudFormation?

Informasi tambahan

Basis data global Amazon Aurora digunakan dalam contoh ini untuk DR karena mereka memberikan tujuan waktu pemulihan yang efektif (RTO) 1 detik dan tujuan titik pemulihan (RPO) kurang dari 1 menit, keduanya lebih rendah dari solusi replikasi tradisional dan ideal untuk skenario DR.

Basis data global Amazon Aurora menawarkan banyak keuntungan lain, termasuk yang berikut:

- Pembacaan global dengan latensi lokal — Konsumen global dapat mengakses informasi di Wilayah lokal, dengan latensi lokal.
- Cluster Amazon Aurora DB sekunder yang dapat diskalakan — Cluster sekunder dapat diskalakan secara independen, menambahkan hingga 16 replika hanya-baca.
- Replikasi cepat dari cluster Amazon Aurora DB primer ke sekunder — Replikasi memiliki dampak kinerja yang kecil pada cluster primer. Ini terjadi pada lapisan penyimpanan, dengan latensi replikasi lintas wilayah khas kurang dari 1 detik.

Pola ini juga menggunakan AWS DMS untuk replikasi. Basis data Amazon Aurora menyediakan kemampuan untuk membuat replika baca, yang dapat menyederhanakan proses replikasi dan pengaturan DR. Namun, AWS DMS sering digunakan untuk mereplikasi ketika transformasi data diperlukan atau ketika database target memerlukan indeks tambahan yang tidak dimiliki database sumber.

Migrasikan fungsi dan prosedur Oracle yang memiliki lebih dari 100 argumen ke PostgreSQL

Srinivas Potlachervoo, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memigrasikan fungsi dan prosedur Oracle Database yang memiliki lebih dari 100 argumen ke PostgreSQL. Misalnya, Anda dapat menggunakan pola ini untuk memigrasikan fungsi dan prosedur Oracle ke salah satu layanan database AWS yang kompatibel dengan PostgreSQL berikut:

- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL
- Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL

PostgreSQL tidak mendukung fungsi atau prosedur yang memiliki lebih dari 100 argumen. Sebagai solusinya, Anda dapat menentukan tipe data baru yang memiliki bidang tipe yang cocok dengan argumen fungsi sumber. Kemudian, Anda dapat membuat dan menjalankan PL/pgSQL fungsi yang menggunakan tipe data kustom sebagai argumen.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sebuah instans database Amazon RDS Oracle (DB)
- Instans Amazon RDS for PostgreSQL DB atau instans DB yang kompatibel dengan Aurora PostgreSQL

Versi produk

- Amazon RDS Oracle DB instans versi 10.2 dan yang lebih baru
- Instans Amazon RDS PostgreSQL DB versi 9.4 dan yang lebih baru, atau instans DB yang kompatibel dengan Aurora PostgreSQL versi 9.4 dan yang lebih baru
- Oracle SQL Developer versi 18 dan yang lebih baru
- pgAdmin versi 4 dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Amazon RDS Oracle DB instans versi 10.2 dan yang lebih baru

Tumpukan teknologi target

- Instans Amazon RDS PostgreSQL DB versi 9.4 dan yang lebih baru, atau instans DB yang kompatibel dengan Aurora PostgreSQL versi 9.4 dan yang lebih baru

Alat

Layanan AWS

- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di AWS Cloud.
- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.

Layanan lainnya

- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi yang menyederhanakan pengembangan dan pengelolaan database Oracle baik dalam penerapan tradisional maupun berbasis cloud.
- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.

Praktik terbaik

Pastikan tipe data yang Anda buat cocok dengan bidang tipe yang disertakan dalam fungsi atau prosedur Oracle sumber.

Epik

Jalankan fungsi atau prosedur Oracle yang memiliki lebih dari 100 argumen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat atau mengident ifikasi Oracle/PLSQL fungsi atau prosedur yang ada yang memiliki lebih dari 100 argumen.	Buat Oracle/PLSQL fungsi atau prosedur yang memiliki lebih dari 100 argumen. -atau- Identifikasi Oracle/PLSQL fungsi atau prosedur yang ada yang memiliki lebih dari 100 argumen. Untuk informasi selengkapnya, lihat bagian 14.7 CREATE FUNCTION Statement dan 14.11 CREATE PROCEDURE Statement dalam dokumentasi Oracle Database.	Pengetahuan Oracle/PLSQL
Kompilasi Oracle/PLSQL fungsi atau prosedur.	Kompilasi Oracle/PLSQL fungsi atau prosedur. Untuk informasi selengkap nya, lihat Mengompilasi fungsi dalam dokumentasi Oracle Database.	Pengetahuan Oracle/PLSQL
Jalankan Oracle/PLSQL fungsinya.	Jalankan Oracle/PLSQL fungsi atau prosedur. Kemudian, simpan outputnya.	Pengetahuan Oracle/PLSQL

Tentukan tipe data baru yang cocok dengan argumen fungsi sumber atau prosedur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan tipe data baru di PostgreSQL.	Tentukan tipe data baru di PostgreSQL yang mencakup semua bidang yang sama yang muncul dalam argumen fungsi atau prosedur Oracle sumber. Untuk informasi selengkap nya, lihat CREATE TYPE di dokumentasi PostgreSQL.	Pengetahuan PL/pgSQL PostgreSQL

Buat fungsi PostgreSQL yang menyertakan argumen TYPE baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi PostgreSQL yang mencakup tipe data baru.	Buat fungsi PostgreSQL yang menyertakan argumen baru. TYPE Untuk meninjau fungsi contoh, lihat bagian Informasi tambahan dari pola ini.	Pengetahuan PL/pgSQL PostgreSQL
Kompilasi fungsi PostgreSQL.	Kompilasi fungsi di PostgreSQ L. Jika bidang tipe data baru cocok dengan argumen fungsi sumber atau prosedur, maka fungsi berhasil dikompilasi.	Pengetahuan PL/pgSQL PostgreSQL
Jalankan fungsi PostgreSQL.	Jalankan fungsi PostgreSQL.	Pengetahuan PL/pgSQL PostgreSQL

Pemecahan masalah

Isu	Solusi
Fungsi mengembalikan kesalahan berikut: KESALAHAN: kesalahan sintaks di dekat "" <statement>	Pastikan bahwa semua pernyataan fungsi diakhiri dengan titik koma (). `;`
Fungsi mengembalikan kesalahan berikut: ERROR: "" bukan variabel yang diketahui <variable>	Pastikan bahwa variabel yang digunakan dalam badan fungsi tercantum dalam `DECLARE` bagian fungsi.

Sumber daya terkait

- Bekerja dengan Amazon Aurora PostgreSQL (Panduan Pengguna Amazon Aurora untuk Aurora)
- BUAT JENIS (dokumentasi PostgreSQL)

Informasi tambahan

Contoh fungsi PostgreSQL yang mencakup argumen TYPE

```
CREATE OR REPLACE FUNCTION test_proc_new
(
    IN p_rec type_test_proc_args
)
RETURNS void
AS
$BODY$
BEGIN

    /*
    **************
    The body would contain code to process the input values.
    For our testing, we will display couple of values.
    ***************
    */
    RAISE NOTICE USING MESSAGE = CONCAT_WS('', 'p_acct_id: ', p_rec.p_acct_id);
    RAISE NOTICE USING MESSAGE = CONCAT_WS('', 'p_ord_id: ', p_rec.p_ord_id);
```

```
    RAISE NOTICE USING MESSAGE = CONCAT_WS('', 'p_ord_date: ', p_rec.p_ord_date);

END;
$BODY$
LANGUAGE plpgsql
COST 100;
```

Migrasikan beban kerja Redis ke Redis Enterprise Cloud di AWS

Antony Prasad Thevaraj, Amazon Web Services

Srinivas Pendyala, Redis

Ringkasan

Pola ini membahas proses tingkat tinggi untuk memigrasikan beban kerja Redis ke Redis Enterprise Cloud di Amazon Web Services (AWS). Ini menjelaskan langkah-langkah migrasi, memberikan informasi tentang pemilihan alat yang tersedia, dan membahas keuntungan, kerugian, dan langkah-langkah untuk menggunakan setiap alat. Secara opsional, jika Anda memerlukan bantuan tambahan dalam memigrasi beban kerja dari Redis, Anda dapat menggunakan Redis Professional Services.

Jika Anda menjalankan Redis OSS atau Redis Enterprise Software di tempat, Anda terbiasa dengan overhead administratif yang signifikan dan kompleksitas operasional pemeliharaan database Redis Anda di pusat data Anda. Dengan memigrasikan beban kerja Anda ke cloud, Anda dapat secara signifikan mengurangi beban operasional ini dan memanfaatkan Redis Enterprise Cloud, yang merupakan database yang dihosting penuh sebagai penawaran layanan dari DBaa Redis. Migrasi ini membantu meningkatkan kelincahan bisnis Anda, meningkatkan keandalan aplikasi, dan mengurangi biaya keseluruhan saat Anda mendapatkan akses ke fitur Redis Enterprise Cloud on AWS terbaru seperti ketersediaan 99,999%, kesederhanaan arsitektur, dan skala.

Ada aplikasi potensial untuk Redis Enterprise Cloud di sektor layanan keuangan, ritel, perawatan kesehatan, dan game, serta dalam kasus penggunaan yang memerlukan solusi untuk deteksi penipuan, inventaris waktu nyata, pemrosesan klaim, dan manajemen sesi. Anda dapat menggunakan Redis Enterprise Cloud untuk menyambung ke sumber daya AWS Anda–misalnya, ke server aplikasi yang berjalan di instans Amazon Elastic Compute Cloud (Amazon EC2), atau ke layanan mikro yang digunakan sebagai layanan AWS Lambda.

Prasyarat dan batasan

Asumsi

- Saat ini Anda sedang mengoperasikan sistem database lokal yang ingin dimigrasikan ke cloud.
- Anda telah mengidentifikasi persyaratan migrasi untuk beban kerja Anda, termasuk:
  - Persyaratan konsistensi data
  - Persyaratan infrastruktur dan lingkungan sistem
  - Pemetaan data dan persyaratan transformasi

  - Persyaratan pengujian fungsional
  - Persyaratan pengujian kinerja
  - Persyaratan validasi
  - Strategi cutover yang ditentukan
- Anda telah menilai jadwal dan perkiraan biaya yang diperlukan untuk migrasi.
- Persyaratan Anda mempertimbangkan ruang lingkup pekerjaan dan sistem serta database yang telah Anda identifikasi sebagai bagian dari migrasi.
- Anda telah mengidentifikasi para pemangku kepentingan bersama dengan peran dan tanggung jawab mereka dalam matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).
- Anda telah menerima persetujuan dan persetujuan yang diperlukan dari semua pemangku kepentingan.

Biaya

Bergantung pada spesifikasi teknis database sumber yang ada (misalnya, ukuran memori, throughput, dan ukuran data total), arsitek solusi Redis dapat mengukur sistem target di Redis Enterprise Cloud. Untuk informasi harga umum, lihat Harga Redis di situs web Redis.

Orang dan keterampilan

Proses migrasi melibatkan peran dan tanggung jawab berikut.

Peran	Deskripsi	Keterampilan yang dibutuhkan
Arsitek solusi migrasi	Seorang arsitek teknis yang memiliki keahlian dalam mendefinisikan, merencana kan, dan menerapkan strategi migrasi	Pemahaman teknis dan tingkat aplikasi tentang sistem sumber dan target; pengalaman dengan migrasi beban kerja ke cloud
Arsitek data	Seorang arsitek teknis yang memiliki pengalaman luas dalam mendefinisikan, mengimplementasikan, dan	Pemodelan data untuk data terstruktur dan tidak terstruktur, pemahaman mendalam dan pengalaman dalam mengimplementasika

	memberikan solusi data untuk berbagai database	n database untuk suatu perusahaan
Arsitek solusi Redis	Seorang arsitek teknis yang dapat membantu arsitek cluster Redis berukuran optimal untuk kasus penggunaan yang sesuai	Keahlian dalam merancang dan menerapkan solusi Redis untuk berbagai kasus penggunaan
Arsitek solusi cloud	Seorang arsitek teknis yang memiliki pemahaman yang lebih dalam tentang solusi cloud, terutama di AWS	Keahlian dalam merancang solusi untuk cloud; migrasi beban kerja dan pengalaman modernisasi aplikasi
Arsitek perusahaan	Seorang arsitek teknis yang memiliki pemahaman lengkap tentang lanskap teknis di organisasi Anda, yang memiliki visi bersama untuk peta jalan future, dan yang mempraktikkan dan menetapkan praktik terbaik arsitektur standar di semua tim di organisasi Anda	Sertifikasi arsitektur perangkat lunak seperti TOGAF, keterampilan rekayasa perangkat lunak dasar, dan arsitektur solusi dan keahlian arsitektur perusahaan
IT atau DevOps insinyur	Seorang insinyur yang bertanggung jawab untuk membuat dan memelihar a infrastruktur, termasuk memantau infrastruktur untuk masalah, melakukan tugas pemeliharaan, dan membuat pembaruan sesuai kebutuhan.	Pemahaman yang kuat tentang berbagai teknologi , termasuk sistem operasi, jaringan, dan komputasi awan; keakraban dengan bahasa pemrograman seperti Python, Bash, dan Ruby, serta alat-alat seperti Docker, Kubernetes, dan Ansible

Arsitektur

Opsi migrasi

Diagram berikut menunjukkan opsi untuk memigrasikan sumber data lokal (berbasis Redis atau lainnya) ke AWS. Ini menunjukkan beberapa alat migrasi yang dapat Anda pilih, seperti mengekspor file Redis Database (RDB) ke Amazon Simple Storage Service (Amazon S3), menggunakan fitur replikasi Redis, atau menggunakan AWS DMS.

1. Sumber data lokal: Database yang tidak didasarkan pada Redis, seperti MySQL, PostgreSQL, Oracle, SQL Server, atau MariaDB.
2. Sumber data lokal: Database berbasis protokol Redis seperti Redis OSS dan Redis Enterprise Software.
3. Cara termudah untuk memigrasikan data dari database berbasis Redis adalah dengan mengekspor file RDB dan mengimpornya ke target Redis Enterprise Cloud di AWS.
4. Atau, Anda dapat memigrasikan data dari sumber ke target dengan menggunakan fitur replikasi (`ReplicaOf`) di Redis.
5. Jika persyaratan migrasi data Anda mencakup transformasi data, Anda dapat menggunakan Redis Input/Output Tools (RIOT) untuk memigrasikan data.
6. Atau, Anda dapat menggunakan AWS Data Migration Service (AWS DMS) untuk memigrasikan data dari database berbasis SQL.
7. Anda harus menggunakan peering virtual private cloud (VPC) untuk AWS DMS agar berhasil memigrasikan data ke target Redis Enterprise Cloud di AWS.

Arsitektur target

Diagram berikut menunjukkan arsitektur penerapan khas untuk Redis Enterprise Cloud di AWS dan menggambarkan bagaimana hal itu dapat digunakan dengan layanan AWS utama.

1. Anda dapat terhubung ke aplikasi bisnis yang didukung oleh Redis Enterprise Cloud di AWS.
2. Anda dapat menjalankan aplikasi bisnis di akun AWS Anda sendiri, di VPC dalam akun tersebut.
3. Anda dapat menggunakan titik akhir database Redis Enterprise Cloud untuk terhubung ke aplikasi Anda. Contohnya termasuk server aplikasi yang berjalan pada EC2 instance, layanan mikro yang digunakan sebagai layanan AWS Lambda, aplikasi Amazon Elastic Container Service (Amazon

ECS) Container Service (Amazon ECS), atau aplikasi Amazon Elastic Kubernetes Service (Amazon EKS).

4. Aplikasi bisnis yang berjalan di VPC Anda memerlukan koneksi rekan VPC ke VPC Redis Enterprise Cloud. Hal ini memungkinkan aplikasi bisnis untuk terhubung dengan aman melalui endpoint pribadi.
5. Redis Enterprise Cloud on AWS adalah platform database NoSQL dalam memori yang digunakan sebagai DBaa S di AWS dan sepenuhnya dikelola oleh Redis.
6. Redis Enterprise Cloud digunakan dalam VPC di akun AWS standar yang dibuat oleh Redis.
7. Untuk alasan keamanan, Redis Enterprise Cloud digunakan di subnet pribadi yang dapat diakses di titik akhir pribadi dan publik. Kami menyarankan Anda menghubungkan aplikasi klien Anda ke Redis pada titik akhir pribadi. Jika Anda berencana untuk menggunakan titik akhir publik, kami sangat menyarankan agar Anda mengaktifkan TLS untuk mengenkripsi data antara aplikasi klien Anda dan Redis Enterprise Cloud.

Metodologi migrasi Redis sejalan dengan metodologi migrasi AWS, yang diilustrasikan dalam Memobilisasi organisasi Anda untuk mempercepat migrasi skala besar di situs web AWS Prescriptive Guidance.

Otomatisasi dan skala

Tugas penyiapan lingkungan untuk migrasi dapat diotomatisasi melalui templat AWS Landing Zone dan infrastruktur sebagai kode (IAc) untuk otomatisasi dan skala. Ini dibahas di bagian Epik dari pola ini.

Alat

Berdasarkan persyaratan migrasi data, Anda dapat memilih dari pilihan opsi teknologi untuk memigrasikan data Anda ke Redis Enterprise Cloud di AWS. Tabel berikut menjelaskan dan membandingkan alat-alat ini.

Alat	Deskripsi	Keuntungan	Kekurangan
Ekspor dan impor RDB	Anda mengekspo r data dari sumber (misalnya, Redis OSS atau Redis Enterprise Software)	• Sederhana. • Bekerja dengan solusi berbasis Redis apa pun yang dapat mengekspor	• Tidak menangani persyaratan transformasi data atau mendukung

	database dalam bentuk file RDB. Jika database Anda disediakan melalui Redis OSS Cluster, Anda mengekspor setiap master shard ke RDB. Anda kemudian mengimpor semua file RDB dalam satu langkah. Jika database sumber Anda didasarkan pada Cluster OSS tetapi database target Anda tidak menggunakan API Cluster OSS, Anda harus mengubah kode sumber aplikasi Anda untuk menggunakan pustaka klien Redis standar. Persyaratan transform asi data atau penggabungan database logis memerlukan proses yang lebih kompleks, yang dijelaskan di bawah Penggabun gan database logis nanti dalam tabel ini.	data dalam format RDB sebagai sumber (termasuk Redis OSS dan Redis Enterprise Software). • Mencapai konsisten si data dengan proses yang sederhana.	penggabungan database logis. • Memakan waktu untuk kumpulan data yang lebih besar. • Tidak ada dukungan migrasi delta yang dapat menyebabkan waktu henti yang lebih lama.

Fitur replikasi Redis (aktif-pasif)	Anda dapat terus mereplikasi data dari Redis OSS, Enterprise Software, atau database Enterprise Cloud ke database Redis Enterprise Cloud. Setelah sinkronisasi awal, fitur replikasi Redis (`REPLICAOF`) melakukan migrasi delta, yang berarti hampir tidak ada waktu henti aplikasi yang diamati. Fitur replikasi Redis dimaksudkan untuk digunakan dengan cara aktif-pasif. Target diasumsikan pasif dan disinkronkan ulang sepenuhnya (disiram dan disinkronkan dari database sumber). Oleh karena itu, beralih antara sumber dan target agak lebih rumit. Dimungkinkan untuk mereplikasi dari Redis OSS Cluster ke database Redis Enterprise Cloud berkerumun standar	• Mendukung replikasi berkelanjutan (beban data awal diikuti oleh delta). • Hampir tidak ada waktu henti (tergantung pada kelambatan replikasi). • Mencapai konsistensi data.	• Hanya satu situs yang dimaksudkan untuk aktif, jadi beralih antar situs lebih rumit. • Mendukung maksimal 32 pecahan master saat Anda bermigrasi dari Cluster OSS.

	dengan menentuka n semua pecahan master Cluster OSS sebagai sumber. Namun, fitur replikasi Redis memungkinkan maksimum 32 basis data sumber.		
AWS DMS	Anda dapat menggunakan AWS DMS untuk memigrasi kan data dari database sumber apa pun yang didukung ke penyimpanan data Redis target dengan waktu henti minimal. Untuk informasi selengkap nya, lihat Menggunak an Redis sebagai target AWS DMS dalam dokumentasi AWS DMS.	• Mendukung migrasi sumber data NoSQL dan SQL. • Bekerja dengan baik dengan layanan AWS lainnya. • Mendukung migrasi langsung dan mengubah kasus penggunaan pengambilan data (CDC).	• Nilai kunci Redis tidak dapat berisi karakter khusus seperti%. • Tidak mendukung migrasi data yang memiliki karakter khusus dalam baris atau nama bidang. • Tidak mendukung mode objek biner besar penuh (LOB).

Penggabungan basis data logis	Persyaratan penggabungan basis data khusus mungkin memerlukan solusi migrasi data khusus. Misalnya, Anda mungkin memiliki empat database logis (`SELECT 0..3`) di Redis OSS, tetapi Anda mungkin ingin menggunakan titik akhir database tunggal alih-alih memindahkan data ke beberapa database Redis Enterprise Cloud. Redis Enterprise tidak mendukung database logis yang dapat dipilih, jadi Anda harus mengubah model data fisik database sumber. Misalnya, Anda dapat memetakan setiap indeks database ke awalan (`0keusr`, `1 kecmp`, dan seterusnya), dan kemudian menggunakan skrip migrasi atau alat ekstrak, transformasi, dan muat (ETL) untuk menampilkan file RDB, yang kemudian	• Kontrol granular pada pembentukan data selama migrasi ke sistem target dengan menggunakan skrip kustom.	• Jika Anda memutuskan untuk tidak menyelesaikan migrasi, rollback bisa sangat menantang, terutama jika data yang lebih baru harus dikembalikan ke sistem sumber. • Biaya untuk membangun bisa tinggi jika tujuannya adalah untuk membangun solusi satu kali untuk migrasi satu kali. • Biaya pemeliharaan untuk kode, infrastruktur, waktu pengembangan, dan area lain bisa tinggi jika persyaratan migrasi sering berubah.

	dapat Anda impor ke database target.

Selain itu, Anda dapat menggunakan alat dan layanan berikut dari AWS.

Alat penilaian dan penemuan:

- Penilai Migrasi

Alat migrasi aplikasi dan server:

- Layanan Migrasi Aplikasi AWS

Alat migrasi basis data:

- Alat Konversi Skema AWS (AWS SCT)
- AWS Database Migration Service (AWS DMS)

Alat migrasi data:

- AWS Storage Gateway
- AWS DataSync
- AWS Direct Connect
- AWS Snowball
- Amazon Data Firehose

Solusi AWS Partner:

- Mitra Kompetensi Migrasi AWS

Epik

Selesaikan tugas penemuan dan penilaian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi beban kerja.	Identifikasi beban kerja kandidat yang sesuai yang ingin Anda migrasikan. Pertimbangkan hal berikut sebelum Anda memilih beban kerja untuk migrasi: • Berapa nilai bisnis dalam memigrasi atau tidak memigrasikan beban kerja ini? • Apakah ada rencana darurat jika beban kerja ini tidak berhasil bermigrasi ke sistem target? Idealnya, pilih beban kerja yang memiliki dampak bisnis maksimal dengan risiko minimum yang terlibat. Jaga agar keseluruhan proses berulang dan bermigrasi dengan sedikit demi sedikit.	Arsitek data, Juara bisnis, sponsor proyek Migrasi
Identifikasi sumber dan persyaratan data; desain model data.	Redis menjalankan lokakarya untuk mempercepat penemuan dan menentukan perencanaan migrasi untuk proyek tersebut. Sebagai bagian dari lokakarya ini, tim Redis mengidentifikasi	Arsitek solusi Redis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber data dan persyarat an model data sumber, dan menganalisis bagaimana ini dapat direnovasi di Redis Enterprise Cloud. Tim migrasi Redis (Layanan Profesional) melakukan latihan desain model data terperinc i dengan organisasi Anda. Sebagai bagian dari latihan ini, tim Redis: • Mengidentifikasi struktur data Redis target. • Mendefinisikan strategi pemetaan data. • Mendokumentasikan pendekatan migrasi dan rekomendasi. • Meninjau dan menyelesa ikan model data dengan para pemangku kepenting an.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi karakteristik database sumber.	Identifikasi produk Redis yang digunakan di lingkungan sumber dan target. Contoh: • Apakah database sumber database OSS Cluster, database Redis mandiri, atau database Redis Enterprise? • Apakah database target akan menjadi database standar Redis Enterpris e atau database yang kompatibel dengan OSS Cluster? • Apa implikasinya mengenai kode sumber aplikasi?	Arsitek data
Kumpulkan SLA sistem saat ini dan metrik ukuran lainnya.	Tentukan perjanjian tingkat layanan saat ini (SLAs) yang dinyatakan dalam hal throughput (operasi per detik), latensi, ukuran memori keseluruhan per database, dan persyaratan ketersediaan tinggi (HA).	Arsitek data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi karakteristik sistem target.	Tentukan jawaban atas pertanyaan-pertanyaan ini: • Berapa banyak data yang harus dimigrasikan? • Berapa lama waktu yang dibutuhkan untuk memigrasikan jumlah data yang diberikan? • Apa saja persyaratan downtime untuk migrasi? Apakah dapat diterima jika layanan atau aplikasi Anda tidak tersedia untuk periode tertentu? Jika demikian, untuk berapa lama? • Seberapa konsisten data yang dimigrasi? Bisakah database target berada dalam keadaan yang sedikit tidak konsisten (usang)? • Apakah data harus diubah sebelum dimuat ke database target? (Misalnya, Anda mungkin ingin mengonversi indeks DB yang dapat dipilih menjadi awalan sebelum migrasi.) • Apakah database sumber dapat dijangkau dari host database target (misalnya, dari VPC rekan atau dari titik akhir publik menggunakan enkripsi)?	Arsitek data, arsitek solusi Redis (opsional)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Selesaikan ukuran data dan latihan ukuran Redis Cluster dengan arsitek teknis Redis. • Identifikasi persyaratan jaringan, persyaratan infrastruktur, versi perangkat lunak, dan lisensi perangkat lunak, dan dapatkan komponen apa pun sebelum migrasi. • Apakah ada masalah keamanan yang terkait dengan transfer data ini?	
Identifikasi dependensi.	Identifikasi dependensi hulu dan hilir dari sistem saat ini yang akan dimigrasikan. Pastikan bahwa pekerjaan migrasi selaras dengan migrasi sistem dependen lainnya. Misalnya, jika Anda berencana untuk memigrasi kan aplikasi bisnis lain dari lokasi ke AWS Cloud, identifik asi aplikasi ini dan selaraskan berdasarkan sasaran proyek, garis waktu, dan pemangku kepentingan.	Arsitek data, arsitek perusahaan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi alat migrasi.	Bergantung pada persyaratan migrasi data Anda (seperti data sumber atau persyaratan waktu henti), Anda dapat menggunakan salah satu alat yang dijelaskan sebelumnya di bagian Alat. Selain itu, Anda dapat menggunakan: • Replikasi dua arah (aktif-aktif) dengan menggunakan penerapan CRDB. • export/import Skrip kustom (misalnya, dengan menggunakan `DUMP/RESTORE` perintah). • export/import Alat tambahan dan alat pembantu seperti RIOT, ECstats2, atau alat ETL. • Alat IAC seperti templat Terraform atau AWS. CloudFormation	Arsitek solusi migrasi, arsitek solusi Redis
Buat rencana kontingensi.	Buat rencana darurat untuk memutar kembali, jika Anda mengalami masalah selama migrasi.	Manajemen proyek, tim teknis, termasuk arsitek

Selesaikan tugas keamanan dan kepatuhan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Amankan konsol administrasi Redis.	Untuk mengamankan konsol administrasi, ikuti instruksi dalam dokumentasi Redis.	Administrator infrastruktur TI
Amankan database Redis.	Lihat halaman berikut dalam dokumentasi Redis untuk: • Tentukan kontrol akses berbasis peran. • Tentukan keamanan jaringan. • Aktifkan TLS.	
Aman Redis Cloud APIs.	Saat mengaktifkan API, Anda dapat mengelola kunci API untuk semua pemilik akun Redis Cloud Anda. Untuk ikhtisar fitur keamanan API, lihat dokumentasi autentikasi API di situs web Redis.	Administrator infrastruktur TI

Siapkan lingkungan baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan lingkungan baru di AWS.	Tugas ini meliputi:
	• Aktivitas penyiapan AWS Landing Zone. Landing zone mendukung:
	  • Penerapan multi-akun
	  • Baseline keamanan minimum
		IT atau DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Cara otomatis untuk menyediakan akun baru dengan garis dasar keamanan dan prasyarat ISV (jaringan , konfigurasi keamanan, dan sebagainya) • Pemberitahuan, pencatatan terpusat, dan pemantauan • Aktivitas konfigurasi perangkat lunak ISV. Ini termasuk konfigurasi yang perlu disertakan dalam migrasi, seperti pengaturan dan perubahan produk dan beban kerja. • Aktivitas IAC seperti mengonfigurasi atau menyesuaikan templat CloudFormation AWS atau Terraform.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan arsitektur migrasi.	1. Siapkan Redis Enterprise Cloud di AWS. 2. Instal alat migrasi seperti RIOT atau AWS DMS. Lihat bagian Alat untuk daftar alat yang tersedia. 3. Membangun konektivitas antara aplikasi, migrasi, dan lapisan basis data. 4. Buat contoh beban kerja yang dapat mengalir melalui setiap lapisan dan memigrasikan sekumpulan kecil data sampel. Anda sekarang siap untuk menjalankan pipeline migrasi data aktual dan mengujinya.	IT atau DevOps insinyur

Mengatur jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membangun konektivitas.	Membangun konektivitas antara infrastruktur lokal dan sumber daya AWS Cloud. Gunakan grup keamanan, AWS Direct Connect, dan sumber daya lainnya untuk mencapai fungsionalitas ini. Untuk informasi selengkap nya, lihat Connect Your Data	IT atau DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Center ke AWS di situs web AWS.	
Siapkan pengintip VPC.	Buat peering VPC antara aplikasi bisnis VPCs yang menjalankan (atau EC2 instance yang menjalank an alat migrasi atau server replikasi AWS DMS) dan VPC yang menjalankan Redis Enterprise Cloud. Untuk petunjuknya, lihat Memulai Amazon VPC dalam dokumentasi Amazon VPC, dan Aktifkan peering VPC di dokumentasi Redis.	IT atau DevOps insinyur

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih alat migrasi data.	Tinjau tabel di bagian Alat untuk melihat deskripsi, kelebihan, dan kekurangan alat ini: • Ekspor dan impor RDS • Fitur replikasi Redis () `ReplicaOf` • AWS DMS • Penggabungan basis data logis	Arsitek solusi migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Baris berikut menjelaskan tugas migrasi data yang terkait dengan setiap alat.	
Opsi 1: Gunakan ekspor dan impor RDB.	1. Putuskan sambungan sumber: Hentikan lalu lintas pada basis data sumber (misalnya, dengan memutuskan sambungan aplikasi bisnis). 2. Ekspor: Ekspor data database sumber sebagai file RDB. 3. Tahap: Unggah data ke lokasi yang dapat diakses oleh instans Redis Enterprise Cloud di AWS (misalnya, Anda dapat mengunggahnya ke bucket S3 atau server FTP). 4. Impor: Impor file RDB (dengan mencantumkan semuanya dalam satu langkah impor) ke database target Redis Enterprise Cloud Anda. 5. Cut over: Pindah ke database target (misalnya, dengan menghubungkan aplikasi Anda terhubung ke sana). Untuk informasi lebih lanjut, lihat dokumentasi Redis.	Arsitek solusi migrasi, arsitek solusi Redis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 2: Gunakan fitur replikasi Redis (aktif-pasif).	1. Connect Database: Buat `ReplicaOf` tautan antara database sumber dan target. 2. Jalankan sinkronisasi awal: Tunggu hingga sinkronis asi awal antara basis data sumber dan target selesai. 3. Putuskan sambungan sumber: Hentikan lalu lintas pada basis data sumber (misalnya, dengan memutuskan sambungan aplikasi). 4. Jalankan replikasi delta: Tunggu hingga delta direplikasi pada database target. 5. Cut over: Pindah ke database target (misalnya , dengan menghubungkan aplikasi Anda ke sana). 6. Hapus: Hapus `ReplicaOf` tautan antara basis data sumber dan target. Untuk informasi lebih lanjut, lihat dokumentasi Redis.	Arsitek solusi migrasi, arsitek solusi Redis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 3: Gunakan AWS DMS.	1. Siapkan instans replikasi AWS DMS: Instans ini menjalankan semua proses migrasi. Untuk petunjuk: Bekerja dengan instans replikasi AWS DMS dalam dokumentasi AWS DMS. 2. Tentukan database sumber: Tentukan titik akhir sumber. Uji konektivitas antara titik akhir sumber dan server replikasi AWS DMS. Untuk petunjuk: Membuat titik akhir sumber dan target dalam dokumentasi AWS DMS. 3. Siapkan database target: Siapkan Redis Enterprise Cloud di AWS dan siapkan database untuk bermigrasi. 4. Tentukan database target: Tentukan titik akhir target. Pastikan bahwa peering VPC dibuat antara VPC tempat AWS DMS berjalan dan VPC yang menghosti ng Redis Enterprise Cloud di AWS. Uji konektivitas antara server replikasi AWS DMS dan database target. 5. Buat tugas AWS DMS: Buat tugas atau serangkaian tugas untuk menentukan tabel dan proses replikasi	Arsitek solusi migrasi, arsitek solusi Redis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang ingin Anda gunakan untuk memigrasikan data. Untuk petunjuk: Bekerja dengan tugas AWS DMS dalam dokumentasi AWS DMS. 6. Migrasi: Migrasikan data dengan menjalankan tugas AWS DMS. 7. Cut over: Pindah ke database target (misalnya , dengan menghubungkan aplikasi Anda ke sana).	
Opsi 4: Gunakan penggabungan database logis.	Opsi ini melibatkan penggunaan skrip migrasi atau alat ETL yang dapat mengubah model data fisik basis data sumber dan menghasilkan file RDB. Redis Professional Services dapat membantu dengan langkah ini, jika diperlukan.	Arsitek solusi migrasi, arsitek solusi Redis

Migrasi aplikasi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sejajarkan jadwal dan tujuan manajemen proyek.	Sejajarkan tujuan proyek migrasi, tonggak sejarah, dan garis waktu lapisan aplikasi dengan proyek migrasi data Redis.	Manajemen proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sejajarkan kegiatan pengujian.	Setelah layer aplikasi dimigrasikan dan dimoderni sasi di AWS Cloud, arahkan layer aplikasi ke Redis Enterprise Cloud yang baru dimigrasi di AWS untuk pengujian.	Pengujian

Uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Melaksanakan rencana pengujian.	Jalankan rutinitas migrasi data dan skrip yang dikembangkan selama fase implementasi di lingkungan pengujian, sesuai persyaratan pengujian, di situs Anda.	Pengujian
Uji kualitas data.	Uji kualitas data setelah Anda memigrasikan data.	Pengujian
Fungsionalitas uji.	Uji kueri data dan lapisan aplikasi untuk memastika n bahwa aplikasi berkinerj a pada tingkat yang sama seperti di sistem sumber.	Pengujian

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat keputusan cutover.	Setelah semua pengujian tingkat aplikasi dan tingkat	Manajemen proyek, Juara bisnis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	basis data selesai, tim kepemimpinan eksekutif dan pemangku kepenting an membuat keputusan akhir mengenai apakah akan memotong ke lingkungan baru di AWS berdasarkan hasil akhir yang dikonfirmasi oleh tim pengujian.	
Potong ke AWS Cloud.	Ketika Anda telah mengonfir masi bahwa semuanya sudah ada, arahkan layer aplikasi ke data yang baru dimigrasi dan arahkan klien ke lapisan aplikasi baru yang berjalan berdasarkan sistem Redis Enterprise Cloud baru di AWS.	TI atau DevOps insinyur, Arsitek data, Arsitek solusi migrasi, arsitek solusi Redis

Sumber daya terkait

Sumber daya Redis

- Dokumentasi Redis Enterprise Cloud
- Alat RIOT (GitHub repositori)
- Penyedia Terraform (unduh)

Sumber daya AWS

- Migrasi demo
- Solusi AWS Partner
- Dokumentasi
- Postingan blog
- Kertas putih

- Tutorial dan video
- Migrasi cloud AWS
- AWS Prescriptive Guidance

Informasi tambahan

Untuk persyaratan keamanan standar untuk memigrasikan beban kerja Redis ke AWS Cloud, lihat Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan di situs web AWS, dan Pusat Kepercayaan Redis di situs web Redis.

Migrasikan SAP HANA ke AWS menggunakan SAP HSR dengan nama host yang sama

Pradeep Puliyampatta, Amazon Web Services

Ringkasan

Migrasi SAP HANA ke Amazon Web Services (AWS) dapat dilakukan dengan menggunakan beberapa opsi, termasuk pencadangan dan pemulihan, ekspor dan impor, dan SAP HANA System Replication (HSR). Pemilihan opsi tertentu tergantung pada konektivitas jaringan antara sumber dan target database SAP HANA, ukuran database sumber, pertimbangan downtime, dan faktor lainnya.

Opsi SAP HSR untuk memigrasikan beban kerja SAP HANA ke AWS berfungsi dengan baik ketika ada jaringan yang stabil antara sumber dan sistem target dan seluruh database (snapshot replikasi SAP HANA DB) dapat sepenuhnya direplikasi dalam 1 hari, sebagaimana ditetapkan oleh SAP untuk persyaratan throughput jaringan untuk SAP HSR. Persyaratan downtime dengan pendekatan ini terbatas untuk melakukan pengambilalihan pada AWS lingkungan target, cadangan SAP HANA DB, dan tugas pasca-migrasi.

SAP HSR mendukung penggunaan nama host yang berbeda (nama host yang dipetakan ke alamat IP yang berbeda) untuk lalu lintas replikasi antara sistem primer, atau sumber, dan sekunder, atau target. Anda dapat melakukan ini dengan mendefinisikan kumpulan nama host tertentu di bawah `[system_replication_hostname_resolution]` bagian di. `global.ini` Pada bagian ini, semua host dari situs primer dan sekunder harus ditentukan pada setiap host. Untuk langkah-langkah konfigurasi terperinci, lihat dokumentasi SAP.

Salah satu kunci utama dari pengaturan ini adalah bahwa nama host di sistem primer harus berbeda dari nama host di sistem sekunder. Jika tidak, kesalahan berikut dapat diamati.

- `"each site must have a unique set of logical hostnames"`
- `"remoteHost does not match with any host of the source site. All hosts of source and target site must be able to resolve all hostnames of both sites correctly"`

Namun, jumlah langkah pasca-migrasi dapat dikurangi dengan menggunakan nama host SAP HANA DB yang sama pada lingkungan target. AWS

Pola ini memberikan solusi untuk menggunakan nama host yang sama pada lingkungan sumber dan target saat menggunakan opsi SAP HSR. Dengan pola ini, Anda dapat menggunakan opsi

SAP HANA Hostname Rename. Anda menetapkan nama host sementara ke target SAP HANA DB untuk memfasilitasi keunikan nama host untuk SAP HSR. Setelah migrasi menyelesaikan tonggak pengambilalihan pada lingkungan SAP HANA target, Anda dapat mengembalikan nama host sistem target kembali ke nama host sistem sumber.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Virtual Private Cloud (VPC) dengan titik akhir virtual private network (VPN) atau router.
- AWS Client VPN atau AWS Direct Connect dikonfigurasi untuk mentransfer file dari sumber ke target.
- Database SAP HANA di lingkungan sumber dan target. Target level patch SAP HANA DB harus sama atau lebih tinggi dari level patch SAP HANA DB sumber, dalam edisi Platform SAP HANA yang sama. Misalnya, replikasi tidak dapat diatur antara sistem HANA 1.0 dan HANA 2.0. Untuk informasi lebih lanjut, lihat pertanyaan 15 di SAP Catatan: 1999880 - FAQ: SAP HANA System Replication.
- Server aplikasi SAP di lingkungan target.
- Volume Amazon Elastic Block Store (Amazon EBS) di lingkungan target.

Batasan

Daftar dokumen SAP berikut mencakup masalah yang diketahui terkait dengan solusi ini, termasuk kendala terkait tiering dinamis SAP HANA dan migrasi scale-out:

- 2956397 — Gagal mengganti nama Sistem Database SAP HANA
- 2222694 - Saat mencoba mengganti nama sistem HANA, kesalahan berikut muncul “File sumber tidak dimiliki oleh pengguna sidadm asli (uid = xxxx)”
- 2607227 - hdblcm: register_rename_system: Gagal mengganti nama instance SAP HANA
- 2630562 - HANA Hostname Rename gagal dan HANA tidak memulai
- 2935639 - sr_register tidak menggunakan nama host yang ditentukan di bawah system_replication_hostname_resolution di bagian global.ini
- 2710211 - Kesalahan: sistem sumber dan sistem target memiliki nama host logis yang tumpang tindih
- 2693441 - Gagal mengganti nama Sistem SAP HANA karena kesalahan

- 2519672 - HANA Primer dan Secondary memiliki sistem PKI SSFS data dan kunci yang berbeda atau tidak dapat memeriksa
- 2457129 - Penggantian Nama Host Sistem SAP HANA Tidak Diizinkan saat Dynamic Tiering Adalah Bagian dari Lanskap
- 2473002 - Menggunakan Replikasi Sistem HANA untuk memigrasikan sistem skala keluar (Tidak ada batasan yang disediakan oleh SAP dalam menggunakan pendekatan penggantian nama host ini untuk sistem SAP HANA skala keluar. Namun, prosedur harus diulang pada setiap host individu. Batasan migrasi scale-out lainnya juga berlaku untuk pendekatan ini.)

Versi produk

- Solusi ini berlaku untuk platform SAP HANA DB edisi 1.0 dan 2.0.

Arsitektur

Pengaturan sumber

Database SAP HANA diinstal pada lingkungan sumber. Semua koneksi server aplikasi SAP dan antarmuka DB menggunakan nama host yang sama untuk koneksi klien. Diagram berikut menunjukkan contoh nama host sumber `hdbhost` dan alamat IP yang sesuai.

Pengaturan target

Lingkungan AWS Cloud target menggunakan nama host yang sama untuk menjalankan database SAP HANA. Lingkungan target di AWS mencakup hal-hal berikut:

- Basis data SAP Hana
- Server aplikasi SAP
- EBS volume

Konfigurasi menengah

Dalam diagram berikut, nama host pada lingkungan AWS target sementara diganti namanya `temp-host` sehingga nama host pada sumber dan target unik. Setelah migrasi menyelesaikan

tonggak pengambilalihan pada lingkungan target, nama host virtual sistem target diganti namanya menggunakan nama asli,. `hdbhost`

Konfigurasi perantara mencakup salah satu opsi berikut:

- AWS Client VPN dengan titik akhir Client VPN
- Direct Connect menghubungkan ke router

Server aplikasi SAP pada lingkungan AWS target dapat diinstal baik sebelum pengaturan replikasi atau setelah pengambilalihan. Namun, menginstal server aplikasi sebelum penyiapan replikasi dapat membantu mengurangi waktu henti selama instalasi, konfigurasi ketersediaan tinggi, dan pencadangan.

Alat

Layanan AWS

- AWS Client VPNadalah layanan VPN berbasis klien terkelola yang memungkinkan Anda mengakses AWS sumber daya dan sumber daya dengan aman di jaringan lokal Anda.
- AWS Direct Connectmenghubungkan jaringan internal Anda ke Direct Connect lokasi melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke publik Layanan AWS, melewati penyedia layanan internet di jalur jaringan Anda.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon Elastic Compute Cloud (EC2Amazon). Volume EBS berfungsi seperti perangkat blok mentah yang tidak terformat. Anda dapat memasang volume ini sebagai perangkat di instans Anda.

Alat lainnya

- Server aplikasi SAP — Server aplikasi SAP menyediakan programmer dengan cara untuk mengekspresikan logika bisnis. Server aplikasi SAP melakukan pemrosesan data berdasarkan logika bisnis. Data aktual disimpan dalam database, yang merupakan komponen terpisah.
- Kokpit SAP HANA dan SAP HANA Studio - Baik kokpit SAP HANA dan SAP HANA Studio menyediakan antarmuka administratif ke database SAP HANA. Di SAP HANA Studio, konsol Administrasi SAP HANA adalah tampilan sistem yang menyediakan konten yang relevan untuk administrasi basis data SAP HANA.

- SAP HANA System Replication - SAP HANA System Replication (SAP HSR) adalah prosedur standar yang disediakan oleh SAP untuk mereplikasi database SAP HANA. Executable yang diperlukan untuk SAP HSR adalah bagian dari kernel server SAP HANA itu sendiri.

Epik

Siapkan lingkungan sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan konfigurasikan database SAP HANA.	Di lingkungan sumber dan target, pastikan bahwa SAP HANA DB diinstal dan dikonfigurasi sesuai dengan SAP HANA pada praktik terbaik. Untuk informasi lebih lanjut, lihat SAP HANA di AWS.	Administrasi SAP Basis
Memetakan alamat IP.	Di lingkungan target, pastikan bahwa nama host sementara ditetapkan ke alamat IP internal. 1. Tetapkan IPv4 alamat sekunder ke EC2 instans di AWS Management Console dengan menavigasi ke EC2, Instans, Tindakan, Jaringan, Kelola alamat IP, Tetapkan alamat IP baru. 2. Untuk menetapkan alamat yang sama ke adaptor EC2 jaringan (NIC), dari sistem operasi, sebagai pengguna root, jalankan perintah`ip addr add <IP>/32 dev`	Administrasi AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`eth0`, ganti `<IP>` dengan alamat IP dari langkah 1.	
Selesaikan nama host target.	Pada SAP HANA DB sekunder, konfirmasikan bahwa kedua nama host (`hdbhostdantemp-host`) diselesaikan untuk jaringan replikasi SAP HANA dengan memperbarui nama host yang relevan dalam file. `/etc/hosts`	Administrasi Linux
Cadangkan sumber dan targetkan database SAP HANA.	Gunakan SAP HANA Studio atau kokpit SAP HANA untuk melakukan backup pada database SAP HANA.	Administrasi SAP Basis
Sistem pertukaran sertifikat PKI.	(Hanya berlaku untuk SAP HANA 2.0 dan yang lebih baru) Sertifikat pertukaran di penyimpanan aman infrastru ktur kunci publik sistem (PKI) di penyimpanan sistem file (SSFS) antara database primer dan sekunder. Untuk informasi selengkapnya, lihat SAP Note 2369981 — Langkah-langkah konfigura si yang diperlukan untuk otentikasi dengan SAP HANA System Replication.	Administrasi SAP Basis

Ganti nama target SAP HANA DB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan koneksi klien target.	Di lingkungan target, matikan server aplikasi SAP dan koneksi klien lainnya.	Administrasi SAP Basis

Ubah nama target SAP HANA DB menjadi nama host sementara.	1. Sebagai pengguna root, ganti nama target SAP HANA DB hostname menjadi hostname sementara dengan menggunakan resident. `hdblcm`
	`root $> cd /hana/shared/<SID/hdblcm`
	`root $> ./hdblcm`
	2. Pilih opsi`9 | rename_system | Rename the SAP HANA Database System`.
	3. Berikan nama baru: `temp-host .`
	4. Anda dapat memvalidasi opsi lain sesuai kebutuhan . Namun, pastikan Anda tidak mencampur nama host dengan perubahan SID (SAP Note 2598814 - hdblcm: penggantian nama SID gagal).
		Administrasi SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Stop dan start SAP HANA DB akan dikendalikan olehhdblcm.	
Tetapkan jaringan replikasi.	Dalam global.ini file sistem sumber, di bawah [system_replication_hostname_resolution] header, berikan rincian jaringan replikasi sumber dan target. Kemudian salin entri ke global.ini file pada sistem target.	Administrasi SAP Basis
Aktifkan replikasi pada primer.	Untuk mengaktifkan replikasi pada sumber SAP HANA DB, jalankan perintah berikut. `hdbnsutil -sr_enable --name=siteA`	Administrasi SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Daftarkan target SAP HANA DB sebagai sistem sekunder.	Untuk mendaftarkan target SAP HANA DB sebagai sistem sekunder ke sumber SAP HSR, pilih replikasi async.
	`(sid)adm $> HDB stop`
	`(sid)adm $> hdbnsutil -sr_register –name=siteB –remotehost=hdbhost /`
	`--remoteInstance=00 –replicationMode=async –operationMode=logreplay`
	`(sid)adm $> HDB start`
	Atau, Anda dapat memilih `–online` opsi untuk mendaftar. Dalam hal ini, Anda tidak perlu berhenti dan memulai SAP HANA DB.
		Administrasi SAP Basis

Validasi sinkronisasi.	Pada sumber SAP HANA DB, verifikasi bahwa semua log diterapkan pada sistem target (karena ini adalah replikasi asinkron).
	Untuk memverifikasi replikasi , pada sumbernya, jalankan perintah berikut.
	`(sid)adm $> cdpy`
	`(sidadm $> python systemReplicationStatus.py`
		Administrasi SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan aplikasi SAP sumber dan SAP HANA DB.	Selama cutover migrasi, lakukan shutdown sistem sumber (aplikasi SAP dan database SAP HANA.	Administrasi SAP Basis
Lakukan pengambilalihan pada target.	Untuk melakukan pengambil alihan pada target di AWS, jalankan perintah`hdbnsutil -sr_takeover` .	Administrasi SAP Basis
Pada target SAP HANA DB, matikan replikasi.	Untuk menghapus metadata replikasi, hentikan replikasi pada sistem target dengan menjalankan perintah. `hdbnsutil -sr_disable` **Note** Ini sesuai dengan SAP Note 2693441 - Gagal mengganti nama Sistem SAP HANA karena kesalahan.	Administrasi SAP Basis
Cadangkan target SAP HANA DB.	Setelah pengambilalihan berhasil, kami sarankan untuk melakukan cadangan SAP HANA DB penuh.	Administrasi SAP Basis

Kembalikan ke nama host asli di sistem target

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kembalikan nama host SAP HANA DB target ke aslinya.	1. Untuk mengembalikan nama host SAP HANA DB target ke nama host virtual asli, gunakan residen. `hdblcm`
	
	`root $> cd /hana/shared/<SID>/hdblcm`
	`root $> ./hdblcm`
	
	2. Pilih opsi9 `| rename_system | Rename the SAP HANA Database System`.
	3. Berikan nama baru:`hdbhost`.
	
	Anda dapat memvalidasi opsi lain sesuai kebutuhan . Namun, pastikan Anda tidak mencampur nama host dengan perubahan SID (SAP Note 2598814 - hdblcm: penggantian nama SID gagal).
		Administrasi SAP Basis

Sesuaikan hdbuserstore.	Sesuaikan `hdbuserstore` detail yang menunjuk ke `schema/user` detail sumber. Untuk langkah-langkah rinci, lihat dokumentasi SAP.	Administrasi SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk memvalidasi langkah ini, jalankan perintah`R3trans -d`. Hasilnya harus mencerminkan koneksi yang berhasil ke database SAP HANA.	
Mulai koneksi klien.	Di lingkungan target, mulai server aplikasi SAP dan koneksi klien lainnya.	Administrasi SAP Basis

Sumber daya terkait

Referensi SAP

Referensi dokumentasi SAP sering diperbarui oleh SAP. Untuk tetap up to date, lihat SAP Note 2407186 — Panduan Cara & Whitepaper Untuk Ketersediaan Tinggi SAP HANA.

Catatan SAP tambahan

- 2550327 - Cara Mengganti Nama Sistem SAP HANA
- 1999880 - FAQ: Replikasi Sistem SAP HANA
- 2078425 - Catatan pemecahan masalah untuk alat manajemen siklus hidup platform SAP HANA hdblcm
- 2592227 — Perubahan akhiran FQDN dalam sistem HANA
- 2048681 - Melakukan tugas administrasi manajemen siklus hidup platform SAP HANA pada sistem multi-host tanpa kredensi SSH atau root

Dokumen SAP

- Koneksi Jaringan Replikasi Sistem
- Resolusi Nama Host untuk Replikasi Sistem

AWS referensi

- Migrasi SAP HANA dari Platform Lain ke AWS

Informasi tambahan

Perubahan yang dilakukan oleh `hdblcm` sebagai bagian dari aktivitas penggantian nama host dikonsolidasikan dalam log verbose berikut.

Migrasi grup ketersediaan Microsoft SQL Server Always On menggunakan AWS Application Migration Service

Sreenivas Nettem, Bharath Kumar Pammi Ramesh, Anantharaman Seshadri, dan Gireesh Sreekantan, Amazon Web Services

Ringkasan

AWS Application Migration Service (AWS MGN) adalah alat pilihan untuk rehosting lingkungan yang ada di AWS Cloud, yang memungkinkan pelanggan untuk menjauh dari pusat data lokal. Pola ini menguraikan proses menggunakan AWS MGN untuk memigrasikan cluster Windows dengan grup ketersediaan Microsoft SQL Server Always On.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Peran AWS Identity and Access Management (IAM) untuk orkestrasi AWS MGN.
- Akses ke server database sumber (grup ketersediaan SQL Server Always On).
- Active Directory di AWS landing zone untuk mempertahankan nama DNS.
- Subnet pementasan dengan komunikasi jaringan tertutup ke Active Directory.
- Target subnet yang dapat berkomunikasi dengan Active Directory.
- Dua alamat IP cadangan untuk cluster Windows di subnet target (satu di setiap Availability Zone).
- Dua alamat IP cadangan untuk pendengar SQL Always On di subnet target (satu di setiap Availability Zone).

Versi produk

- Windows Server 2012 atau yang lebih baru
- SQL Server 2012 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

Cluster Microsoft Windows (mesin fisik atau virtual lokal) Grup ketersediaan Microsoft SQL Server Selalu Aktif

Tumpukan teknologi target

Contoh Amazon EC2 Windows

Arsitektur target

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Application Migration Servicembantu Anda meng-host ulang (mengangkat dan menggeser) aplikasi ke aplikasi AWS Cloud tanpa perubahan dan dengan waktu henti minimal.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Alat-alat lainnya

- Microsoft SQL Server Management Studio (SSMS) adalah alat untuk mengelola SQL Server, termasuk mengakses, mengkonfigurasi, dan mengelola komponen SQL Server.

Praktik terbaik

Untuk AWS MGN, lihat Praktik terbaik untuk AWS Application Migration Service.

Epik

Siapkan akun target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi AWS MGN.	Inisialisasi AWS MGN di target. Wilayah AWS Ini menciptakan peran dan kebijakan IAM yang diperluka n. Untuk informasi selengkap nya, lihat Menginisialisasi Layanan Migrasi Aplikasi dengan konsol.	Administrator awan
Buat replikasi dan luncurkan templat.	Konfigurasikan templat replikasi dan luncurkan untuk digunakan dengan AWS MGN. Untuk informasi selengkapnya, lihat Mengonfigurasi templat dalam AWS dokumentasi.	Administrator awan
Izinkan port komunikasi.	Untuk mengaktifkan komunikasi jaringan untuk AWS MGN, izinkan lalu lintas melalui port TCP 443 dan 1500. Untuk informasi selengkapnya, lihat Persyarat an jaringan untuk Layanan Migrasi Aplikasi dalam AWS dokumentasi.	Administrator awan, Administr ator jaringan

Siapkan server sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi prasyarat AWS MGN.	Verifikasi bahwa server sumber memenuhi prasyarat untuk instalasi agen MGN. AWS Untuk informasi selengkapnya, lihat Persyaratan instalasi dalam AWS dokumentasi.	Insinyur migrasi
Instal agen AWS MGN.	Instal agen AWS MGN di server sumber. Selama instalasi, pilih Wilayah AWS tempat server akan dimigrasikan. Setelah instalasi, agen berkomunikasi dengan layanan dan memulai replikasi. Untuk informasi selengkapnya, lihat Menginstal Agen AWS Replikasi di server Windows.	Insinyur migrasi
Periksa status server sumber.	Di konsol AWS MGN, periksa status server sumber. Server menampilkan Siap untuk pengujian saat replikasi dimulai. Jika Anda menemukan kesalahan, lihat Memecahkan masalah kesalahan komunikasi dalam dokumentasi AWS MGN.	Administrator cloud, Insinyur migrasi
Optimalkan pengaturan replikasi.	SQL Always On cluster menggunakan replikasi I/	Administrator cloud, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	O sinkron tinggi dari server utama ke server sekunder. Untuk mengoptimalkan replikasi dan menghindari lag, gunakan server replikasi khusus untuk setiap server SQL Always On. Jika database lebih besar dari 5 TB, pertimbangkan untuk memilih ukuran instance server replikasi yang lebih besar seperti m5.large, bukan t3.small default.	
Perbarui template peluncuran.	Perbarui pengaturan peluncuran dan pilih subnet untuk server SQL Always On. Server cluster SQL Always On tersebar di berbagai tempat AWS Zona Ketersediaan untuk ketersediaan tinggi.	Insinyur migrasi, pimpinan Migrasi
Perbarui pengaturan peluncuran.	Berdasarkan ukuran dan persyaratan kinerja Anda, perbarui jenis instans dan input/output operasi per detik (IOPS) di pengaturan peluncuran. (Opsional) Pilih elastic network interface yang ada di pengaturan peluncuran.	Insinyur migrasi, pimpinan Migrasi

Tes cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi server sumber.	Di konsol AWS MGN, verifikasi bahwa status server sumber Siap untuk pengujian.	Administrator cloud, Insinyur migrasi
Luncurkan contoh pengujian.	1. Luncurkan instance pengujian dan verifikasi di EC2 konsol Amazon bahwa pemeriksaan otomatis lulus. 2. Pilih dan luncurkan contoh pengujian untuk server saksi. 3. Verifikasi bahwa Anda dapat masuk ke server melalui konsol AWS MGN. 4. Pilih server cluster SQL Always On dan luncurkan instance pengujian bersama-sama.	Administrator cloud, Insinyur migrasi
Uji konektivitas dan integritas database.	Uji konektivitas dan integritas database dari instance pengujian. Kemudian tandai server sumber sebagai Siap untuk cutover di konsol AWS MGN.	Administrator cloud, Insinyur migrasi

Tugas pra-migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji integritas database.	Ini membantu memastikan tidak ada masalah	Insinyur data, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	integritas database di sumber sebelum migrasi. Jalankan `DBCC CHECKDB` dan tentukan`WITH_PHYSICAL_ONLY` . Menjalankan pemeriksaan ini tanpa `WITH_PHYSICAL_ONLY` dapat menyebabkan masalah kinerja di sumbernya. Untuk menjaga integritas database, jalankan pemeriksaan penuh mingguan database. Perintah ini memeriksa integritas logis dan fisik database dengan mendeteksi potensi masalah korupsi. Pemeriksaan memverifikasi struktur database, seperti halaman, baris, indeks, dan tabel sistem.	
Uji koneksi ke server yang terhubung.	Uji koneksi di antara semua server yang ada, dan dokumentasikan statusnya . Ini membantu memastikan bahwa server tertaut berfungsi sebagaimana dimaksud setelah migrasi.	Insinyur data, DBA
Verifikasi cadangan.	Konfirmasikan integritas cadangan sumber.	Insinyur data, DBA

AWS Potongan MGN

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan SQL Server dan layanan cluster.	Hentikan layanan SQL Server dan Microsoft cluster pada semua node cluster SQL.	DBA, insinyur Migrasi
Verifikasi server.	Di konsol AWS MGN, verifikasi bahwa status server sumber Siap untuk dipotong dan status replikasi data Sehat.	Insinyur migrasi
Luncurkan cutover.	1. Luncurkan cutover AWS MGN untuk server saksi. 2. Luncurkan cutover AWS MGN untuk instance cluster SQL Always On. 3. Verifikasi bahwa status berubah menjadi Cutover yang sedang berlangsung. Untuk informasi selengkapnya, lihat Meluncurkan instance cutover dalam dokumentasi AWS MGN.	Insinyur migrasi
Uji server yang diluncurkan.	Masuk ke EC2 instans Amazon yang diluncurkan dan validasi kesehatan klaster. Verifikasi bahwa server berada di subnet yang benar, ukuran instans dan pengaturan IOPS sudah benar, dan server saksi dapat diakses.	DBA, insinyur Migrasi

Tugas pasca-cutover basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui alamat IP cluster.	Perbarui alamat IP cluster untuk cluster Windows menggunakan dua alamat IP cadangan di subnet target. Untuk informasi selengkap nya, lihat Mengubah Alamat IP Instance Failover Cluster.	DBA, insinyur Migrasi
Perbarui pendengar IPs grup ketersediaan Selalu Aktif.	1. Buka pengelola cluster failover. 2. Pilih peran grup ketersedi aan Selalu Aktif. 3. Perluas nama pendengar grup ketersediaan. 4. Dalam menu konteks (klik kanan), pilih properti alamat IP. 5. Perbarui alamat IP menggunakan alamat yang disediakan untuk pendengar di subnet target. 6. Connect ke instance utama SQL Server menggunak an SSMS, dan verifikas i bahwa listener Always On menggunakan kedua subnet. IPs	DBA, insinyur Migrasi
Verifikasi koneksi.	Menggunakan SSMS, sambungkan ke pendengar grup ketersediaan Selalu Aktif,	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan konfirmasikan bahwa koneksi berhasil.	
Periksa kesehatan grup ketersediaan Selalu Aktif.	1. Arahkan ke folder Grup ketersediaan, buka menu konteks (klik kanan), dan pilih Tampilkan dasbor. 2. Untuk semua replika, verifikasi bahwa status sinkronisasi disinkronkan.	DBA, insinyur Migrasi
Periksa log kesalahan.	Buka log kesalahan, dan verifikasi kesalahan yang dilaporkan untuk instance SQL Server. Pastikan pemulihan selesai untuk semua database.	DBA, insinyur Migrasi
Uji server yang ditautkan.	Uji konektivitas dari setiap server yang terhubung. Jika terjadi masalah konektivitas, pastikan server dan port target dapat diakses.	DBA, insinyur Migrasi

Selesaikan cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Selesaikan cutover.	Setelah Anda memvalidasi cluster SQL Always On target, selesaikan cutover melalui konsol MGN. AWS Ini menghentikan replikasi data dari server sumber dan membuang data dari server	Administrator cloud, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	replikasi. Ini juga menghapus server replikasi dan sumber daya yang terkait.	

Pemecahan masalah

Isu	Solusi
AWS Pemecahan masalah MGN	Untuk masalah dan resolusi umum, lihat bagian Pemecahan Masalah dan FAQ di dokumentasi MGN. AWS

Sumber daya terkait

AWS sumber daya

- Opsi-1 Rehost - () AWS Application Migration ServiceAWS
- Apa itu AWS Application Migration Service?

Sumber daya SQL Server

- Apa itu SQL Server Management Studio (SSMS)?

Informasi tambahan

Untuk persyaratan keamanan standar untuk memigrasikan beban kerja ke AWS Cloud, lihat Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan di situs web. AWS

Migrasikan SQL Server ke AWS menggunakan grup ketersediaan terdistribusi

Praveen Marthala, Amazon Web Services

Ringkasan

Grup ketersediaan Microsoft SQL Server Always On menyediakan solusi ketersediaan tinggi (HA) dan pemulihan bencana (DR) untuk SQL Server. Grup ketersediaan terdiri dari replika utama yang menerima read/write lalu lintas, dan hingga delapan replika sekunder yang menerima lalu lintas baca. Grup ketersediaan dikonfigurasi pada Windows Server Failover Cluster (WSFC) dengan dua atau lebih node.

Grup ketersediaan terdistribusi Microsoft SQL Server Always On menyediakan solusi untuk mengonfigurasi dua grup ketersediaan terpisah antara dua grup independen WFSCs. Grup ketersediaan yang merupakan bagian dari grup ketersediaan terdistribusi tidak harus berada di pusat data yang sama. Satu grup ketersediaan dapat berada di lokasi, dan grup ketersediaan lainnya dapat berada di Amazon Web Services (AWS) Cloud pada instans Amazon Elastic Compute Cloud (Amazon EC2) di domain yang berbeda.

Pola ini menguraikan langkah-langkah untuk menggunakan grup ketersediaan terdistribusi untuk memigrasikan database SQL Server lokal yang merupakan bagian dari grup ketersediaan yang ada ke SQL Server dengan grup ketersediaan yang disiapkan di Amazon. EC2 Dengan mengikuti pola ini, Anda dapat memigrasikan database ke AWS Cloud dengan waktu henti minimal selama cutover. Basis data sangat tersedia di AWS segera setelah cutover. Anda juga dapat menggunakan pola ini untuk mengubah sistem operasi yang mendasari dari lokal ke AWS sambil mempertahankan versi SQL Server yang sama.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Direct Connect atau AWS Site-to-Site VPN
- Versi SQL Server yang sama diinstal lokal dan pada dua node di AWS

Versi produk

- SQL Server versi 2016 dan yang lebih baru
- SQL Server Enterprise Edition

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server dengan grup ketersediaan Selalu Aktif di tempat

Tumpukan teknologi target

- Database Microsoft SQL Server dengan grup ketersediaan Selalu Aktif di Amazon EC2 di AWS Cloud

Arsitektur migrasi

Terminologi

- WSFC 1 - WSFC di tempat
- WSFC 2 - WSFC di AWS Cloud
- AG 1 - Grup ketersediaan pertama, yang ada di WSFC 1
- AG 2 - Grup ketersediaan kedua, yang ada di WSFC 2
- Replika primer SQL Server — Node di AG 1 yang dianggap sebagai primer global untuk semua penulisan
- SQL Server forwarder — Node di AG 2 yang menerima data secara asinkron dari replika utama SQL Server
- Replika sekunder SQL Server — Node di AG 1 atau AG 2 yang menerima data secara sinkron dari replika utama atau forwarder

Alat

- AWS Direct Connect — AWS Direct Connect menautkan jaringan internal Anda ke lokasi AWS Direct Connect melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke layanan AWS publik, melewati penyedia layanan internet di jalur jaringan Anda.

- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, dan Anda dapat meningkatkan skala atau meningkatkan skala.
- AWS Site-to-Site VPN — AWS Site-to-Site VPN mendukung pembuatan jaringan pribadi site-to-site virtual (VPN). Anda dapat mengonfigurasi VPN untuk meneruskan lalu lintas antara instans yang Anda luncurkan di AWS dan jaringan jarak jauh Anda sendiri.
- Microsoft SQL Server Management Studio — Microsoft SQL Server Management Studio (SSMS) adalah lingkungan terintegrasi untuk mengelola infrastruktur SQL Server. Ini menyediakan antarmuka pengguna dan sekelompok alat dengan editor skrip kaya yang berinteraksi dengan SQL Server.

Epik

Siapkan grup ketersediaan kedua di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat WSFC di AWS.	Buat WSFC 2 di EC2 instans Amazon dengan dua node untuk HA. Anda akan menggunakan klaster failover ini untuk membuat grup ketersediaan kedua (AG 2) di AWS.	Administrator sistem, SysOps administrator
Buat grup ketersediaan kedua di WSFC 2.	Menggunakan SSMS, buat AG 2 pada dua node di WSFC 2. Node pertama di WSFC 2 akan bertindak sebagai forwarder. Node kedua di WSFC 2 akan bertindak sebagai replika sekunder AG 2. Pada tahap ini, tidak ada database yang tersedia di AG	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Ini adalah titik awal untuk menyiapkan grup ketersediaan terdistribusi.	
Buat database tanpa opsi pemulihan di AG 2.	Cadangkan database pada grup ketersediaan lokal (AG 1). Kembalikan database ke forwarder dan replika sekunder AG 2 tanpa opsi pemulihan. Saat memulihkan database, tentukan lokasi dengan ruang disk yang cukup untuk file data database dan file log. Pada tahap ini, database berada dalam keadaan pemulihan. Mereka bukan bagian dari AG 2 atau grup ketersediaan terdistribusi, dan mereka tidak melakukan sinkronisasi.	DBA, Pengembang

Konfigurasikan grup ketersediaan terdistribusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup ketersediaan terdistribusi di AG 1.	Untuk membuat grup ketersediaan terdistribusi di AG 1, gunakan `DISTRIBUTED` opsi `CREATE AVAILABILITY GROUP` with the.	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Gunakan alamat `LISTENER_URL` titik akhir untuk AG 1 dan AG 2. 2. Untuk`AVAILABILITY-MODE`, gunakan `ASYNCHRONOUS_COMMIT` untuk menghindari latensi jaringan, jika ada. Ini tidak akan mempengaruhi kinerja database. 3. Untuk`FAILOVER_MODE` , gunakan`MANUAL`. Ini adalah satu-satunya mode ketersediaan yang bekerja dengan grup ketersediaan terdistribusi. 4. Untuk mengembalikan database secara manual pada AG 2 dan memiliki kontrol lebih besar pada database yang lebih besar, gunakan untuk`MANUAL`. `SEEDING_MODE`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup ketersediaan terdistribusi di AG 2.	Untuk membuat grup ketersediaan terdistribusi di AG 2, gunakan `ALTER AVAILABILITY GROUP` dengan `DISTRIBUTED` opsi. 1. Gunakan alamat `LISTENER_URL` titik akhir untuk AG 1 dan AG 2. 2. Untuk`AVAILABILITY-MODE`, gunakan `ASYNCHRONOUS_COMMIT` untuk menghindari latensi jaringan, jika ada. Ini tidak akan mempengaruhi kinerja database. 3. Untuk`FAILOVER_MODE` , gunakan`MANUAL`. Ini adalah satu-satunya mode ketersediaan yang bekerja dengan grup ketersediaan terdistribusi. 4. Untuk mengembalikan database secara manual pada AG 2 dan memiliki kontrol lebih besar pada database yang lebih besar, gunakan untuk`MANUAL`. `SEEDING_MODE` Grup ketersediaan terdistribusi dibuat antara AG 1 dan AG 2.	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Database di AG 2 belum dikonfigurasi untuk mengambil bagian dalam aliran data dari AG 1 ke AG 2.	
Tambahkan database ke forwarder dan replika sekunder pada AG 2.	Tambahkan database ke grup ketersediaan terdistribusi `ALTER DATABASE` dengan menggunakan `SET HADR AVAILABILITY GROUP` opsi di forwarder dan replika sekunder pada AG 2. Ini memulai aliran data asinkron antara database pada AG 1 dan AG 2. Primer global mengambil penulisan, mengirimkan data secara sinkron ke replika sekunder pada AG 1, dan mengirimkan data secara asinkron ke forwarder di AG 2. Forwarder pada AG 2 mengirimkan data secara sinkron ke replika sekunder pada AG 2.	DBA, Pengembang

Memantau aliran data asinkron antara AG 1 dan AG 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan DMVs dan log SQL Server.	Pantau status aliran data antara dua grup ketersedi aan dengan menggunakan	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tampilan manajemen dinamis (DMVs) dan log SQL Server. DMVs yang menarik untuk pemantauan termasuk `sys.dm_hadr_availability_replica_states` dan `sys.dm_hadr_automatic_seeding`. Untuk status sinkronisasi forwarder, pantau status yang disinkronkan di log SQL Server pada forwarder.	

Lakukan aktivitas cutover untuk migrasi akhir

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan semua lalu lintas ke replika utama.	Hentikan lalu lintas masuk ke replika utama di AG 1 sehingga tidak ada aktivitas tulis yang terjadi pada database dan database siap untuk migrasi.	Pemilik aplikasi, Pengembang
Ubah mode ketersediaan grup ketersediaan terdistribusi di AG 1.	Pada replika utama, atur mode ketersediaan grup ketersediaan terdistribusi menjadi sinkron. Setelah Anda mengubah mode ketersediaan menjadi sinkron, data dikirim secara	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sinkron dari replika utama di AG 1 ke forwarder di AG 2.	
Periksa LSNs di kedua grup ketersediaan.	Periksa Nomor Urutan Log terakhir (LSNs) di AG 1 dan AG 2. Karena tidak ada penulisan yang terjadi di replika utama di AG 1, data disinkronkan, dan terakhir LSNs untuk kedua grup ketersediaan harus cocok.	DBA, Pengembang
Perbarui AG 1 ke peran sekunder.	Saat Anda memperbarui AG 1 ke peran sekunder, AG 1 kehilangan peran replika utama dan tidak menerima penulisan, dan aliran data antara dua grup ketersediaan berhenti.	DBA, Pengembang

Gagal ke grup ketersediaan kedua

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gagal secara manual ke AG 2.	Pada forwarder di AG 2, ubah grup ketersediaan terdistri busi untuk memungkinkan kehilangan data. Karena Anda sudah mengecek dan mengonfirmasi bahwa yang terakhir LSNs pada pertandin gan AG 1 dan AG 2, kehilanga n data tidak menjadi perhatian.	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Saat Anda mengizinkan kehilangan data pada forwarder di AG 2, peran AG 1 dan AG 2 berubah: • AG 2 menjadi grup ketersediaan dengan replika primer dan replika sekunder. • AG 1 menjadi grup ketersediaan dengan forwarder dan replika sekunder.	
Ubah mode ketersediaan grup ketersediaan terdistribusi di AG 2.	Pada replika utama di AG 2, ubah mode ketersediaan menjadi asinkron. Ini mengubah pergerakan data dari AG 2 ke AG 1, dari sinkron ke asinkron. Langkah ini diperlukan untuk menghindari latensi jaringan antara AG 2 dan AG 1, jika ada, dan tidak akan memengaruhi kinerja database.	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai mengirim lalu lintas ke replika utama yang baru.	Perbarui string koneksi untuk menggunakan titik akhir URL pendengar di AG 2 untuk mengirim lalu lintas ke database. AG 2 sekarang menerima penulisan dan pengiriman data ke forwarder di AG 1, bersama dengan mengirim data ke replika sekundernya sendiri di AG 2. Data bergerak secara asinkron dari AG 2 ke AG 1.	Pemilik aplikasi, Pengembang

Lakukan kegiatan pasca-cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jatuhkan grup ketersediaan terdistribusi di AG 2.	Pantau migrasi untuk jumlah waktu yang direncanakan. Kemudian jatuhkan grup ketersediaan terdistribusi pada AG 2 untuk menghapus pengaturan grup ketersedi aan terdistribusi antara AG 2 dan AG 1. Ini menghapus konfigurasi grup ketersediaan terdistribusi, dan aliran data dari AG 2 ke AG 1 berhenti. Pada titik ini, AG 2 sangat tersedia di AWS, dengan replika utama yang mengambil penulisan dan replika	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sekunder dalam grup ketersediaan yang sama.	
Menonaktifkan server lokal.	Menonaktifkan server lokal di WSFC 1 yang merupakan bagian dari AG 1.	Administrator sistem, SysOps administrator

Sumber daya terkait

- Grup ketersediaan terdistribusi
- SQL Docs: Grup ketersediaan terdistribusi
- SQL Docs: Grup ketersediaan Selalu Aktif: solusi ketersediaan tinggi dan pemulihan bencana

Migrasi database relasional ke MongoDB Atlas di AWS

Battulga Purevragchaa dan Igor Alekseev, Amazon Web Services

Babu Srinivasan, MongoDB

Ringkasan

Pola ini menjelaskan langkah-langkah untuk bermigrasi dari database relasional seperti SQL Server, MySQL, atau PostgreSQL ke MongoDB Atlas di. AWS Cloud Ini menggunakan MongoDB Relational Migrator untuk membantu mempercepat migrasi data dari database relasional ke MongoDB Atlas.

Pola ini menyertai panduan Migrasi ke MongoDB Atlas di situs web Prescriptive Guidance. AWS AWS Ini memberikan langkah-langkah implementasi untuk salah satu skenario migrasi yang dibahas dalam panduan itu. Untuk skenario migrasi tambahan, lihat pola berikut di situs web Panduan AWS Preskriptif:

- Migrasikan lingkungan MongoDB yang dihosting sendiri ke MongoDB Atlas di AWS
- Streaming data dari IBM Db2, SAP, Sybase, dan database lainnya ke MongoDB Atlas di AWS

Pola ini ditujukan untuk Mitra dan AWS pengguna AWS System Integrator (SI).

Prasyarat dan batasan

Prasyarat

- Database relasional sumber (Oracle Database, SQL Server, PostgreSQL, MySQL SAP/Sybase , ASE, dan lain-lain) untuk bermigrasi ke MongoDB Atlas.
- Keakraban dengan database relasional, MongoDB Atlas, dan. Layanan AWS Pola ini menjelaskan beberapa langkah migrasi pada tingkat tinggi. Rincian tambahan akan ditambahkan di versi future.

Versi produk

- MongoDB versi 5.0 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan migrasi dari database sistem manajemen database relasional (RDBMS) ke MongoDB Atlas pada. AWS

Untuk arsitektur referensi MongoDB Atlas yang mendukung skenario penggunaan yang berbeda, lihat Migrasi ke MongoDB Atlas di situs web Panduan Preskriptif. AWS AWS

Alat

- MongoDB Atlas adalah database yang dikelola sepenuhnya sebagai layanan DBaa (S) untuk menyebarkan dan mengelola database MongoDB di cloud.
- MongoDB Relational Migrator menyediakan transisi data yang mulus dari database relasional tradisional ke MongoDB. Ini membantu mengotomatiskan proses konversi dan mengubah model data terstruktur dari database relasional menjadi format dokumen fleksibel yang disediakan oleh MongoDB. Migrator Relasional mempertahankan integritas data dan hubungan untuk menyederhanakan migrasi. Organizations dapat memanfaatkan manfaat skalabilitas, kinerja, dan fleksibilitas yang diberikan MongoDB sambil mempertahankan keakraban data mereka yang ada.

Praktik terbaik

Untuk praktik terbaik untuk menggunakan MongoDB, lihat posting AWS di Blog Jaringan Mitra AWS .

Epik

Penemuan dan penilaian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan parameter dan ukuran database relasional.	Perkirakan ukuran set kerja dengan menggunakan rekomendasi Relational Migrator dan informasi dari `db.stats()` untuk total ruang indeks. Asumsikan bahwa persentase ruang data Anda akan sering diakses. Tugas ini harus memakan waktu sekitar satu minggu. Untuk informasi lebih lanjut dan contoh untuk ini dan cerita	Pemilik aplikasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lainnya dalam epik ini, lihat bagian Sumber daya terkait.	
Perkirakan kebutuhan bandwidth jaringan.	Untuk memperkirakan kebutuhan bandwidth jaringan Anda, kalikan ukuran dokumen rata-rata dengan jumlah dokumen yang disajikan per detik. Pertimbangkan lalu lintas maksimum yang akan ditanggung oleh setiap node di cluster Anda sebagai dasarnya. Untuk menghitung kecepatan transfer data hilir dari klaster Anda ke aplikasi klien, gunakan jumlah total dokumen yang dikembali kan selama periode waktu tertentu. Jika aplikasi Anda membaca dari node sekunder, bagilah jumlah total dokumen ini dengan jumlah node yang dapat melayani operasi baca. Untuk menemukan ukuran dokumen rata-rata untuk database, gunakan `db.stats().avgObjS ize` perintah. Tugas ini biasanya akan memakan waktu satu hari.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih tingkat Atlas.	Ikuti petunjuk dalam dokumentasi MongoDB untuk memilih tingkat cluster Atlas yang benar.	DBA
Rencanakan untuk cutover.	Rencanakan pemotongan aplikasi.	DBA, Pemilik aplikasi

Siapkan lingkungan MongoDB Atlas baru di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster MongoDB Atlas baru di. AWS	Di MongoDB Atlas, pilih Build a Cluster. Di kotak dialog Create New Cluster, pilih AWS sebagai penyedia cloud.	DBA
Pilih Wilayah AWS dan konfigurasi cluster global.	Pilih dari daftar yang tersedia Wilayah AWS untuk cluster Atlas Anda. Konfigurasikan cluster global jika diperlukan.	DBA
Pilih tingkat cluster.	Pilih tingkat klaster pilihan Anda. Pemilihan tingkat Anda menentukan faktor-faktor seperti memori, penyimpanan, dan spesifikasi IOPS.	DBA
Konfigurasikan pengaturan cluster tambahan.	Konfigurasikan pengaturan cluster tambahan seperti versi MongoDB, cadangan, dan opsi enkripsi. Untuk informasi selengkapnya tentang opsi ini, lihat bagian Sumber daya terkait.	DBA

Konfigurasikan keamanan dan kepatuhan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan daftar akses.	Untuk terhubung ke cluster Atlas, Anda harus menambahkan entri ke daftar akses proyek. Atlas menggunakan TLS/SSL untuk mengenkripsi koneksi ke virtual private cloud (VPC) untuk database Anda. Untuk menyiapkan daftar akses untuk proyek dan untuk informasi selengkapnya tentang cerita dalam epik ini, lihat bagian Sumber daya terkait.	DBA
Otentikasi dan otorisasi pengguna.	Anda harus membuat dan mengotentikasi pengguna database yang akan mengakses cluster MongoDB Atlas. Untuk mengakses cluster dalam proyek, pengguna harus menjadi bagian dari proyek itu, dan mereka dapat menjadi bagian dari beberapa proyek.	DBA
Buat peran khusus.	(Opsional) Atlas mendukung pembuatan peran khusus jika izin pengguna database Atlas bawaan tidak mencakup kumpulan izin yang Anda inginkan.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan pengintip VPC.	(Opsional) Atlas mendukung VPC mengintip dengan yang lain. VPCs AWS	Administrator AWS
Siapkan AWS PrivateLink titik akhir.	(Opsional) Anda dapat mengatur titik akhir pribadi AWS dengan menggunak an AWS PrivateLink. Untuk informasi selengkapnya, lihat dokumentasi Amazon VPC.	Administrator AWS
Aktifkan otentikasi dua faktor.	(Opsional) Atlas mendukung otentikasi dua faktor (2FA) untuk membantu pengguna mengontrol akses ke akun Atlas mereka.	Administrator AWS
Siapkan otentikasi dan otorisasi pengguna dengan LDAP.	(Opsional) Atlas mendukung melakukan otentikasi dan otorisasi pengguna dengan Lightweight Directory Access Protocol (LDAP).	DBA
Siapkan AWS akses terpadu.	(Opsional) Beberapa fitur Atlas, termasuk Atlas Data Lake dan enkripsi saat istirahat menggunakan manajemen kunci pelanggan , menggunakan peran AWS Identity and Access Management (IAM) untuk otentikasi.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan enkripsi saat istirahat menggunakan AWS KMS.	(Opsional) Atlas mendukung penggunaan AWS Key Management Service (AWS KMS) untuk mengenkripsi mesin penyimpanan dan cadangan penyedia cloud.	Administrator AWS
Siapkan enkripsi tingkat bidang sisi klien.	(Opsional) Atlas mendukung enkripsi tingkat bidang sisi klien, termasuk enkripsi otomatis bidang.	Administrator AWS

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan MongoDB Relational Migrator ke daftar akses.	Tambahkan Relational Migrator ke daftar akses untuk database sumber Anda. Ini membantu mempersiapkan lingkungan sumber untuk terhubung ke cluster Atlas target.	DBA
Menilai objek database relasional.	Luncurkan MongoDB Relational Migrator dan sambungkan ke database relasional Anda. Mulai penilaian.	DBA
Terima pola migrasi atau pilih untuk mengubahnya berdasarkan kebutuhan bisnis Anda.	Baik menerima pola database yang direkomendasikan oleh Migrator Relasional berdasarkan penilaian awal dan parameter kinerja, atau	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memilih untuk mengubahnya berdasarkan kebutuhan bisnis Anda.	
Luncurkan replika target Anda yang ditetapkan di MongoDB Atlas.	Luncurkan replika target Anda yang ditetapkan di MongoDB Atlas. Di Relational Migrator, pilih I'm ready to migrate.	DBA

Konfigurasikan integrasi operasional

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke cluster MongoDB Atlas.	Pastikan konektivitas cluster MongoDB Atlas berfungsi seperti yang diharapkan.	Pemilik aplikasi
Berinteraksi dengan data cluster.	Verifikasi data cluster.	DBA
Pantau cluster Anda.	Verifikasi bahwa cluster Anda sudah diatur dengan benar.	DBA
Cadangkan dan pulihkan data cluster.	Jadwalkan pencadangan dengan irama reguler untuk data cluster.	DBA

Sumber daya terkait

Semua link berikut, kecuali dinyatakan lain, pergi ke halaman web dalam dokumentasi MongoDB.

Panduan migrasi

- Bermigrasi ke MongoDB AWS Atlas on (Panduan Preskriptif)AWS

Penemuan dan penilaian

- Memori
- Contoh ukuran dengan kumpulan data sampel Atlas
- Contoh ukuran untuk aplikasi seluler
- Lalu Lintas Jaringan
- Penskalaan Otomatis Cluster
- Template ukuran Atlas

Mengkonfigurasi keamanan dan kepatuhan

- Konfigurasikan Entri Daftar Akses IP
- Konfigurasikan Pengguna Database
- Konfigurasikan Akses ke UI Atlas
- Konfigurasikan Peran Database Kustom
- Konfigurasikan Pengguna Database
- Siapkan Koneksi Peering Jaringan
- Pelajari Tentang Titik Akhir Pribadi di Atlas
- Kelola Opsi Otentikasi Multi-Faktor Anda
- Mengatur Otentikasi dan Otorisasi Pengguna dengan LDAP
- Danau Atlas Data
- Enkripsi saat Istirahat menggunakan Manajemen Kunci Pelanggan
- Metode untuk mengambil peran (dokumentasi IAM)
- Enkripsi Tingkat Bidang Sisi Klien
- Enkripsi Otomatis
- Kontrol Keamanan MongoDB Atlas
- Pusat Kepercayaan MongoDB
- Konfigurasikan Fitur Keamanan untuk Cluster

Menyiapkan lingkungan MongoDB Atlas baru di AWS

- Penyedia dan Wilayah Cloud
- Kelola Global Cluster
- Pilih Tingkat Cluster

- Konfigurasikan Pengaturan Tambahan
- Memulai dengan Atlas
- Konfigurasikan Akses ke UI Atlas

Migrasi data

- Migrasi atau Impor Data

Cluster pemantauan

- Pantau Cluster Anda

Mengintegrasikan operasi

- Connect ke Cluster
- Berinteraksi dengan Data Anda
- Pantau Cluster Anda
- Cadangkan, Pulihkan, dan Arsipkan Data

Migrasikan lingkungan MongoDB yang dihosting sendiri ke MongoDB Atlas di AWS

Battulga Purevragchaa dan Igor Alekseev, Amazon Web Services

Babu Srinivasan, MongoDB

Ringkasan

Pola ini menjelaskan langkah-langkah untuk bermigrasi dari lingkungan MongoDB yang dikelola sendiri (termasuk MongoDB Community Server, Enterprise Server, Enterprise Advanced, mLab, atau cluster MongoDB yang dikelola) ke MongoDB Atlas di. AWS Cloud Ini menggunakan Layanan Migrasi Langsung Atlas untuk membantu mempercepat migrasi data dari MongoDB ke MongoDB Atlas.

Pola ini menyertai panduan Migrasi ke MongoDB Atlas di situs web Prescriptive Guidance. AWS AWS Ini memberikan langkah-langkah implementasi untuk salah satu skenario migrasi yang dibahas dalam panduan itu. Untuk skenario migrasi tambahan, lihat pola berikut di situs web Panduan AWS Preskriptif:

- Migrasi database relasional ke MongoDB Atlas di AWS
- Streaming data dari IBM Db2, SAP, Sybase, dan database lainnya ke MongoDB Atlas AWS

Pola ini ditujukan untuk Mitra dan AWS pengguna AWS Systems Integrator (SI).

Prasyarat dan batasan

Prasyarat

- Sumber MongoDB Enterprise Advanced, Server Komunitas, atau lingkungan MongoDB yang dikelola sendiri lainnya untuk bermigrasi ke MongoDB Atlas.
- Keakraban dengan MongoDB, MongoDB Atlas, dan. Layanan AWS Pola ini menjelaskan beberapa langkah migrasi pada tingkat tinggi. Rincian tambahan akan ditambahkan di versi future.

Versi produk

- MongoDB versi 6.0.13 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan Atlas Live Migration Service yang digunakan untuk memigrasikan data dari database MongoDB Enterprise Advanced dan database Komunitas MongoDB ke MongoDB Atlas pada. AWS Gunakan layanan ini saat Anda perlu memigrasikan database yang besar dan kompleks ke MongoDB Atlas dengan waktu henti minimal dan sinkronisasi data berkelanjutan. Pola ini menggunakan Atlas Live Migration Service.

Diagram berikut menunjukkan MongoDB mirror service `mongomirror` () yang juga dapat Anda gunakan untuk memigrasikan data dari database MongoDB Enterprise Advanced dan database Komunitas MongoDB ke MongoDB Atlas melalui koneksi aman. AWS AWS PrivateLink Gunakan `mongomirror` untuk replikasi data yang sedang berlangsung antara MongoDB lokal dan MongoDB Atlas. Alat ini sangat ideal untuk pemulihan bencana atau migrasi bertahap tetapi berada di luar cakupan pola ini.

Untuk arsitektur referensi MongoDB Atlas lainnya yang mendukung skenario penggunaan yang berbeda, lihat Migrasi ke MongoDB Atlas di situs web Panduan Preskriptif. AWS AWS

Alat

- MongoDB Atlas adalah database yang dikelola sepenuhnya sebagai layanan (dBaaS) untuk menyebarkan dan mengelola database MongoDB di cloud.
- Atlas Live Migration Service adalah utilitas MongoDB gratis yang membantu memigrasikan database ke Atlas. Layanan ini membuat database sumber tetap sinkron dengan database tujuan hingga cutover. Ketika Anda siap untuk memotong, Anda menghentikan instance aplikasi Anda, mengarahkannya ke cluster Atlas tujuan, dan memulai ulang. Untuk mengakses layanan ini, pilih opsi Database dari cluster MongoDB Atlas Anda.
- mongomirror adalah alat untuk memigrasikan data secara manual dari replika MongoDB yang ada yang disetel ke set replika MongoDB Atlas. `mongomirror`tidak mengharuskan Anda untuk mematikan set replika atau aplikasi yang ada, tidak mengimpor data pengguna atau peran, atau menyalin database konfigurasi. Anda dapat mengunduh `mongomirror` dari dokumentasi MongoDB.

Praktik terbaik

Untuk praktik terbaik untuk menggunakan MongoDB, lihat posting AWS di Blog Jaringan Mitra AWS .

Epik

Penemuan dan penilaian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan ukuran cluster.	Perkirakan ukuran set kerja dengan menggunakan informasi dari `db.stats( )` untuk total ruang indeks. Asumsikan bahwa persentase ruang data Anda akan sering diakses. Atau, Anda dapat memperkirakan kebutuhan memori Anda berdasarkan asumsi Anda sendiri. Tugas ini harus memakan waktu sekitar satu minggu. Untuk informasi lebih lanjut dan contoh untuk ini dan cerita lainnya dalam epik ini, lihat bagian Sumber daya terkait.	DBA, Pemilik aplikasi
Perkirakan kebutuhan bandwidth jaringan.	Untuk memperkirakan kebutuhan bandwidth jaringan Anda, kalikan ukuran dokumen rata-rata dengan jumlah dokumen yang disajikan per detik. Pertimbangkan lalu lintas maksimum yang akan ditanggung oleh node mana pun di cluster Anda sebagai dasarnya. Untuk menghitung kecepatan transfer data hilir dari cluster Anda ke aplikasi klien, gunakan jumlah total dokumen	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang dikembalikan selama periode waktu tertentu. Jika aplikasi Anda membaca dari node sekunder, bagilah jumlah total dokumen ini dengan jumlah node yang dapat melayani operasi baca. Untuk menemukan ukuran dokumen rata-rata untuk database, gunakan `db.stats().avgObjSize` perintah. Tugas ini biasanya akan memakan waktu satu hari.	
Pilih tingkat Atlas.	Ikuti petunjuk dalam dokumentasi MongoDB untuk memilih tingkat cluster Atlas yang benar.	DBA
Rencanakan untuk cutover.	Rencanakan pemotongan aplikasi.	DBA, Pemilik aplikasi

Siapkan lingkungan MongoDB Atlas baru di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster MongoDB Atlas baru di. AWS	Masuk ke Atlas dan buka halaman Ikhtisar untuk proyek Anda. Pilih tombol Create untuk membuat cluster. Untuk informasi selengkapnya tentang penggunaan, lihat dokumentasi MongoDB .	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih Wilayah AWS dan konfigurasi cluster global.	Pilih dari daftar yang tersedia Wilayah AWS untuk cluster Atlas Anda. Konfigurasikan cluster global jika diperluka n. Untuk informasi selengkap nya tentang penggunaan, lihat dokumentasi MongoDB .	DBA
Pilih tingkat cluster.	Pilih tingkat klaster pilihan Anda. Pemilihan tingkat Anda menentukan faktor-faktor seperti memori, penyimpanan, dan spesifikasi IOPS.	DBA
Konfigurasikan pengaturan cluster tambahan.	Konfigurasikan pengaturan cluster tambahan seperti versi MongoDB, cadangan, dan opsi enkripsi. Untuk informasi selengkapnya tentang opsi ini, lihat bagian Sumber daya terkait.	DBA

Konfigurasikan keamanan dan kepatuhan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Otentikasi dan otorisasi pengguna.	Anda harus membuat dan mengotentikasi pengguna database yang akan mengakses cluster MongoDB Atlas. Untuk mengakses cluster dalam proyek, pengguna harus menjadi bagian dari proyek itu, dan mereka dapat menjadi bagian	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dari beberapa proyek. Atlas juga mendukung otentikasi berdasarkan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya tentang penggunaan, lihat dokumentasi MongoDB .	
Buat peran khusus.	(Opsional) Atlas mendukung pembuatan peran khusus jika izin pengguna database Atlas bawaan tidak mencakup kumpulan izin yang Anda inginkan.	DBA
Siapkan pengintip VPC.	(Opsional) Atlas mendukung virtual private cloud (VPC) mengintip dengan yang lain. VPCs AWS	Administrator AWS
Siapkan AWS PrivateLink titik akhir.	(Opsional) Anda dapat mengatur titik akhir pribadi AWS dengan menggunak an AWS PrivateLink. Untuk informasi selengkapnya, lihat dokumentasi Amazon VPC.	Administrator AWS
Aktifkan otentikasi dua faktor.	(Opsional) Atlas mendukung otentikasi dua faktor (2FA) untuk membantu pengguna mengontrol akses ke akun Atlas mereka.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan otentikasi dan otorisasi pengguna dengan LDAP.	(Opsional) Atlas mendukung melakukan otentikasi dan otorisasi pengguna dengan Lightweight Directory Access Protocol (LDAP).	Administrator AWS
Siapkan AWS akses terpadu.	(Opsional) Beberapa fitur Atlas, termasuk Atlas Data Lake dan enkripsi saat istirahat menggunakan manajemen kunci pelanggan , menggunakan peran IAM untuk otentikasi.	Administrator AWS
Siapkan enkripsi saat istirahat menggunakan AWS KMS.	(Opsional) Atlas mendukung penggunaan AWS Key Management Service (AWS KMS) untuk mengenkripsi mesin penyimpanan dan cadangan penyedia cloud.	Administrator AWS
Siapkan enkripsi tingkat bidang sisi klien.	(Opsional) Atlas mendukung enkripsi tingkat bidang sisi klien, termasuk enkripsi otomatis bidang.	Administrator AWS

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih replika target Anda yang ditetapkan di MongoDB Atlas.	Arahkan ke cluster Atlas tujuan dan pilih tombol elipsis (...). Pada daftar cluster, tombol ini muncul di bawah nama cluster.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Dalam detail cluster, tombol muncul di sebelah kanan, di sebelah tombol Connect and Configuration. Untuk informasi selengkapnya tentang penggunaan, lihat dokumentasi MongoDB .	
Tambahkan Layanan Migrasi Langsung Atlas ke daftar akses.	Tambahkan Layanan Migrasi Atlas Live ke daftar akses di kluster AWS sumber Anda. Ini membantu mempersia pkan lingkungan sumber untuk terhubung ke cluster Atlas target.	DBA
Lakukan migrasi dengan Atlas Live Migration Service.	Pilih Mulai migrasi. Saat tombol Siapkan untuk Cutover berubah menjadi hijau, lakukan cutover. Tinjau metrik kinerja klaster Atlas. Pertimbangkan untuk memperbarui koneksi database di semua lapisan aplikasi untuk menunjuk ke database baru.	DBA

Konfigurasikan integrasi operasional

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke cluster MongoDB Atlas.	Pastikan konektivitas cluster MongoDB Atlas berfungsi seperti yang diharapkan.	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berinteraksi dengan data cluster.	Uji data cluster.	DBA
Pantau cluster Anda.	Verifikasi bahwa cluster Anda sudah diatur dengan benar.	DBA
Cadangkan dan pulihkan data cluster.	Jadwalkan pencadangan dengan irama reguler untuk data cluster.	DBA

Pemecahan masalah

Isu	Solusi

Kesalahan: Tidak dapat mencapai sumber yang ditentukan	• Pastikan Anda menambahkan rentang subnet yang benar ke daftar akses IP di cluster sumber. Anda dapat menemukan empat rentang subnet yang diperlukan di jendela modal Migrasi Langsung.
	• Konfirmasikan bahwa nama host yang Anda tentukan diselesaikan ke alamat IP publik. Pada prompt perintah, gunakan salah satu perintah berikut:
	`nslookup <hostname>`
	`ping <hostname>`
	• Pastikan Anda tidak menggunakan koneksi peering VPC, yang tidak kompatibel dengan migrasi tarik langsung. Jika koneksi peering VPC adalah satu-satunya pilihan Anda, gunakan sebagai gantinya. `mongomirror`

Kesalahan: Tidak dapat menyelesaikan nama host	Tidak ada alamat IP yang ditemukan untuk nama host yang diberikan. Konfirmasikan

Isu	Solusi
	bahwa nama host yang diberikan benar dan dapat diakses publik.
Kesalahan lainnya	Jika Anda menemukan kesalahan lain, lihat Memecahkan Masalah Migrasi Langsung (Tarik) di dokumentasi MongoDB.

Sumber daya terkait

Semua link berikut, kecuali dinyatakan lain, pergi ke halaman web dalam dokumentasi MongoDB.

Panduan migrasi

- Bermigrasi ke MongoDB AWS Atlas on (Panduan Preskriptif)AWS

Migrasi warisan

- Migrasi versi lama MongoDB

Penemuan dan penilaian

- Memori
- Contoh ukuran dengan kumpulan data sampel Atlas
- Contoh ukuran untuk aplikasi seluler
- Lalu Lintas Jaringan
- Penskalaan Otomatis Cluster
- Template ukuran Atlas

Mengkonfigurasi keamanan dan kepatuhan

- Konfigurasikan Entri Daftar Akses IP
- Konfigurasikan Pengguna Database
- Konfigurasikan Akses ke UI Atlas
- Konfigurasikan Peran Database Kustom

- Konfigurasikan Pengguna Database
- Siapkan Koneksi Peering Jaringan
- Pelajari Tentang Titik Akhir Pribadi di Atlas
- Kelola Opsi Otentikasi Multi-Faktor Anda
- Mengatur Otentikasi dan Otorisasi Pengguna dengan LDAP
- Danau Atlas Data
- Enkripsi saat Istirahat menggunakan Manajemen Kunci Pelanggan
- Metode untuk mengambil peran (dokumentasi IAM)
- Enkripsi Tingkat Bidang Sisi Klien
- Enkripsi Otomatis
- Kontrol Keamanan MongoDB Atlas
- Pusat Kepercayaan MongoDB
- Konfigurasikan Fitur Keamanan untuk Cluster

Menyiapkan lingkungan MongoDB Atlas baru AWS

- Penyedia dan Wilayah Cloud
- Kelola Global Cluster
- Pilih Tingkat Cluster
- Konfigurasikan Pengaturan Tambahan
- Memulai dengan Atlas
- Konfigurasikan Akses ke UI Atlas

Migrasi data

- Migrasi atau Impor Data

Cluster pemantauan

- Pantau Cluster Anda

Mengintegrasikan operasi

- Connect ke Cluster
- Berinteraksi dengan Data Anda
- Pantau Cluster Anda
- Cadangkan, Pulihkan, dan Arsipkan Data

Pelatihan

- Migrasi Langsung dengan MongoDB Atlas

Informasi tambahan

Untuk informasi tambahan, lihat topik berikut dalam dokumentasi MongoDB:

- Untuk memindahkan data ke instance tanpa server, gunakan Kompas untuk mengekspor dan mengimpor data, atau memigrasikan data dengan alat yang dikelola sendiri. Untuk mempelajari selengkapnya, lihat Batasan Instans Tanpa Server.
- Untuk memuat data ke dalam klaster baru di Atlas, lihat Memuat Data ke Atlas.
- Untuk membuat salinan klaster Anda untuk tujuan pengujian, lihat Metode Pencadangan untuk Penerapan yang Dikelola Sendiri.
- Jika aplikasi yang ingin Anda migrasi memerlukan uptime yang hampir berkesinambungan, hubungi MongoDB Support dan bagikan persyaratan uptime dan konfigurasi cluster Anda.
- Untuk informasi tambahan, lihat Memigrasi atau Mengimpor Data.

Migrasikan database Oracle ke Amazon DynamoDB menggunakan AWS DMS

Rambabu Karnena, Amazon Web Services

Ringkasan

Pola ini memandu Anda melalui langkah-langkah untuk memigrasikan database Oracle ke Amazon DynamoDB menggunakan AWS Database Migration Service (AWS DMS). Ini mencakup tiga jenis database sumber:

- Database Oracle lokal
- Database Oracle di Amazon Elastic Compute Cloud (Amazon) EC2
- Amazon Relational Database Service (Amazon RDS) untuk instans Oracle DB

Dalam bukti konsep ini, pola ini berfokus pada migrasi dari Amazon RDS for Oracle DB instance.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Aplikasi yang terhubung ke database Amazon RDS for Oracle
- Tabel yang dibuat di sumber Amazon RDS for Oracle database dengan kunci utama dan data sampel

Batasan

- Objek database Oracle, seperti prosedur, fungsi, paket, dan pemicu, tidak dipertimbangkan untuk migrasi karena Amazon DynamoDB tidak mendukung objek database ini.

Versi produk

- Pola ini berlaku untuk semua edisi dan versi database Oracle yang didukung oleh AWS DMS. Untuk informasi selengkapnya, lihat menggunakan database Oracle sebagai sumber AWS DMS dan menggunakan database Amazon DynamoDB sebagai target AWS DMS. Kami menyarankan

Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif.

Arsitektur

Tumpukan teknologi sumber

- Amazon RDS for Oracle DB instans, Oracle di EC2 Amazon, atau database Oracle lokal

Tumpukan teknologi target

- Amazon DynamoDB

Arsitektur migrasi data AWS

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan dapat diskalakan.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud. Pola ini menggunakan Amazon RDS for Oracle.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Di akun AWS Anda, buat virtual private cloud (VPC) dan subnet pribadi.	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup keamanan dan daftar kontrol akses jaringan.	Untuk informasi selengkapnya, lihat dokumentasi AWS.	Administrator sistem
Konfigurasikan dan mulai instans Amazon RDS for Oracle DB.	Untuk informasi selengkapnya, lihat dokumentasi AWS.	DBA, Administrator sistem

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk mengakses DynamoDB.	Di konsol AWS Identity and Access Management (IAM), buat peran, lampirkan kebijakan`AmazonDyn amoDBFullAccess to it`, dan pilih AWS DMS sebagai layanan.	Administrator sistem
Buat instance replikasi AWS DMS untuk migrasi.	Instance replikasi harus berada di Availability Zone dan VPC yang sama dengan database sumber.	Administrator sistem
Buat titik akhir sumber dan target di AWS DMS.	Untuk membuat titik akhir basis data sumber, Anda memiliki dua opsi: • Di konsol Amazon RDS, pilih Databases, DB identifie r, Connectivity & Security, dan pilih endpoint. • Pada konsol AWS DMS, pilih Pilih instans RDS DB.	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk membuat titik akhir database target, pilih peran Amazon Resource Name (ARN) dari tugas sebelumnya untuk mengakses DynamoDB.	
Buat tugas AWS DMS untuk memuat tabel database Oracle sumber ke DynamoDB.	Pilih nama titik akhir sumber dan tujuan serta contoh replikasi dari langkah sebelumnya. Jenisnya bisa full load. Pilih skema Oracle dan tentukan% untuk memilih semua tabel.	Administrator sistem
Validasi tabel di DynamoDB.	Untuk melihat hasil migrasi, pilih Tabel dari panel navigasi kiri di konsol DynamoDB.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah kode aplikasi.	Untuk menyambung ke dan mengambil data dari DynamoDB, perbarui kode aplikasi.	Pemilik aplikasi, DBA, administrator Sistem

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ganti klien aplikasi untuk menggunakan DynamoDB.		DBA, Pemilik aplikasi, Administrator sistem

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS.	Misalnya, mematikan Amazon RDS untuk instans Oracle, DynamoDB, dan instans replikasi AWS DMS.	DBA, Administrator sistem
Kumpulkan metrik.	Metrik mencakup waktu untuk bermigrasi, persentase pekerjaan manual dan pekerjaan yang dilakukan oleh alat, dan penghematan biaya.	DBA, Pemilik aplikasi, Administrator sistem

Sumber daya terkait

- AWS Database Migration Service dan Amazon DynamoDB: Apa yang Perlu Anda Ketahui (posting blog)
- Menggunakan Database Oracle sebagai sumber AWS DMS
- Menggunakan database Amazon DynamoDB sebagai target AWS Database Migration Service
- Praktik Terbaik untuk Migrasi dari RDBMS ke Amazon DynamoDB (whitepaper)

Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Oracle dan AWS DMS SharePlex

Ramu Jagini, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Oracle 8i atau 9i lokal ke Amazon Relational Database Service (Amazon RDS) untuk database Oracle. Anda dapat menggunakan pola ini untuk menyelesaikan migrasi dengan waktu henti yang dikurangi dengan menggunakan Quest SharePlex untuk replikasi sinkron.

Anda harus menggunakan instans database Oracle perantara untuk migrasi Anda karena AWS Database Migration Service (AWS DMS) tidak mendukung Oracle 8i atau 9i sebagai lingkungan sumber. Anda dapat menggunakan SharePlex 7.6.3 untuk mereplikasi dari versi database Oracle sebelumnya ke versi database Oracle yang lebih baru. Instans database Oracle perantara kompatibel sebagai target untuk SharePlex 7.6.3 dan didukung sebagai sumber untuk AWS DMS atau rilis yang lebih baru. SharePlex Dukungan ini memungkinkan replikasi data selanjutnya ke lingkungan target Amazon RDS for Oracle.

Pertimbangkan bahwa beberapa tipe dan fitur data yang tidak digunakan lagi dapat memengaruhi migrasi dari Oracle 8i atau 9i ke Oracle Database versi terbaru. Untuk mengurangi dampak ini, pola ini menggunakan Oracle 11.2.0.4 sebagai versi database perantara untuk membantu mengoptimalkan kode skema sebelum bermigrasi ke lingkungan target Amazon RDS for Oracle.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sumber database Oracle 8i atau 9i di lingkungan lokal
- Oracle Database 12c Rilis 2 (12CR2) untuk pementasan di Amazon Elastic Compute Cloud (Amazon) EC2
- Quest SharePlex 7.6.3 (kelas komersial)

Batasan

- RDS untuk keterbatasan Oracle

Versi produk

- Oracle 8i atau 9i untuk database sumber
- Oracle 12 CR2 untuk database pementasan (harus cocok dengan Amazon RDS for Oracle versi)
- Oracle 12 CR2 atau yang lebih baru untuk database target (Amazon RDS for Oracle)

Arsitektur

Tumpukan teknologi sumber

- Database Oracle 8i atau 9i
- SharePlex

Tumpukan teknologi target

- Amazon RDS for Oracle

Arsitektur migrasi

Diagram berikut menunjukkan cara memigrasikan database Oracle 8i atau 9i dari lingkungan lokal ke instans Amazon RDS for Oracle DB di AWS Cloud.

Diagram menunjukkan alur kerja berikut:

1. Aktifkan database sumber Oracle dengan mode log arsip, pencatatan paksa, dan pencatatan tambahan.
2. Kembalikan database pementasan Oracle dari database sumber Oracle dengan menggunakan pemulihan Recovery Manager (RMAN) point-in-time dan FLASHBACK_SCN.
3. Konfigurasikan SharePlex untuk membaca redo log dari database sumber Oracle dengan menggunakan `FLASHBACK_SCN` (digunakan dalam RMAN).
4. Mulai SharePlex replikasi untuk menyinkronkan data dari database sumber Oracle ke database pementasan Oracle.
5. Kembalikan database target Amazon RDS for Oracle dengan menggunakan EXPDP dan IMPDP dengan. `FLASHBACK_SCN`

6. Konfigurasikan AWS DMS dan tugas sumbernya sebagai database pementasan Oracle dan Amazon RDS for Oracle sebagai database target `FLASHBACK_SCN` dengan menggunakan (digunakan dalam EXPDP).
7. Mulai tugas AWS DMS untuk menyinkronkan data dari database pementasan Oracle ke database target Oracle.

Alat

- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Quest SharePlex adalah alat replikasi Oracle-to-Oracle data untuk memindahkan data dengan downtime minimal dan tidak ada kehilangan data.
- Recovery Manager (RMAN) adalah klien Oracle Database yang melakukan tugas pencadangan dan pemulihan pada database Anda. Ini sangat menyederhanakan pencadangan, pemulihan, dan pemulihan file database.
- Ekspor Pompa Data membantu Anda mengunggah data dan metadata ke dalam satu set file sistem operasi yang disebut kumpulan file dump. Kumpulan file dump hanya dapat diimpor oleh utilitas Impor Pompa Data atau paket DBMS_DATAPUMP.

Epik

Siapkan SharePlex dan database pementasan Oracle di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah EC2 instance.	1. Buat sebuah EC2 instance. 2. Instal Oracle 12 CR2 pada EC2 instance untuk berfungsi sebagai database pementasan Oracle.	Administrasi Oracle
Siapkan database pementasan.	Siapkan database pementasan Oracle untuk dipulihkan sebagai upgrade pada	Administrasi Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Oracle 12 CR2 dengan mengambil cadangan RMAN dari lingkungan sumber database Oracle 8i atau 9i. Untuk informasi selengkap nya, lihat Panduan Pengguna Oracle 9i Recovery Manager dan Panduan Pengguna Backup dan Pemulihan Database dalam dokumentasi Oracle.	
Konfigurasikan SharePlex.	Konfigurasikan SharePlex sumber sebagai database Oracle 8i atau 9i lokal, dan konfigurasikan target sebagai database CR2 pementasan Oracle 12 yang dihosting di Amazon. EC2	SharePlex, Administrasi Oracle

Siapkan Amazon RDS for Oracle sebagai lingkungan target Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance Oracle DB.	Buat database Amazon RDS for Oracle, lalu hubungkan CR2 Oracle 12 ke database. Untuk informasi selengkapnya, lihat Membuat instans Oracle DB dan menghubungkan ke database pada instans Oracle DB dalam dokumentasi Amazon RDS.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan Amazon RDS for Oracle dari database pementasan.	1. Ambil cadangan EXPDP dari server database pementasan Oracle dengan menggunakan. `FLASHBACK_SCN` 2. Kembalikan Amazon RDS for Oracle dari database pementasan. Untuk informasi selengkapnya, lihat 54 DBMS_DATAPUMP di dokumentasi Oracle.	DBA

Mengatur AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir untuk database.	Buat titik akhir sumber untuk database pementasan Oracle dan titik akhir target untuk database Amazon RDS for Oracle. Untuk informasi selengkap nya, lihat Bagaimana cara membuat titik akhir sumber atau target menggunak an AWS DMS? di Pusat Pengetahuan AWS.	DBA
Buat contoh replikasi.	Gunakan AWS DMS untuk meluncurkan instance replikasi untuk database pementasan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Oracle ke database Amazon RDS for Oracle. Untuk informasi selengkap nya, lihat Bagaimana cara membuat instans replikasi AWS DMS? di Pusat Pengetahuan AWS.	
Buat dan mulai tugas replikasi.	Buat tugas replikasi AWS DMS untuk pengambilan data perubahan (CDC) dengan menggunakan `FLASHBACK _SCN` dari EXPDP (karena beban penuh sudah terjadi melalui EXPDP). Untuk informasi selengkapnya, lihat Membuat tugas dalam dokumentasi AWS DMS.	DBA

Dipotong ke Amazon RDS for Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan beban kerja aplikasi.	Hentikan server aplikasi dan aplikasinya selama jendela cutover yang direncanakan.	Pengembang aplikasi, DBA
Validasi sinkronisasi database pementasan Oracle lokal dengan instance. EC2	Konfirmasikan bahwa semua pesan telah diposting untuk tugas replikasi dari instance SharePlex replikasi ke database pementasan Oracle di Amazon EC2 dengan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	melakukan beberapa sakelar log pada database sumber lokal. Untuk informasi selengkapnya, lihat 6.4.2 Mengganti File Log di dokumentasi Oracle.	
Validasi sinkronisasi database pementasan Oracle dengan database Amazon RDS for Oracle.	Konfirmasikan bahwa semua tugas AWS DMS Anda tidak memiliki lag dan tidak ada kesalahan, lalu periksa status validasi tugas.	DBA
Hentikan replikasi SharePlex dan Amazon RDS.	Jika replikasi DMS SharePlex dan AWS tidak menunjukkan kesalahan apa pun, hentikan kedua replikasi.	DBA
Memetakan ulang aplikasi ke Amazon RDS.	Bagikan detail endpoint Amazon RDS for Oracle dengan server aplikasi dan aplikasinya, lalu mulai aplikasi untuk melanjutkan operasi bisnis.	Pengembang aplikasi, DBA

Uji lingkungan target AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji lingkungan database pementasan Oracle di AWS.	1. Uji SharePlex replikasi dan verifikasi bahwa tidak ada celah sinkronisasi atau kesalahan replikasi pada	SharePlex, Administrasi Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	database pementasan Oracle. 2. Verifikasi bahwa aplikasi berperilaku seperti yang diharapkan melalui tolok ukur yang ditentukan di lingkungan lokal.	
Uji lingkungan Amazon RDS.	1. Verifikasi bahwa semua data yang disebarkan ke Amazon RDS setelah replikasi bebas dari kesalahan. 2. Arahkan aplikasi lain ke instans Amazon RDS DB, lalu jalankan pengujian kinerja untuk memverifikasi perilaku yang diharapkan. Untuk informasi selengkapnya, lihat Amazon RDS for Oracle di dokumentasi Amazon RDS.	Administrasi Oracle

Sumber daya terkait

- Migrasi dengan percaya diri
- Amazon EC2
- Amazon RDS for Oracle
- AWS Database Migration Service
- Debugging Migrasi AWS DMS Anda: Apa yang Harus Dilakukan Ketika Ada yang Salah (Bagian 1)
- Debugging Migrasi AWS DMS Anda: Apa yang Harus Dilakukan Ketika Ada yang Salah (Bagian 2)
- Debugging Migrasi AWS DMS Anda: Apa yang Harus Dilakukan Ketika Ada yang Salah? (Bagian 3)

- SharePlex untuk Replikasi Database
- SharePlex: replikasi database untuk lingkungan apa pun

Memigrasikan database MySQL lokal ke Amazon EC2

Lorenzo Mota, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan database MySQL lokal ke database MySQL di instans Amazon Elastic Compute Cloud (Amazon EC2). Pola ini membahas penggunaan AWS Database Migration Service (AWS DMS) atau alat MySQL asli seperti mysqldump untuk migrasi. Ini berfokus pada migrasi database penuh ke instance MySQL DB.

Pola ini terutama untuk DBAs dan solusi arsitek. Ini dapat digunakan dalam proyek kecil atau besar, dalam pengujian atau fase migrasi akhir. Kami menyarankan Anda menjalankan setidaknya satu siklus pengujian sebelum Anda menggunakan pola ini di lingkungan produksi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber MySQL di pusat data lokal

Versi produk

- MySQL versi 5.5 dan yang lebih baru
- Sistem operasi target yang didukung oleh Amazon EC2; lihat Amazon EC2 FAQs

Arsitektur

Tumpukan teknologi sumber

- Database MySQL lokal

Tumpukan teknologi target

- Instans database MySQL di Amazon EC2

Metode migrasi data AWS

- AWS DMS
- Alat MySQL asli seperti mysqldump, atau alat pihak ketiga seperti Percona XtraBackup

Arsitektur target

Diagram berikut menggambarkan implementasi target Amazon EC2 setelah cutover.

Arsitektur migrasi data AWS

Menggunakan AWS DMS:

Diagram berikut menggambarkan alur kerja migrasi data berdasarkan AWS DMS pengiriman perubahan penuh dan bertahap ke database MySQL target hingga cutover. Koneksi jaringan dari tempat ke AWS tergantung pada persyaratan klien SQL dan berada di luar cakupan pola ini.

Menggunakan alat MySQL lainnya:

Diagram berikut menggambarkan alur kerja migrasi data berdasarkan penggunaan alat MySQL untuk menghasilkan file dump ekspor dari database lokal. File-file ini dipindahkan ke Amazon Simple Storage Service (Amazon S3) dan diimpor ke database MySQL target sebelum cutover. Koneksi jaringan dari tempat ke AWS tergantung pada persyaratan klien SQL dan berada di luar cakupan pola ini.

Catatan:

- Bergantung pada pertimbangan down time dan ukuran database untuk cutover akhir, Anda dapat menggunakan AWS DMS atau alat change data capture (CDC) lainnya untuk meminimalkan waktu cutover. Bila Anda menggunakan alat CDC seperti AWS DMS, Anda dapat bermigrasi ke database target dalam hitungan menit.
- Strategi offline dengan mysqldump bisa cukup jika ukuran database dan latensi jaringan memungkinkan jendela migrasi cutover pendek. (Kami menyarankan Anda melakukan pengujian untuk mendapatkan perkiraan waktu.)
- Biasanya strategi CDC melalui AWS DMS membutuhkan lebih banyak pemantauan dan kompleksitas daripada opsi offline.

Alat

AWS layanan

- AWS Database Migration Service (AWS DMS) mendukung beberapa basis data sumber dan target. Untuk informasi tentang sumber MySQL dan basis data target yang didukung AWS DMS oleh, lihat Menggunakan database yang kompatibel dengan MySQL sebagai sumber AWS DMS untuk dan Menggunakan database yang kompatibel dengan MySQL sebagai target. AWS DMS Jika database sumber Anda tidak didukung oleh AWS DMS, Anda harus memilih metode lain untuk memigrasi data Anda.

Alat-alat lainnya

- mysqldump adalah utilitas MySQL yang membuat file dump dari database MySQL untuk tujuan pencadangan atau migrasi.
- Percona XtraBackup adalah utilitas open source untuk melakukan backup non-blocking pada database MySQL.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi database.	Validasi versi database sumber dan target. Untuk informasi tentang versi MySQL yang didukung AWS DMS oleh, lihat Sumber AWS DMS untuk dan Target AWS DMS untuk dalam dokumentasi. AWS DMS	DBA
Identifikasi sistem operasi target.	Tentukan versi sistem operasi target. Untuk daftar sistem operasi target yang didukung	DBA, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	oleh Amazon EC2, lihat Amazon EC2. FAQs	
Identifikasi persyaratan perangkat keras.	Tentukan persyaratan perangkat keras untuk instance server target berdasarkan daftar kompatibi litas MySQL dan persyaratan kapasitas.	DBA, Administrator sistem
Identifikasi persyaratan penyimpanan.	Tentukan jenis penyimpanan dan kapasitas untuk database target.	DBA, Administrator sistem
Identifikasi persyaratan jaringan.	Tentukan persyaratan jaringan seperti latensi dan bandwidth.	DBA, Administrator sistem
Pilih jenis instance target.	Pilih jenis instans target berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.	DBA, Administrator sistem
Identifikasi persyaratan keamanan.	Tentukan persyaratan keamanan akses jaringan atau host untuk basis data sumber dan target.	DBA, Administrator sistem
Identifikasi pengguna.	Tentukan daftar pengguna sistem operasi untuk instalasi perangkat lunak MySQL. Untuk informasi selengkapnya, lihat dokumentasi MySQL.	DBA, Administrator sistem
Tentukan strategi cadangan.		DBA
Tentukan persyaratan ketersediaan.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi migrasi aplikasi atau strategi peralihan.		DBA, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat virtual private cloud (VPC) dan subnet.	Konfigurasikan tabel rute, gateway internet, gateway NAT, dan subnet. Untuk informasi selengkapnya, lihat opsi konfigurasi VPC di dokumentasi Amazon VPC.	Administrator sistem
Buat grup keamanan dan daftar kontrol akses jaringan (ACLs).	Konfigurasikan port (default untuk MySQL adalah 3306) dan rentang CIDR atau spesifik tergantung pada kebutuhan Anda. IPs	Administrator sistem
Konfigurasikan dan mulai instance EC2.	Untuk petunjuknya, lihat Meluncurkan instans EC2 di dokumentasi Amazon EC2.	Administrator sistem

Instal perangkat lunak MySQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna dan grup.	Buat pengguna dan grup sistem operasi yang membutuhkan akses ke server dan database. Untuk informasi selengkapnya, lihat Kontrol	DBA, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Akses dan Manajemen Akun di dokumentasi MySQL.	
Unduh MySQL.	Unduh perangkat lunak MySQL. Untuk petunjuk dan binari, lihat Menginstal MySQL di dokumentasi MySQL.	DBA, Administrator sistem
Instal MySQL pada instans EC2 dan konfigurasikan server.	Connect ke instans EC2 Anda dan instal perangkat lunak MySQL. Untuk informasi selengkapnya, lihat Connect ke instans EC2 Anda di dokumentasi Amazon EC2.	DBA, Administrator sistem

Migrasi data - opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan MySQL asli atau alat pihak ketiga.	Opsi ini menggunakan alat MySQL asli atau alat pihak ketiga untuk memigrasikan objek dan data database. Untuk petunjuk, lihat dokumentasi untuk mysqldump atau Percona XtraBackup (untuk migrasi fisik). Untuk informasi selengkapnya tentang penggunaan alat ini, lihat opsi migrasi posting AWS blog untuk MySQL ke Amazon RDS for MySQL atau Amazon Aurora MySQL.	DBA

Migrasikan data - opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan AWS DMS.	Untuk informasi selengkapnya, lihat Tampilan tingkat tinggi AWS DMS dalam AWS DMS dokumentasi.	DBA

Bersiaplah untuk cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan jumlah objek.	Kumpulkan jumlah objek dari database sumber dan database target baru. Perbaiki perbedaan apa pun dalam database target.	DBA
Periksa dependensi.	Konfirmasikan bahwa dependensi (tautan) ke dan dari database lain masih valid dan berfungsi dengan benar.	DBA
Uji.	Jika ini adalah siklus pengujian, lakukan pengujian kueri, kumpulkan metrik, dan perbaiki masalah apa pun.	DBA

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pindahkan klien.	Alihkan klien aplikasi ke infrastruktur baru.	DBA, Pemilik aplikasi, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan dukungan.	Memberikan dukungan selama pengujian aplikasi fungsional.	DBA

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya.	Matikan contoh AWS DMS replikasi dan AWS sumber daya sementara lainnya.	DBA, Administrator sistem
Tinjau dan dokumen proyek.	Tinjau dan validasi dokumen proyek.	DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik.	Kumpulkan metrik seperti waktu untuk bermigras i, persentase perubahan manual dibandingkan dengan perubahan bantuan alat, dan penghematan biaya.	DBA, Pemilik aplikasi, Administrator sistem
Tutup proyek.	Tutup proyek migrasi dan berikan umpan balik.	DBA, Pemilik aplikasi, Administrator sistem
Database sumber penonakti fan.	Menonaktifkan database MySQL lokal.	DBA, Administrator sistem

Sumber daya terkait

Referensi

- Dokumentasi Amazon EC2
- AWS DMS dokumentasi
- Harga Amazon EC2

- AWS DMS Step-by-Step Penelusuran
- mysqldump
- Percona XtraBackup

Tutorial dan video

- Memulai dengan AWS DMS
- Pengantar Amazon EC2 - Elastic Cloud Server & Hosting dengan AWS(video)

Pantau Amazon Aurora untuk instance tanpa enkripsi

Mansi Suratwala, Amazon Web Services

Ringkasan

Pola ini menyediakan CloudFormation template Amazon Web Services (AWS) yang dapat Anda terapkan untuk mengatur notifikasi otomatis saat instans Amazon Aurora dibuat tanpa enkripsi diaktifkan.

Aurora adalah mesin basis data relasional yang dikelola sepenuhnya dan kompatibel dengan MySQL dan PostgreSQL. Dengan sejumlah beban kerja, Aurora dapat memberikan hingga lima kali throughput MySQL dan hingga tiga kali throughput PostgreSQL tanpa memerlukan perubahan pada sebagian besar aplikasi Anda saat ini.

CloudFormation Template membuat CloudWatch acara Amazon Events dan fungsi AWS Lambda. Acara ini menggunakan AWS CloudTrail untuk memantau pembuatan instans Aurora apa pun atau pemulihan titik waktu dari instans yang ada. Acara Cloudwatch Events memulai fungsi Lambda, yang memeriksa apakah enkripsi diaktifkan. Jika enkripsi tidak diaktifkan, fungsi Lambda mengirimkan notifikasi Amazon Simple Notification Service (Amazon SNS) yang memberi tahu Anda tentang pelanggaran tersebut.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif

Batasan

- Kontrol layanan ini hanya berfungsi dengan instans Amazon Aurora. Ini tidak mendukung instans Amazon Relational Database Service (Amazon RDS) lainnya.
- CloudFormation Template harus digunakan untuk `CreateDBInstance` dan `RestoreDBClusterToPointInTim` e saja.

Versi produk

- Versi PostgreSQL yang didukung di Amazon Aurora

- Versi MySQL yang didukung di Amazon Aurora

Arsitektur

Tumpukan teknologi target

- Amazon Aurora
- AWS CloudTrail
- Amazon CloudWatch
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon SNS

Arsitektur target

Otomatisasi dan skala

Anda dapat menggunakan CloudFormation template beberapa kali untuk Wilayah dan akun yang berbeda. Anda hanya perlu menjalankannya sekali di setiap Wilayah atau akun.

Alat

Alat

- Amazon Aurora - Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang kompatibel dengan MySQL dan PostgreSQL.
- AWS CloudTrail — AWS CloudTrail membantu Anda mengelola tata kelola, kepatuhan, serta audit operasional dan risiko akun AWS Anda. Tindakan yang diambil oleh pengguna, peran, atau layanan AWS dicatat sebagai peristiwa di CloudTrail.
- CloudWatch Acara Amazon — CloudWatch Acara Amazon menghadirkan near-real-time aliran peristiwa sistem yang menjelaskan perubahan sumber daya AWS.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.

- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel yang dapat Anda gunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan danau data.
- Amazon SNS - Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan menggunakan Lambda, HTTP, email, notifikasi push seluler, dan pesan teks seluler (SMS).

Kode

File.zip proyek tersedia sebagai lampiran.

Epik

Buat bucket S3 untuk skrip Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan bucket S3.	Buka konsol Amazon S3, dan pilih atau buat bucket S3. Bucket S3 ini akan meng-host file kode Lambda .zip. Bucket S3 Anda harus berada di Wilayah yang sama dengan Aurora. Nama bucket S3 tidak dapat berisi garis miring di depan.	Arsitek awan

Unggah kode Lambda ke bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode Lambda.	Unggah file kode Lambda .zip yang disediakan di bagian Lampiran ke bucket S3 yang Anda tentukan.	Arsitek awan

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	Di CloudFormation konsol, terapkan `RDS_Aurora_Encryption_At_Rest.yml` CloudFormation template yang disediakan sebagai lampiran ke pola ini. Dalam epik berikutnya, berikan nilai untuk parameter template.	Arsitek awan

Lengkapi parameter dalam CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan nama bucket S3.	Masukkan nama bucket S3 yang Anda buat atau pilih di epik pertama.	Arsitek awan
Berikan kunci S3.	Berikan lokasi file kode Lambda .zip di bucket S3 Anda, tanpa garis miring di depan (misalnya,). `<directory>/<file-name>.zip`	Arsitek awan
Berikan alamat email.	Berikan alamat email aktif untuk menerima notifikasi Amazon SNS.	Arsitek awan
Tentukan tingkat logging.	Tentukan tingkat logging dan frekuensi untuk fungsi Lambda Anda. `Info`menunjuk pesan informasi rinci	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tentang kemajuan aplikasi. `Error`menunjuk peristiwa kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. `Warning`menunjuk situasi yang berpotensi berbahaya.	

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Ketika template berhasil digunakan, ia mengirimkan pesan email berlangganan ke alamat email yang disediaka n. Untuk menerima pemberita huan, Anda harus mengonfir masi langganan email ini.	Arsitek awan

Sumber daya terkait

- Membuat bucket S3
- Mengunggah file ke bucket S3
- Membuat klaster DB Amazon Aurora
- Membuat aturan CloudWatch Acara yang dipicu pada panggilan AWS API menggunakan AWS CloudTrail

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Memantau GoldenGate log Oracle dengan menggunakan Amazon CloudWatch

Chithra Krishnamurthy, Amazon Web Services

Ringkasan

Oracle GoldenGate menyediakan replikasi real-time antara Amazon Relational Database Service (Amazon RDS) untuk database Oracle, atau antara database Oracle yang dihosting di Amazon Elastic Compute Cloud (Amazon). EC2 Ini mendukung replikasi searah dan dua arah.

Ketika Anda menggunakan GoldenGate untuk replikasi, pemantauan sangat penting untuk memverifikasi bahwa GoldenGate proses aktif dan berjalan, untuk memastikan bahwa sumber dan database target sinkron.

Pola ini menjelaskan langkah-langkah untuk menerapkan CloudWatch pemantauan Amazon untuk log GoldenGate kesalahan, dan cara menyetel alarm untuk mengirim pemberitahuan untuk peristiwa tertentu seperti `STOP` atau `ABEND` agar Anda dapat mengambil tindakan yang sesuai untuk melanjutkan replikasi dengan cepat.

Prasyarat dan batasan

Prasyarat

- GoldenGate diinstal dan dikonfigurasi pada EC2 instance, sehingga Anda dapat mengatur CloudWatch pemantauan pada EC2 instance tersebut. Jika Anda ingin memantau GoldenGate seluruh Wilayah AWS untuk replikasi dua arah, Anda harus menginstal CloudWatch agen di setiap EC2 instance tempat GoldenGate proses berjalan.

Keterbatasan

- Pola ini menjelaskan cara memantau GoldenGate proses dengan menggunakan CloudWatch. CloudWatch tidak memantau kelambatan replikasi atau masalah sinkronisasi data selama replikasi. Anda harus menjalankan kueri SQL terpisah untuk memantau kelambatan replikasi atau kesalahan terkait data, seperti yang dijelaskan dalam dokumentasi. GoldenGate

Versi produk

- Dokumen ini didasarkan pada implementasi Oracle GoldenGate 19.1.0.0.4 untuk Oracle di Linux x86-64. Namun, solusi ini berlaku untuk semua versi utama GoldenGate.

Arsitektur

Tumpukan teknologi target

- GoldenGate binari untuk Oracle diinstal pada sebuah instance EC2
- Amazon CloudWatch
- Amazon Simple Notification Service (Amazon SNS)

Arsitektur target

Alat

Layanan AWS

- Amazon CloudWatch adalah layanan pemantauan yang digunakan dalam pola ini untuk memantau log GoldenGate kesalahan.
- Amazon SNS adalah layanan notifikasi pesan yang digunakan dalam pola ini untuk mengirim notifikasi email.

Alat-alat lainnya

- Oracle GoldenGate adalah alat replikasi data yang dapat Anda gunakan untuk Amazon RDS for Oracle database atau Oracle database yang di-host di Amazon. EC2

Langkah-langkah implementasi tingkat tinggi

1. Buat peran AWS Identity and Access Management (IAM) untuk CloudWatch agen.
2. Lampirkan peran IAM ke EC2 instance di mana log GoldenGate kesalahan dihasilkan.
3. Instal CloudWatch agen pada EC2 instance.
4. Konfigurasikan file konfigurasi CloudWatch agen: `awscli.conf` dan `awslogs.conf`.
5. Mulai CloudWatch agen.
6. Buat filter metrik di grup log.

7. Siapkan Amazon SNS.
8. Buat alarm untuk filter metrik. Amazon SNS mengirimkan peringatan email saat filter tersebut menangkap peristiwa.

Untuk petunjuk terperinci, lihat bagian selanjutnya.

Epik

Langkah 1. Buat peran IAM untuk agen CloudWatch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM.	Akses ke sumber daya AWS memerlukan izin, sehingga Anda membuat peran IAM untuk menyertakan izin yang diperlukan untuk setiap server untuk menjalankan agen. CloudWatch Untuk membuat peran IAM: 1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/. 2. Di panel navigasi, pilih Peran, lalu pilih Buat peran. 3. Untuk jenis entitas Tepercaya, pilih layanan AWS. 4. Untuk kasus penggunaan umum, pilih EC2, lalu pilih Berikutnya. 5. Dalam daftar kebijakan , pilih kotak centang	AWS umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di sebelah CloudWatchAgentServerPolicy. Jika perlu, gunakan kotak pencarian untuk menemukan kebijakan. 6. Pilih Berikutnya. 7. Untuk Nama peran, masukkan nama untuk peran baru Anda, seperti `goldengate-cw-monitoring-role` atau nama lain yang Anda inginkan. 8. (Opsional) Untuk Deskripsi peran, masukkan deskripsi. 9. Konfirmasikan yang CloudWatchAgentServerPolicymuncul di bawah Nama kebijakan. 10. (Opsional) Tambahkan satu atau beberapa tag (pasangan nilai kunci) untuk mengatur, melacak, atau mengontrol akses untuk peran ini, lalu pilih Buat peran.	

Langkah 2. Lampirkan peran IAM ke instance GoldenGate EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan peran IAM ke EC2 instance di mana log	Log kesalahan yang dihasilkan oleh GoldenGate harus diisi	AWS umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
GoldenGate kesalahan dihasilkan.	CloudWatch dan dipantau, jadi Anda perlu melampirkan peran IAM yang Anda buat di langkah 1 ke EC2 instance yang sedang berjalan. GoldenGate Untuk melampirkan peran IAM ke sebuah instance: 1. Buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/. 2. Di panel navigasi, pilih Instances, dan kemudian temukan instance di mana GoldenGate sedang berjalan. 3. Pilih instance, lalu pilih Actions, Security, Modify peran IAM. 4. Pilih peran IAM yang dibuat pada langkah pertama untuk dilampirkan ke instance Anda, lalu pilih Simpan.	

Langkah 3-5. Instal dan konfigurasikan CloudWatch agen pada instance Goldengate EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal CloudWatch agen pada GoldenGate EC2 instance.	Untuk menginstal agen, jalankan perintah:	AWS umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`sudo yum install -y awslogs`	

Edit file konfigurasi agen.	1. Jalankan perintah berikut.
	`sudo su -`
	2. Edit file ini untuk memperbarui Wilayah AWS seperlunya.
	`cat /etc/awslogs/conf`
	`[plugins]`
	`cwlogs = cwlogs`
	`[default]`
	`region = us-east-1`
	3. Edit `/etc/awslogs/awslogs.conf` file untuk memperbarui nama file, nama grup log, dan date/time format. Anda harus menentukan date/time untuk mencocokkan format tanggal di `ggserror.log` ; jika tidak, aliran log tidak akan mengalir ke CloudWatch. Contoh:
	`datetime_format = %Y-%m-%dT%H:%M:%S%z`
	`file = /u03/oracle/oragg/ggserr.log`
	`log_group_name = goldengate_monitor`
		AWS umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai CloudWatch agen.	Untuk memulai agen, gunakan perintah berikut. `$ sudo service awslogsd start` Setelah Anda memulai agen, Anda dapat melihat grup log di CloudWatch konsol. Aliran log akan memiliki isi file.	AWS umum

Langkah 6. Buat filter metrik untuk grup log

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat filter metrik untuk kata kunci ABEND dan STOPPED.	Saat Anda membuat filter metrik untuk grup log, setiap kali filter diidentifikasi dalam log kesalahan, filter akan memulai alarm dan mengirimkan pemberitahuan email berdasarkan konfigurasi Amazon SNS. Untuk membuat filter metrik: 1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/. 2. Pilih nama grup log. 3. Pilih Tindakan, lalu pilih Buat filter metrik. 4. Untuk pola Filter, tentukan pola sepertiABEND.	CloudWatch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Pilih Berikutnya, lalu masukkan nama untuk filter metrik Anda. 6. Di bawah Detail metrik, untuk namespace Metrik, masukkan nama untuk CloudWatch namespace tempat metrik akan dipublikasikan. Jika namespace belum ada, pastikan Create new dipilih. 7. Untuk nilai Metrik1, masukkan, karena filter metrik Anda menghitun g kemunculan kata kunci dalam filter. 8. Setel Unit ke Tidak Ada. 9. Pilih Create metric filter (Buat filter metrik). Anda dapat menemukan filter metrik yang Anda buat dari panel navigasi. 10. Buat filter metrik lain untuk `STOPPED` pola tersebut. Dalam satu grup log, Anda dapat membuat beberapa filter metrik dan mengatur alarm satu per satu.	

Langkah 7. Siapkan Amazon SNS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat sebuah topik SNS.	Pada langkah ini, Anda mengonfigurasi Amazon SNS untuk membuat alarm untuk filter metrik. Untuk membuat topik SNS: 1. Masuk ke konsol Amazon SNS di https://console.aws.amazon.com/sns/ rumah. 2. Di kotak Buat topik, masukkan nama topik seperti`goldengate-alert`, lalu pilih Langkah berikutnya. 3. Untuk Tipe, pilih Standar. 4. Gulir ke akhir formulir dan pilih Create topic (Buat topik). Konsol membuka halaman Detail topik baru.	Amazon SNS
Buat langganan.	Untuk membuat langganan ke topik: 1. Di panel navigasi kiri, pilih Subscriptions (Langganan). 2. Di halaman Subscript ions (Langganan), pilih Create subscription (Buat langganan). 3. Pada halaman Buat langganan, pilih bidang Topik ARN untuk melihat	Amazon SNS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	daftar topik di akun AWS Anda. 4. Pilih topik yang Anda buat di langkah sebelumnya. 5. Untuk Protocol (Protokol), pilih Email. 6. Untuk Titik Akhir, masukkan alamat email yang dapat Anda gunakan untuk menerima pemberitahuan. 7. Pilih Buat langganan. Konsol membuka halaman Detail langganan baru. 8. Periksa kotak masuk email Anda untuk pesan dari Pemberitahuan AWS, lalu pilih Konfirmasi langganan di email. Amazon SNS membuka browser web Anda dan menampilkan konfirmasi berlangganan dengan ID langganan Anda.	

Langkah 8. Buat alarm untuk mengirim pemberitahuan untuk filter metrik

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat alarm untuk topik SNS.	Untuk membuat alarm berdasarkan filter metrik grup log:	CloudWatch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/. 2. Dari panel navigasi, silakan pilih Log, kemudian pilih Grup log. 3. Pilih grup log yang menyertakan filter metrik Anda. 4. Pilih Filter metrik. 5. Di tab Filter metrik, pilih kotak centang untuk filter metrik yang ingin Anda gunakan sebagai dasar alarm. 6. Pilih Buat alarm. 7. Untuk Kondisi, tentukan hal berikut di setiap bagian: • Untuk Jenis ambang batas, pilih Statis. • Untuk Kapanpun. <metric-name> , pilih Lebih Besar. • Untuk dari. , tentukan 0. 8. Pilih Berikutnya. 9. Di bawah Pemberitahuan: • Untuk Pemicu status alarm, pilih Dalam alarm. • Untuk Kirim pemberita huan ke topik SNS berikut, pilih Pilih topik yang ada.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Di kotak email, pilih topik Amazon SNS yang Anda buat di langkah sebelumnya. 10. Pilih Berikutnya. 11. Untuk Nama dan deskripsi, Anda harus memasukkan nama dan deskripsi untuk alarm Anda. **Note** Untuk deskripsi, Anda dapat menentukan nama instance sehingga email notifikasi bersifat deskriptif. 12. Untuk Pratinjau dan buat, periksa apakah konfigurasi Anda sudah benar, lalu pilih Buat alarm. Setelah langkah-langkah ini, setiap kali pola-pola ini terdeteksi dalam file log GoldenGate kesalahan (`ggserr.log`) yang Anda pantau, Anda akan mendapatkan pemberitahuan email.	

Pemecahan masalah

Isu	Solusi
Aliran log dari log GoldenGate kesalahan tidak mengalir ke CloudWatch.	Periksa `/etc/awslogs/awslogs.conf` file untuk memverifikasi nama file, nama grup log, dan date/time formatnya. Anda harus menentukan date/time untuk mencocokkan format tanggal diggserror.log . Jika tidak, aliran log tidak akan mengalir ke CloudWatch.

Sumber daya terkait

- CloudWatch Dokumentasi Amazon
- Mengumpulkan metrik dan log dengan agen CloudWatch
- Dokumentasi Amazon SNS

Replatform Oracle Database Enterprise Edition ke Edisi Standar 2 di Amazon RDS for Oracle

Lanre (Lan-Ray) Showunmi dan Tarun Chawla, Amazon Web Services

Ringkasan

Oracle Database Enterprise Edition (EE) adalah pilihan populer untuk menjalankan aplikasi di banyak perusahaan. Dalam beberapa kasus, bagaimanapun, aplikasi menggunakan sedikit atau tidak ada fitur Oracle Database EE, sehingga ada kurangnya pembenaran untuk menimbulkan biaya lisensi yang besar. Anda dapat mencapai penghematan biaya dengan menurunkan database tersebut ke Oracle Database Standard Edition 2 (SE2) saat Anda bermigrasi ke Amazon RDS.

Pola ini menjelaskan cara menurunkan versi dari Oracle Database EE ke Oracle Database SE2 saat bermigrasi dari lokal ke Amazon RDS for Oracle. Langkah-langkah yang disajikan dalam pola ini juga berlaku jika database EE Oracle Anda sudah berjalan di Amazon RDS atau pada instans Amazon Elastic Compute Cloud (Amazon EC2).

Untuk informasi selengkapnya, lihat panduan AWS Prescriptive Guidance tentang cara Mengevaluasi penurunan database Oracle ke Standard Edition 2 di AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Oracle Database Edisi Perusahaan
- Alat klien, seperti Oracle SQL Developer atau SQL* Plus, untuk menghubungkan ke dan menjalankan perintah SQL pada database Oracle
- Pengguna database untuk melakukan penilaian; misalnya, salah satu dari berikut ini:
  - Pengguna dengan hak istimewa yang memadai untuk menjalankan penilaian AWS Schema Conversion Tool (AWS SCT)
  - Pengguna dengan hak istimewa yang cukup untuk menjalankan kueri SQL pada tabel kamus database Oracle
- Pengguna database untuk melakukan migrasi database; misalnya, salah satu dari berikut ini:
  - Pengguna dengan hak istimewa yang memadai untuk menjalankan AWS Database Migration Service (AWS DMS)

- Pengguna dengan hak istimewa yang cukup untuk melakukan ekspor dan impor Oracle Data Pump
- Pengguna dengan hak istimewa yang cukup untuk menjalankan Oracle GoldenGate

Batasan

- Amazon RDS for Oracle memiliki ukuran database maksimum. Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.

Versi produk

Logika umum yang dijelaskan dalam dokumen ini berlaku untuk versi Oracle dari 9i dan yang lebih baru. Untuk versi yang didukung dari database Amazon RDS for Oracle yang dikelola sendiri, lihat dokumentasi AWS DMS.

Untuk mengidentifikasi penggunaan fitur jika AWS SCT tidak didukung, jalankan kueri SQL pada database sumber. Untuk bermigrasi dari versi Oracle sebelumnya di mana AWS DMS dan Oracle Data Pump tidak didukung, gunakan utilitas Ekspor dan Impor Oracle.

Untuk daftar versi dan edisi yang didukung saat ini, lihat Oracle di Amazon RDS dalam dokumentasi AWS. Untuk detail tentang harga dan kelas instans yang didukung, lihat harga Amazon RDS for Oracle.

Arsitektur

Tumpukan teknologi sumber

- Oracle Database Enterprise Edition berjalan di tempat atau di Amazon EC2

Targetkan tumpukan teknologi menggunakan alat Oracle asli

- Amazon RDS for Oracle menjalankan Oracle Database SE2

1. Ekspor data dengan menggunakan Oracle Data Pump.
2. Salin file dump ke Amazon RDS melalui tautan database.

3. Impor file dump ke Amazon RDS dengan menggunakan Oracle Data Pump.

Targetkan tumpukan teknologi menggunakan AWS DMS

- Amazon RDS for Oracle menjalankan Oracle Database SE2
- AWS DMS

1. Ekspor data dengan menggunakan Oracle Data Pump dengan FLASHBACK_SCN.
2. Salin file dump ke Amazon RDS melalui tautan database.
3. Impor file dump ke Amazon RDS dengan menggunakan Oracle Data Pump.
4. Gunakan pengambilan data perubahan AWS DMS (CDC).

Alat

Layanan AWS

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud. Pola ini menggunakan Amazon RDS for Oracle.
- AWS SCT menyediakan antarmuka pengguna berbasis proyek untuk secara otomatis menilai, mengonversi, dan menyalin skema database database sumber Oracle Anda ke dalam format yang kompatibel dengan Amazon RDS for Oracle. AWS SCT memungkinkan Anda menganalisis potensi penghematan biaya yang dapat dicapai dengan mengubah jenis lisensi Anda dari Enterprise ke Standard Edition of Oracle. Bagian Evaluasi Lisensi dan Dukungan Cloud dari laporan AWS SCT memberikan informasi terperinci tentang fitur Oracle yang digunakan sehingga Anda dapat membuat keputusan berdasarkan informasi saat bermigrasi ke Amazon RDS for Oracle.

Alat-alat lainnya

- Utilitas impor dan ekspor Oracle asli mendukung pemindahan data Oracle masuk dan keluar dari database Oracle. Oracle menawarkan dua jenis utilitas impor dan ekspor database: Original Export

and Import (untuk rilis sebelumnya) dan Oracle Data Pump Export and Import (tersedia di Oracle Database 10g rilis 1 dan yang lebih baru).

- Oracle GoldenGate menawarkan kemampuan replikasi real-time sehingga Anda dapat menyinkronkan database target Anda setelah pemuatan awal. Opsi ini dapat membantu mengurangi waktu henti aplikasi selama go-live.

Epik

Buat penilaian pra-migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi persyaratan database untuk aplikasi Anda.	Pastikan aplikasi Anda disertifikasi untuk berjalan di Oracle Database SE2. Periksa langsung dengan vendor perangkat lunak, pengembang, atau dokumentasi aplikasi.	Pengembang aplikasi, DBA, Pemilik aplikasi
Selidiki penggunaan fitur EE secara langsung di database.	Untuk menentukan penggunaan fitur EE, lakukan salah satu hal berikut: • Buat laporan penilaian AWS SCT untuk database Oracle EE Anda. Laporan ini memberi tahu Anda fitur mana dari database EE Anda saat ini yang harus dihapus jika Anda ingin mengubah jenis lisensi. • Jika Anda memiliki akun Oracle Support, dapatkan dan jalankan skrip `options_packs_usage_statistics.sql` di Support document	Pemilik aplikasi, DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1317265.1 untuk menghasilkan laporan opsi dan fitur yang sedang digunakan pada database Oracle Anda. • Query DBA_FEATURE_USAGE_STATISTICS untuk menampilkan rincian semua fitur yang sedang digunakan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi penggunaan fitur EE untuk kegiatan operasional.	Administrator database atau aplikasi terkadang mengandalkan fitur khusus E untuk kegiatan operasional. Contoh umum termasuk aktivitas pemeliharaan online (membangun kembali indeks, pemindahan tabel) dan penggunaan paralelisme oleh pekerjaan batch. Dependensi ini dapat dikurangi dengan memodifikasi operasi Anda jika memungkinkan. Identifikasi penggunaan fitur-fitur ini dan buat keputusan berdasarkan biaya dibandingkan dengan manfaat. Gunakan Comparing Oracle Database EE dan tabel SE2 fitur sebagai panduan untuk mengidentifikasi fitur yang tersedia di Oracle Database. SE2	Pengembang aplikasi, DBA, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau pola beban kerja database EE Oracle.	Oracle Database SE2 secara otomatis membatasi penggunaan hingga maksimal 16 thread CPU setiap saat. Jika database Oracle EE Anda dilisensikan untuk menggunakan Oracle Diagnostic Pack, gunakan alat Automatic Workload Repository (AWR), atau tampilan DBA_HIST_*, untuk menganalisis pola beban kerja database untuk menentukan apakah batas maksimum 16 thread CPU akan berdampak negatif pada tingkat layanan saat Anda downgrade ke. SE2 Pastikan penilaian Anda mencakup periode aktivitas puncak, seperti pemrosesan akhir hari, bulan, atau tahun.	Pemilik aplikasi, DBA, Pengembang aplikasi

Siapkan infrastruktur target di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan dan mengkonfigurasi infrastruktur jaringan.	Buat virtual private cloud (VPC) dan subnet, grup keamanan, dan daftar kontrol akses jaringan.	Administrator AWS, arsitek Cloud, Administrator jaringan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan database Amazon RDS SE2 for Oracle.	Menyediakan basis data Amazon RDS SE2 for Oracle target untuk memenuhi persyaratan performa, ketersediaan, dan keamanan aplikasi Anda. Kami merekomendasikan konfigura si Multi-AZ untuk beban kerja produksi. Namun, untuk meningkatkan kinerja migrasi, Anda dapat menunda mengaktifkan Multi-AZ hingga setelah migrasi data.	Administrator cloud, arsitek Cloud, DBA, DevOps insinyur, administrator AWS
Sesuaikan lingkungan Amazon RDS.	Konfigurasikan parameter dan opsi khusus, dan aktifkan pemantauan tambahan. Untuk informasi selengkapnya, lihat Praktik terbaik untuk bermigrasi ke Amazon RDS for Oracle.	Administrator AWS, administr ator sistem AWS, administr ator Cloud, DBA, arsitek Cloud

Lakukan migrasi dry run dan pengujian aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data (dry run).	Migrasikan data dari database Oracle EE sumber ke instans database Amazon RDS for SE2 Oracle menggunak an pendekatan yang paling sesuai dengan lingkungan spesifik Anda. Pilih strategi migrasi berdasarkan faktor-fa	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ktor seperti ukuran, kompleksitas, dan jendela downtime yang tersedia. Gunakan satu atau kombinasi dari berikut ini: • Alat Oracle asli seperti Oracle Data Pump (disarankan), utilitas Oracle Import-Export, dan Oracle. GoldenGate • AWS DMS, menggunakan beban penuh dengan replikasi berkelanjutan melalui CDC.	
Validasi basis data target.	Lakukan validasi pasca-migrasi penyimpanan database dan objek kode. Tinjau log migrasi, dan perbaiki masalah yang teridentifikasi. Untuk informasi selengkapnya, lihat panduan Memigrasi database Oracle ke AWS Cloud.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aplikasi.	Administrator aplikasi dan database harus melakukan tes fungsional, kinerja, dan operasional yang sesuai. Untuk informasi selengkap nya, lihat Praktik terbaik untuk bermigrasi ke Amazon RDS for Oracle. Terakhir, dapatkan sign-off pada hasil tes dari pemangku kepentingan.	Pengembang aplikasi, Pemilik aplikasi, DBA, Insinyur migrasi, Pimpinan migrasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Segarkan data dari Oracle Database EE.	Pilih pendekatan penyegaran data berdasarkan persyaratan ketersediaan aplikasi. Untuk informasi selengkapnya, lihat metode migrasi di Strategi untuk Memigrasi Database Oracle ke AWS. Misalnya, Anda dapat mencapai downtime mendekati nol dengan menggunakan alat seperti Oracle GoldenGate atau AWS DMS dengan replikasi berkelanjutan. Jika jendela downtime memungkinkan, Anda dapat melakukan cutover data akhir menggunak	Pemilik aplikasi, Cutover lead, DBA, Insinyur migrasi, pemimpin Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an metode offline seperti Oracle Data Pump atau utilitas Ekspor-Impor Asli.	
Arahkan aplikasi ke instance database target.	Perbarui parameter koneksi dalam aplikasi dan klien lain untuk menunjuk ke database Amazon RDS SE2 for Oracle.	Pengembang aplikasi, Pemilik aplikasi, insinyur Migrasi, Pimpinan migrasi, Cutover lead
Lakukan kegiatan pasca-migrasi.	Lakukan tugas migrasi data pasca seperti mengaktifkan Multi-AZ, validasi data, dan pemeriksaan lainnya.	DBA, insinyur Migrasi
Lakukan pemantauan pasca-cutover.	Gunakan alat seperti Amazon CloudWatch dan Amazon RDS Performance Insights untuk memantau database Amazon RDS for Oracle. SE2	Pengembang aplikasi, Pemilik aplikasi, administrator AWS, DBA, insinyur migrasi

Sumber daya terkait

AWS Prescriptive Guidance

- Memigrasi database Oracle ke AWS Cloud (panduan)
- Mengevaluasi penurunan database Oracle ke Edisi Standar 2 di AWS (panduan)
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle (pola)
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle menggunakan Oracle Data Pump (pola)

Postingan blog

- Migrasi database Oracle dengan downtime mendekati nol menggunakan AWS DMS
- Menganalisis manajemen kinerja di Oracle SE menggunakan Amazon RDS for Oracle
- Mengelola paket SQL Anda di Oracle SE dengan Amazon RDS for Oracle

- Menerapkan partisi tabel di Oracle Standard Edition: Bagian 1

Replikasi database mainframe ke AWS dengan menggunakan Exactly Connect

Lucio Pereira, Sayantan Giri, dan Balaji Mohan, Amazon Web Services

Ringkasan

Pola ini menguraikan langkah-langkah untuk mereplikasi data dari basis data mainframe ke penyimpanan data Amazon dalam waktu dekat dengan menggunakan Exacently Connect. Ini mengimplementasikan arsitektur berbasis acara dengan Amazon Managed Streaming for Apache Kafka (Amazon MSK) dan konektor basis data khusus di cloud untuk meningkatkan skalabilitas, ketahanan, dan kinerja.

Tepat Connect adalah alat replikasi yang menangkap data dari sistem mainframe lama dan mengintegrasikannya ke dalam lingkungan cloud. Data direplikasi dari mainframe ke AWS melalui change data capture (CDC) dengan menggunakan aliran pesan mendekati real-time dengan jalur data heterogen latensi rendah dan throughput tinggi.

Pola ini juga mencakup strategi pemulihan bencana untuk jaringan data tangguh dengan replikasi data Multi-region dan routing failover.

Prasyarat dan batasan

Prasyarat

- Database mainframe yang ada—misalnya, IBM, IBM Information Management System (IMS) DB2, atau Virtual Storage Access Method (VSAM) —yang ingin Anda tiru ke AWS Cloud
- Akun AWS yang aktif
- AWS Direct Connect atau AWS Virtual Private Network (AWS VPN) dari lingkungan perusahaan Anda ke AWS
- Cloud pribadi virtual dengan subnet yang dapat dijangkau oleh platform lama Anda

Arsitektur

Tumpukan teknologi sumber

Lingkungan mainframe yang mencakup setidaknya satu dari database berikut:

- Basis data IBM IMS

- Basis data IBM DB2
- File VSAM

Tumpukan teknologi target

- Amazon MSK
- Amazon Elastic Kubernetes Service (Amazon EKS) dan Amazon EKS Anywhere
- Docker
- Database relasional AWS atau NoSQL seperti berikut ini:
  - Amazon DynamoDB
  - Amazon Relational Database Service (Amazon RDS) untuk Oracle, Amazon RDS for PostgreSQL, atau Amazon Aurora
  - Amazon ElastiCache untuk Redis
  - Amazon Keyspaces (untuk Apache Cassandra)

Arsitektur target

Mereplikasi data mainframe ke database AWS

Diagram berikut menggambarkan replikasi data mainframe ke database AWS seperti DynamoDB, Amazon RDS, Amazon, atau Amazon Keyspaces. ElastiCache Replikasi terjadi dalam waktu dekat dengan menggunakan Tangkap dan Penerbit Secara Tepat di lingkungan mainframe lokal Anda, Pengirim dengan Tepat di Amazon EKS Anywhere di lingkungan terdistribusi lokal Anda, serta Terapkan konektor Mesin dan database dengan Tepat di AWS Cloud.

Diagram menunjukkan alur kerja berikut:

1. Tepat Capture mendapatkan data mainframe dari log CDC dan memelihara data dalam penyimpanan transien internal.
2. Tepatnya Publisher mendengarkan perubahan dalam penyimpanan data internal dan mengirimkan catatan CDC ke Excently Dispatcher melalui koneksi. TCP/IP
3. Tepatnya Dispatcher menerima catatan CDC dari Publisher dan mengirimkannya ke Amazon MSK. Dispatcher membuat kunci Kafka berdasarkan konfigurasi pengguna dan beberapa tugas pekerja untuk mendorong data secara paralel. Dispatcher mengirimkan pengakuan kembali ke Publisher ketika catatan telah disimpan di Amazon MSK.

4. Amazon MSK memegang catatan CDC di lingkungan cloud. Ukuran partisi topik tergantung pada persyaratan sistem pemrosesan transaksi (TPS) Anda untuk throughput. Kunci Kafka wajib untuk transformasi lebih lanjut dan pemesanan transaksi.
5. Mesin Terapkan Secara Tepat mendengarkan catatan CDC dari Amazon MSK dan mengubah data (misalnya, dengan memfilter atau memetakan) berdasarkan persyaratan basis data target. Anda dapat menambahkan logika yang disesuaikan ke skrip SQD Tepatnya. (SQD adalah bahasa hak milik Tepat.) Mesin Terapkan Secara Tepat mengubah setiap rekaman CDC ke format Apache Avro atau JSON dan mendistribusikannya ke berbagai topik berdasarkan kebutuhan Anda.
6. Topik Kafka target menyimpan catatan CDC dalam berbagai topik berdasarkan basis data target, dan Kafka memfasilitasi pemesanan transaksi berdasarkan kunci Kafka yang ditentukan. Tombol partisi sejajar dengan partisi yang sesuai untuk mendukung proses berurutan.
7. Konektor database (aplikasi Java yang disesuaikan) mendengarkan catatan CDC dari Amazon MSK dan menyimpannya di database target.
8. Anda dapat memilih database target berdasarkan kebutuhan Anda. Pola ini mendukung NoSQL dan database relasional.

Pemulihan bencana

Kesinambungan bisnis adalah kunci keberhasilan organisasi Anda. AWS Cloud menyediakan kemampuan untuk ketersediaan tinggi (HA) dan pemulihan bencana (DR), serta mendukung paket failover dan fallback organisasi Anda. Pola ini mengikuti strategi active/passive DR dan memberikan panduan tingkat tinggi untuk menerapkan strategi DR yang memenuhi persyaratan RTO dan RPO Anda.

Diagram berikut menggambarkan alur kerja DR.

Diagram menunjukkan yang berikut:

1. Failover semi-otomatis diperlukan jika terjadi kegagalan di AWS Region 1. Dalam kasus kegagalan di Wilayah 1, sistem harus memulai perubahan perutean untuk menghubungkan Tepatnya Dispatcher ke Wilayah 2.
2. Amazon MSK mereplikasi data melalui mirroring antar Wilayah, Untuk alasan ini, selama failover, cluster MSK Amazon di Wilayah 2 harus dipromosikan sebagai pemimpin utama.
3. Tepat Apply Engine dan konektor database adalah aplikasi stateless yang dapat bekerja di Wilayah mana pun.

4. Sinkronisasi database tergantung pada database target. Misalnya, DynamoDB dapat menggunakan tabel global, ElastiCache dan dapat menggunakan datastores global.

Pemrosesan latensi rendah dan throughput tinggi melalui konektor database

Konektor database adalah komponen penting dalam pola ini. Konektor mengikuti pendekatan berbasis pendengar untuk mengumpulkan data dari Amazon MSK dan mengirim transaksi ke database melalui pemrosesan throughput tinggi dan latensi rendah untuk aplikasi kritis misi (tingkatan 0 dan 1). Diagram berikut menggambarkan proses.

Pola ini mendukung pengembangan aplikasi yang disesuaikan dengan konsumsi ulir tunggal melalui mesin pemrosesan multithreaded.

1. Utas utama konektor mengkonsumsi catatan CDC dari Amazon MSK dan mengirimkannya ke kumpulan utas untuk diproses.
2. Thread dari kumpulan thread memproses catatan CDC dan mengirimkannya ke database target.
3. Jika semua utas sibuk, catatan CDC ditahan oleh antrian utas.
4. Utas utama menunggu untuk menghapus semua catatan dari antrian utas dan melakukan offset ke Amazon MSK.
5. Utas anak menangani kegagalan. Jika kegagalan terjadi selama pemrosesan, pesan yang gagal dikirim ke topik DLQ (antrian surat mati).
6. Thread anak memulai pembaruan bersyarat (lihat Ekspresi kondisi dalam dokumentasi DynamoDB), berdasarkan stempel waktu mainframe, untuk menghindari duplikasi atau pembaruan apa pun dalam database. out-of-order

Untuk informasi tentang cara menerapkan aplikasi konsumen Kafka dengan kemampuan multi-threading, lihat posting blog Konsumsi Pesan Multi-Threaded dengan Konsumen Apache Kafka di situs web Confluent.

Alat

Layanan AWS

- Amazon Managed Streaming for Apache Kafka (Amazon MSK) adalah layanan yang dikelola sepenuhnya yang membantu Anda membangun dan menjalankan aplikasi yang menggunakan Apache Kafka untuk memproses data streaming.

- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa harus menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- Amazon EKS Anywhere membantu Anda menerapkan, menggunakan, dan mengelola klaster Kubernetes yang berjalan di pusat data Anda sendiri.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan dapat diskalakan.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- Amazon ElastiCache membantu Anda mengatur, mengelola, dan menskalakan lingkungan cache dalam memori terdistribusi di AWS Cloud.
- Amazon Keyspaces (untuk Apache Cassandra) adalah layanan database terkelola yang membantu Anda memigrasi, menjalankan, dan menskalakan beban kerja Cassandra Anda di AWS Cloud.

Alat-alat lainnya

- Connect mengintegrasikan data dari sistem mainframe lama seperti dataset VSAM atau database mainframe IBM ke dalam platform cloud dan data generasi berikutnya.

Praktik terbaik

- Temukan kombinasi terbaik dari partisi Kafka dan konektor multi-ulir untuk menyeimbangkan kinerja dan biaya yang optimal. Beberapa Instans Pengambilan dan Dispatcher Secara Tepat dapat meningkatkan biaya karena konsumsi MIPS (juta instruksi per detik) yang lebih tinggi.
- Hindari menambahkan manipulasi data dan logika transformasi ke konektor database. Untuk tujuan ini, gunakan Mesin Terapkan Tepatnya, yang menyediakan waktu pemrosesan dalam mikrodetik.
- Buat permintaan berkala atau panggilan pemeriksaan kesehatan ke database (detak jantung) di konektor database untuk sering menghangatkan koneksi dan mengurangi latensi.
- Terapkan logika validasi kumpulan utas untuk memahami tugas yang tertunda dalam antrian utas dan tunggu semua utas selesai sebelum polling Kafka berikutnya. Ini membantu menghindari kehilangan data jika node, wadah, atau proses mogok.
- Paparkan metrik latensi melalui titik akhir kesehatan untuk meningkatkan kemampuan observabilitas melalui dasbor dan mekanisme penelusuran.

Epik

Siapkan lingkungan sumber (di tempat)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan proses mainframe (batch atau utilitas online) untuk memulai proses CDC dari database mainframe.	1. Identifikasi lingkungan mainframe. 2. Identifikasi database mainframe yang akan terlibat dalam proses CDC. 3. Di lingkungan mainframe , kembangkan proses yang meluncurkan alat CDC untuk menangkap perubahan dalam database sumber. Untuk petunjuk, lihat dokumentasi mainframe Anda. 4. Dokumentasikan proses CDC, termasuk konfigurasi. 5. Menyebarkan proses di lingkungan pengujian dan produksi.	Insinyur mainframe
Aktifkan aliran log basis data mainframe.	1. Konfigurasikan aliran log di lingkungan mainframe untuk menangkap log CDC. Untuk petunjuk, lihat dokumentasi mainframe Anda. 2. Uji aliran log untuk memastikan bahwa mereka menangkap data yang diperlukan.	Spesialis DB Mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Menyebarkan aliran log di lingkungan pengujian dan produksi.	
Gunakan komponen Capture untuk menangkap catatan CDC.	1. Instal dan konfigurasikan komponen Tangkap Tepatnya di lingkungan mainframe. Untuk instruksi, lihat dokumentasi Tepatnya. 2. Uji konfigurasi untuk memastikan bahwa komponen Capture berfungsi dengan benar. 3. Siapkan proses replikasi untuk mereplikasi catatan CDC yang ditangkap melalui komponen Capture. 4. Dokumentasikan konfigurasi Capture untuk setiap database sumber. 5. Kembangkan sistem pemantauan untuk memastikan bahwa komponen Capture mengumpulkan log dengan benar dari waktu ke waktu. 6. Menyebarkan instalasi dan konfigurasi di lingkungan pengujian dan produksi.	Insinyur mainframe, Tepatnya Connect SME

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan komponen Publisher untuk mendengar kan komponen Capture.	1. Instal dan konfigurasikan komponen Tepat Penerbit di lingkungan mainframe . Untuk instruksi, lihat dokumentasi Tepatnya. 2. Uji konfigurasi untuk memastikan bahwa komponen Publisher berfungsi dengan benar. 3. Siapkan proses replikasi untuk mempublikasikan catatan CDC ke komponen Pengirim Tepatnya dari Publisher. 4. Dokumentasikan konfigura si Penerbit. 5. Mengembangkan sistem pemantauan untuk memastikan bahwa komponen Publisher bekerja dengan baik dari waktu ke waktu. 6. Menyebarkan instalasi dan konfigurasi di lingkungan pengujian dan produksi.	Insinyur mainframe, Tepatnya Connect SME

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan Amazon EKS Anywhere di lingkungan terdistribusi lokal.	1. Instal Amazon EKS Anywhere di infrastru ktur lokal dan pastikan sudah dikonfigurasi dengan benar. Untuk petunjuk, lihat dokumentasi Amazon EKS Anywhere. 2. Siapkan lingkungan jaringan yang aman untuk klaster Kubernetes, termasuk firewall. 3. Menerapkan dan menguji contoh penerapan aplikasi ke cluster Amazon EKS Anywhere. 4. Menerapkan kemampuan penskalaan otomatis untuk cluster. 5. Mengembangkan dan menerapkan prosedur pencadangan dan pemulihan bencana.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan dan konfigurasikan komponen Dispatcher di lingkungan terdistribusi untuk mempublikasikan topik di AWS Cloud.	1. Konfigurasikan dan kontainerisasi komponen Tepat Dispatcher. Untuk instruksi, lihat dokumentasi Tepatnya. 2. Menerapkan image Dispatcher Docker ke lingkungan Amazon EKS Anywhere lokal. 3. Siapkan koneksi aman antara AWS Cloud dan Dispatcher. 4. Kembangkan sistem pemantauan untuk memastikan bahwa komponen Dispatcher berfungsi dengan baik dari waktu ke waktu. 5. Menyebarkan instalasi dan konfigurasi di lingkungan pengujian dan produksi.	DevOps insinyur, Tepatnya Connect SME

Siapkan lingkungan target (AWS)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan kluster Amazon EKS di Wilayah AWS yang ditentukan.	1. Masuk ke akun AWS Anda dan konfigurasikan untuk memastikan bahwa izin yang diperlukan tersedia untuk membuat dan mengelola klaster Amazon EKS.	DevOps insinyur, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buat virtual private cloud (VPC) dan subnet di Wilayah AWS yang dipilih. Untuk petunjuk, lihat dokumentasi Amazon EKS. 3. Buat dan konfigurasikan grup keamanan jaringan yang diperlukan untuk memungkinkan komunikasi antara kluster Amazon EKS dan sumber daya lainnya di VPC. Untuk informasi selengkapnya, lihat dokumentasi Amazon EKS. 4. Buat klaster Amazon EKS dan konfigurasikan dengan ukuran grup node dan jenis instance yang benar. 5. Validasi klaster Amazon EKS dengan menerapkan contoh aplikasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan kluster MSK dan mengkonfigurasi topik Kafka yang berlaku.	1. Konfigurasikan akun AWS Anda untuk memastikan bahwa izin yang diperlukan tersedia untuk membuat dan mengelola klaster MSK. 2. Buat dan konfigurasikan grup keamanan jaringan yang diperlukan untuk memungkinkan komunikasi antara cluster MSK dan sumber daya lainnya di VPC. Untuk informasi selengkapnya, lihat dokumentasi Amazon VPC. 3. Buat cluster MSK dan konfigurasikan untuk memasukkan topik Kafka yang akan digunakan oleh aplikasi. Untuk informasi selengkapnya, lihat dokumentasi MSK Amazon.	DevOps insinyur, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan komponen Apply Engine untuk mendengarkan topik Kafka yang direplikasi.	1. Konfigurasikan dan kontainerisasi komponen Exactly Apply Engine. 2. Menerapkan image Apply Engine Docker ke cluster Amazon EKS di akun AWS Anda. 3. Siapkan Apply Engine untuk mendengarkan topik MSK. 4. Kembangkan dan konfigura sikan skrip SQD di Apply Engine untuk menangani penyaringan dan transform asi. Untuk informasi selengkapnya, lihat dokumentasi Tepatnya. 5. Menyebarkan Apply Engine di lingkungan pengujian dan produksi.	Tepatnya Connect SME

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan instans DB di AWS Cloud.	1. Konfigurasikan akun AWS Anda untuk memastikan bahwa izin yang diperlukan tersedia untuk membuat dan mengelola kluster dan tabel DB. Untuk petunjuk, lihat dokumentasi AWS untuk layanan database AWS yang ingin Anda gunakan. (Lihat bagian Sumber Daya untuk tautan.) 2. Buat VPC dan subnet di Wilayah AWS yang dipilih. 3. Buat dan konfigurasikan grup keamanan jaringan yang diperlukan untuk memungkinkan komunikasi antara instans DB dan sumber daya lainnya di VPC. 4. Buat database dan konfigurasikan untuk menyertakan tabel yang akan digunakan aplikasi. 5. Merancang dan memvalidasi skema database.	Insinyur data, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan gunakan konektor database untuk mendengarkan topik yang diterbitkan oleh Apply Engine.	1. Rancang konektor database untuk menghubungkan topik Kafka dengan database AWS yang Anda buat di langkah sebelumnya. 2. Kembangkan konektor berdasarkan basis data target. 3. Konfigurasikan konektor untuk mendengarkan topik Kafka yang diterbitkan oleh Apply Engine. 4. Terapkan konektor ke cluster Amazon EKS.	Pengembang aplikasi, arsitek Cloud, Insinyur data

Menyiapkan kelangsungan bisnis dan pemulihan bencana

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan tujuan pemulihan bencana untuk aplikasi bisnis Anda.	1. Tentukan tujuan RPO dan RTO untuk jaringan pipa CDC berdasarkan kebutuhan bisnis Anda dan analisis dampak. 2. Menentukan prosedur komunikasi dan pemberitahuan untuk memastikan bahwa semua pemangku kepentingan mengetahui rencana pemulihan bencana.	Arsitek cloud, Insinyur data, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Menentukan anggaran dan sumber daya yang dibutuhkan untuk melaksanakan rencana pemulihan bencana. 4. Dokumentasikan tujuan pemulihan bencana, termasuk tujuan RPO dan RTO.	
Merancang strategi pemulihan bencana berdasarkan RTO/RPO yang ditentukan.	1. Tentukan strategi pemulihan bencana yang paling tepat untuk jaringan pipa CDC berdasarkan kekritisan dan persyaratan pemulihan Anda. 2. Mendefinisikan arsitektur pemulihan bencana dan topologi. 3. Tentukan prosedur failover dan failback untuk saluran pipa CDC untuk memastikan bahwa mereka dapat dengan cepat dan mulus beralih ke Wilayah cadangan. 4. Mendokumentasikan strategi dan prosedur pemulihan bencana dan memastikan bahwa semua pemangku kepentingan memiliki pemahaman yang jelas tentang desain.	Arsitek cloud, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Penyediaan cluster dan konfigurasi pemulihan bencana.	1. Menyediakan Wilayah AWS sekunder untuk pemulihan bencana. 2. Di Wilayah AWS sekunder, buat lingkungan yang identik dengan Wilayah AWS utama. 3. Konfigurasikan Apache Kafka MirrorMaker antara Wilayah primer dan sekunder. Untuk informasi selengkapnya, lihat dokumentasi MSK Amazon. 4. Konfigurasikan aplikasi siaga di Wilayah sekunder. 5. Konfigurasikan replikasi database antara Wilayah primer dan sekunder.	DevOps insinyur, Administrator jaringan, Arsitek cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji pipa CDC untuk pemulihan bencana.	1. Tentukan ruang lingkup dan tujuan uji pemulihan bencana untuk pipa CDC, termasuk skenario pengujian dan RTO yang akan dicapai. 2. Identifikasi lingkungan pengujian dan infrastru ktur untuk melakukan uji pemulihan bencana. 3. Siapkan dataset pengujian dan skrip untuk mensimula sikan skenario kegagalan. 4. Verifikasi integritas dan konsistensi data untuk memastikan bahwa tidak ada kehilangan data.	Pemilik aplikasi, Insinyur data, arsitek Cloud

Sumber daya terkait

Sumber daya AWS

- Amazon DynamoDB
- Ekspresi kondisi dengan Amazon DynamoDB
- Amazon EKS
- Amazon EKS Anywhere
- Amazon ElasticCache
- Keyspaces Amazon
- Amazon MSK
- Amazon RDS dan Amazon Aurora
- Amazon VPC

Tepat Connect sumber daya

- Ikhtisar Connect Tepatnya
- Ubah Pengambilan Data dengan Tepat Connect

Sumber daya pertemuan

- Konsumsi Pesan Multi-Threaded dengan Konsumen Apache Kafka

Jadwalkan pekerjaan untuk Amazon RDS untuk PostgreSQL dan Aurora PostgreSQL dengan menggunakan Lambda dan Secrets Manager

Yaser Raja, Amazon Web Services

Ringkasan

Untuk database lokal dan database yang di-host di instans Amazon Elastic Compute Cloud (Amazon EC2), administrator database sering menggunakan utilitas cron untuk menjadwalkan pekerjaan.

Misalnya, pekerjaan untuk ekstraksi data atau pekerjaan untuk pembersihan data dapat dengan mudah dijadwalkan menggunakan cron. Untuk pekerjaan ini, kredensi database biasanya dikodekan keras atau disimpan dalam file properti. Namun, saat Anda bermigrasi ke Amazon Relational Database Service (Amazon RDS) atau Amazon Aurora PostgreSQL Compatible Edition, Anda kehilangan kemampuan untuk masuk ke instans host untuk menjadwalkan tugas cron.

Pola ini menjelaskan cara menggunakan AWS Lambda dan menjadwalkan pekerjaan AWS Secrets Manager untuk Amazon RDS for PostgreSQL dan database yang kompatibel dengan Aurora PostgreSQL setelah migrasi.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Database yang kompatibel dengan Amazon RDS for PostgreSQL atau Aurora PostgreSQL

Batasan

- Pekerjaan harus diselesaikan dalam waktu 15 menit, yang merupakan batas waktu tunggu fungsi Lambda. Untuk batasan lainnya, lihat AWS Lambda dokumentasi.
- Kode Job harus ditulis dalam bahasa yang didukung oleh Lambda.

Arsitektur

Tumpukan teknologi sumber

Tumpukan ini menampilkan pekerjaan yang ditulis dalam bahasa seperti Bash, Python, dan Java. Kredensi database disimpan dalam file properti, dan pekerjaan dijadwalkan menggunakan Linux cron.

Tumpukan teknologi target

Tumpukan ini memiliki fungsi Lambda yang menggunakan kredensil yang disimpan di Secrets Manager untuk terhubung ke database dan untuk melakukan aktivitas. Fungsi Lambda dimulai pada interval terjadwal dengan menggunakan Amazon Events. CloudWatch

Arsitektur target

Alat

- Amazon CloudWatch Events memberikan aliran peristiwa sistem yang mendekati real-time yang menjelaskan perubahan AWS sumber daya. Dengan menggunakan aturan sederhana yang dapat Anda atur dengan cepat, Anda dapat mencocokkan acara dan mengarahkannya ke satu atau lebih fungsi atau aliran target. CloudWatch Peristiwa menjadi sadar akan perubahan operasional saat terjadi. Ini menanggapi perubahan operasional ini dan mengambil tindakan korektif seperlunya, dengan mengirim pesan untuk menanggapi lingkungan, mengaktifkan fungsi, membuat perubahan, dan menangkap informasi negara. Anda juga dapat menggunakan CloudWatch Acara untuk menjadwalkan tindakan otomatis yang dimulai sendiri pada waktu-waktu tertentu menggunakan ekspresi cron atau rate.
- AWS Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda konsumsi; tidak ada biaya ketika kode Anda tidak berjalan. Dengan Lambda, Anda dapat menjalankan kode untuk hampir semua jenis aplikasi atau layanan backend tanpa administrasi. Lambda menjalankan kode Anda pada infrastruktur komputasi ketersediaan tinggi dan mengelola semua sumber daya komputasi, termasuk pemeliharaan server dan sistem operasi, penyediaan kapasitas dan penskalaan otomatis, pemantauan kode, dan pencatatan. Yang perlu Anda lakukan adalah memberikan kode Anda dalam salah satu bahasa yang didukung Lambda.
- AWS Secrets Managermembantu Anda melindungi rahasia untuk mengakses aplikasi, layanan, dan sumber daya TI Anda. Anda dapat dengan mudah memutar, mengelola, dan mengambil kredenal database, kunci API, dan rahasia lainnya sepanjang siklus hidupnya. Pengguna dan aplikasi mengambil rahasia dengan memanggil Secrets Manager APIs, yang menghilangkan

kebutuhan untuk hard-code informasi sensitif dalam teks biasa. Secrets Manager menawarkan rotasi rahasia dengan integrasi bawaan untuk Amazon RDS, Amazon Redshift, dan Amazon DocumentDB. Layanan ini dapat diperluas ke jenis rahasia lainnya, termasuk kunci API dan OAuth token. Secrets Manager memungkinkan Anda mengontrol akses ke rahasia menggunakan izin halus dan mengaudit rotasi rahasia secara terpusat untuk sumber daya di, layanan pihak ketiga AWS Cloud, dan di tempat.

Epik

Simpan kredensil database di Secrets Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna database untuk fungsi Lambda.	Ini adalah praktik yang baik untuk menggunakan pengguna database terpisah untuk berbagai bagian aplikasi Anda. Jika pengguna database terpisah sudah ada untuk pekerjaan cron Anda, gunakan itu. Jika tidak, buat pengguna database baru. Untuk informasi selengkapnya, lihat Mengelola pengguna dan peran PostgreSQL (posting blog).AWS	DBA
Simpan kredensi database sebagai rahasia di Secrets Manager.	Ikuti petunjuk di Buat rahasia database (Dokumentasi Secrets Manager).	DBA, DevOps

Penulis kode untuk fungsi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih bahasa pemrograman yang didukung oleh Lambda.	Untuk daftar bahasa yang didukung, lihat runtime Lambda (dokumentasi Lambda).	Developer
Tulis logika untuk mengambil kredensi database dari Secrets Manager.	Untuk kode contoh, lihat Cara menyediakan kredensil database secara aman ke fungsi Lambda dengan menggunakan AWS Secrets Manager (posting blog).AWS	Developer
Tulis logika untuk melakukan aktivitas database terjadwal.	Migrasikan kode yang ada untuk pekerjaan penjadwalan yang Anda gunakan di lokasi ke fungsi Lambda. Untuk informasi selengkapnya, lihat Menerapkan fungsi Lambda (dokumentasi Lambda).	Developer

Terapkan kode dan buat fungsi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat paket penyebaran fungsi Lambda.	Paket ini berisi kode dan dependensinya. Untuk informasi selengkapnya, lihat Paket penerapan (dokumentasi Lambda).	Developer
Buat fungsi Lambda.	Di konsol Lambda, pilih Buat fungsi, masukkan nama	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	fungsi, pilih lingkungan runtime, lalu pilih Buat fungsi.	
Unggah paket deployment.	Pilih fungsi Lambda yang Anda buat untuk membuka konfigurasinya. Anda dapat menulis kode Anda langsung di bagian kode atau mengunggah paket penyebaran Anda. Untuk mengunggah paket Anda, buka bagian Kode fungsi, pilih jenis entri Kode untuk mengunggah file.zip, lalu pilih paket.	DevOps
Konfigurasikan fungsi Lambda sesuai kebutuhan Anda.	Misalnya, Anda dapat mengatur parameter Timeout ke durasi yang Anda harapkan dari fungsi Lambda. Untuk informasi selengkapnya, lihat Mengonfigurasi opsi fungsi (Dokumentasi Lambda).	DevOps
Tetapkan izin untuk peran fungsi Lambda untuk mengakses Secrets Manager.	Untuk petunjuk, lihat Menggunakan rahasia dalam AWS Lambda fungsi (Dokumentasi Secrets Manager).	DevOps
Uji fungsi Lambda.	Memulai fungsi Lambda secara manual untuk memastikannya berfungsi seperti yang diharapkan.	DevOps

Jadwalkan fungsi Lambda dengan menggunakan Acara CloudWatch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan untuk menjalankan fungsi Lambda sesuai jadwal.	Jadwalkan fungsi Lambda dengan menggunakan CloudWatch Acara. Untuk petunjuk, lihat Menjadwalkan fungsi Lambda menggunakan CloudWatch Acara (tutorial CloudWatch Acara).	DevOps

Sumber daya terkait

- AWS Secrets Manager
- Memulai dengan Lambda
- Membuat Aturan CloudWatch Acara yang Memicu Peristiwa
- AWS Lambda Batas
- Kueri AWS database Anda dari aplikasi tanpa server Anda (posting blog)

Mengirim pemberitahuan untuk instans database Amazon RDS for SQL Server dengan menggunakan server SMTP lokal dan Database Mail

Nishad Mankar, Amazon Web Services

Ringkasan

Database Mail (dokumentasi Microsoft) mengirimkan pesan email, seperti pemberitahuan atau peringatan, dari database Microsoft SQL Server dengan menggunakan server Simple Mail Transfer Protocol (SMTP). Dokumentasi Amazon Relational Database Service (Amazon RDS) untuk Microsoft SQL Server memberikan petunjuk untuk menggunakan Amazon Simple Email Service (Amazon SES) sebagai server SMTP untuk Database Mail. Untuk informasi selengkapnya, lihat Menggunakan Database Mail di Amazon RDS for SQL Server. Sebagai konfigurasi alternatif, pola ini menjelaskan cara mengonfigurasi Database Mail untuk mengirim email dari instans Amazon RDS for SQL Server database (DB) dengan menggunakan server SMTP lokal sebagai server email.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Instans Amazon RDS DB yang menjalankan SQL Server edisi Standar atau Perusahaan
- Alamat IP atau nama host server SMTP lokal
- Aturan grup keamanan masuk yang memungkinkan koneksi ke instans Amazon RDS for SQL Server DB dari alamat IP server SMTP
- Sambungan, seperti koneksi AWS Direct Connect, antara jaringan lokal Anda dan virtual private cloud (VPC) yang berisi instans Amazon RDS DB

Batasan

- Edisi ekspres SQL Server tidak didukung.
- Untuk informasi selengkapnya tentang batasan, lihat Batasan dalam Menggunakan Email Database di Amazon RDS for SQL Server dalam dokumentasi Amazon RDS.

Versi produk

- Edisi standar dan Enterprise dari versi SQL Server didukung dalam RDS

Arsitektur

Tumpukan teknologi target

- Contoh database Amazon RDS for SQL Server
- Aturan penerusan Amazon Route 53
- Database Mail
- Server SMTP lokal
- Studio Manajemen Server Microsoft SQL (SSMS)

Arsitektur target

Gambar berikut menunjukkan arsitektur target untuk pola ini. Ketika peristiwa atau tindakan terjadi yang memulai pemberitahuan atau peringatan mengenai instance database, Amazon RDS for SQL Server menggunakan Database Mail untuk mengirim pemberitahuan email. Database Mail menggunakan server SMTP lokal untuk mengirim email.

Alat

Layanan AWS

- Amazon Relational Database Service (Amazon RDS) untuk Microsoft SQL Server membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional SQL Server di AWS Cloud.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan terukur.

Alat-alat lainnya

- Database Mail adalah alat yang mengirim pesan e-mail, seperti pemberitahuan dan peringatan, dari SQL Server Database Engine kepada pengguna.
- Microsoft SQL Server Management Studio (SSMS) adalah alat untuk mengelola SQL Server, termasuk mengakses, mengkonfigurasi, dan mengelola komponen SQL Server. Dalam pola ini, Anda menggunakan SSMS untuk menjalankan perintah SQL untuk mengatur Database Mail pada instans Amazon RDS for SQL Server DB.

Epik

Mengaktifkan konektivitas jaringan dengan server SMTP lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus Multi-AZ dari instans RDS DB.	Jika Anda menggunakan instans Multi-Zone RDS DB, ubah instans Multi-AZ menjadi instans Single-AZ . Setelah selesai mengonfig urasi Database Mail, Anda akan mengonversi instans DB kembali ke penerapan Multi-AZ. Konfigurasi Database Mail kemudian bekerja di kedua node primer dan sekunder. Untuk petunjuk, lihat Menghapus Multi-AZ dari instans Microsoft SQL Server DB.	DBA
Buat daftar izin untuk titik akhir Amazon RDS atau alamat IP di server SMTP lokal.	Server SMTP berada di luar jaringan AWS. Di server SMTP lokal, buat daftar izin yang memungkinkan server berkomunikasi dengan titik akhir keluar atau alamat IP untuk instans Amazon RDS atau instans Amazon Elastic Compute Cloud (Amazon) yang dihosting di Amazon RDS. EC2 Prosedur ini bervariasi dari organisasi ke organisasi. Untuk informasi selengkapnya tentang titik akhir instans DB, lihat	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Menemukan titik akhir instans DB dan nomor port.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus batasan port 25.	Secara default, AWS membatasi port 25 pada EC2 instans. Untuk menghapus pembatasan port 25, lakukan hal berikut: 1. Masuk dengan akun AWS Anda, lalu buka formulir Permintaan untuk menghapus batasan pengiriman email. 2. Masukkan alamat email Anda sehingga AWS Support dapat menghubungi Anda dengan pembaruan tentang permintaan Anda. 3. Berikan informasi yang diperlukan di bidang Use case description. 4. Pilih Kirim. **Note** • Jika Anda memiliki instans di lebih dari satu Wilayah AWS, kirimkan permintaan terpisah untuk setiap Wilayah. • Diperlukan waktu hingga 48 jam untuk memproses permintaan Anda.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan aturan Route 53 untuk menyelesaikan kueri DNS untuk server SMTP.	Gunakan Route 53 untuk menyelesaikan kueri DNS antara sumber daya AWS Anda dan server SMTP lokal. Anda harus membuat aturan yang meneruskan kueri DNS ke domain server SMTP, seperti. `example.com` Untuk petunjuknya, lihat Membuat aturan penerusan dalam dokumentasi Route 53.	Administrator jaringan

Mengatur Database Mail di Amazon RDS for SQL Server DB instans

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan Surat Database.	Buat grup parameter untuk Database Mail, atur `database mail xps` parameternya1, lalu kaitkan grup parameter Database Mail dengan instans RDS DB target. Untuk petunjuk, lihat Mengaktifkan Mail Database dalam dokumentasi Amazon RDS. Jangan lanjutkan ke bagian Mengkonfigurasi Database Mail dalam petunjuk ini. Konfigurasi untuk server SMTP lokal berbeda dari Amazon SES.	DBA
Hubungkan ke instans DB.	Dari host bastion, gunakan Microsoft SQL Server	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Management Studio (SSMS) untuk terhubung ke instans database Amazon RDS for SQL Server. Untuk petunjuk, lihat Menyambungkan ke instans DB yang menjalankan mesin database Microsoft SQL Server. Jika Anda menemukan kesalahan, lihat referensi pemecahan masalah koneksi di bagian Sumber daya terkait.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat profil.	Di SSMS, masukkan pernyataan SQL berikut untuk membuat profil Database Mail. Ganti nilai-nilai berikut: • Untuk`profile_name` , masukkan nama untuk profil baru. • Untuk`description` , masukkan deskripsi singkat tentang profil baru. Untuk informasi selengkapnya tentang prosedur tersimpan ini dan argumennya, lihat sysmail_add_profile_sp di dokumentasi Microsoft. `EXECUTE msdb.dbo.sysmail_add_profile_sp @profile_name = 'SQL Alerts profile', @description = 'Profile used for sending outgoing notifications using OM SMTP Server.';`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan prinsipal ke profil.	Masukkan pernyataan SQL berikut untuk menambahk an prinsip publik atau pribadi ke profil Database Mail. Pengguna utama adalah entitas yang dapat meminta sumber daya SQL Server. Ganti nilai-nilai berikut: • Untuk`profile_name` , masukkan nama profil yang Anda buat sebelumnya. • Untuk`principal_name` , masukkan nama pengguna database atau peran. Nilai ini harus dipetakan ke pengguna otentikas i SQL Server, pengguna Otentikasi Windows, atau grup Otentikasi Windows. Untuk informasi selengkapnya tentang prosedur tersimpan ini dan argumennya, lihat sysmail_add_principalprofil e_sp di dokumentasi Microsoft . `EXECUTE msdb.dbo. sysmail_add_princi palprofile_sp @profile_name = 'SQL Alerts profile', @principal_name = 'public',`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`@is_default = 1 ;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat akun.	Masukkan pernyataan SQL berikut untuk membuat akun Database Mail. Ganti nilai-nilai berikut: • Untuk`account_name`, masukkan nama untuk akun baru. • Untuk`description`, masukkan deskripsi singkat tentang akun baru. • Untuk`email_address`, masukkan alamat e-mail untuk mengirim pesan Database Mail dari. • Untuk`display_address`, masukkan nama tampilan yang akan digunakan untuk pesan keluar untuk akun ini, seperti`SQL Server Automated Notification`. Anda juga dapat menggunakan nilai yang Anda masukkan`email_address`. • Untuk`mailserver_name`, masukkan nama atau alamat IP server email SMTP. • Untuk`port`, tinggalkan nilai`25`. • Untuk`enable_ssl`, tinggalkan nilai di 1 atau	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	masukkan `0` jika Anda tidak ingin Database Mail mengenkripsi komunikasi dengan menggunakan SSL. • Untuk`username`, masukkan nama pengguna untuk masuk ke server email SMTP. Jika server tidak memerlukan otentikasi, masukkan`NULL`. • Untuk`password`, masukkan kata sandi untuk masuk ke server email SMTP. Jika server tidak memerlukan otentikasi, masukkan`NULL`. Untuk informasi selengkapnya tentang prosedur tersimpan ini dan argumennya, lihat sysmail_add_account_sp di dokumentasi Microsoft. `EXECUTE msdb.dbo.sysmail_add_account_sp @account_name = 'SQL Alerts account', @description = 'Database Mail account for sending outgoing notifications.', @email_address = 'xyz@example.com', @display_name = 'xyz@example.com',`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>@mailserver_name = 'test_smtp.example.com', @port = 25, @enable_ssl = 1, @username = 'SMTP-username', @password = 'SMTP-password';</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan akun ke profil.	Masukkan pernyataan SQL berikut untuk menambahkan akun Database Mail ke profil Database Mail. Ganti nilai-nilai berikut: • Untuk`profile_name` , masukkan nama profil yang Anda buat sebelumnya. • Untuk`account_name` , masukkan nama akun yang Anda buat sebelumnya. Untuk informasi selengkapnya tentang prosedur tersimpan ini dan argumennya, lihat sysmail_add_profileaccount_sp di dokumentasi Microsoft. `EXECUTE msdb.dbo.sysmail_add_profileaccount_sp @profile_name = 'SQL Alerts profile', @account_name = 'SQL Alerts account', @sequence_number = 1;`	DBA
(Opsional) Tambahkan Multi-AZ ke instans RDS DB.	Jika Anda ingin menambahkan Multi-AZ dengan Database Mirroring (DBM) atau Always On Availability Groups (AGs), lihat petunjuk di Menambahkan Multi-AZ ke instans Microsoft SQL Server DB.	DBA

Sumber daya terkait

- Menggunakan Database Mail di Amazon RDS for SQL Server (dokumentasi Amazon RDS)
- Bekerja dengan lampiran file (dokumentasi Amazon RDS)
- Memecahkan masalah koneksi ke instans SQL Server DB Anda (dokumentasi Amazon RDS)
- Tidak dapat terhubung ke instans Amazon RDS DB (dokumentasi Amazon RDS)

Siapkan pemulihan bencana untuk SAP di IBM Db2 di AWS

Ambarish Satarkar dan Debasis Sahoo, Amazon Web Services

Ringkasan

Pola ini menguraikan langkah-langkah untuk menyiapkan sistem pemulihan bencana (DR) untuk beban kerja SAP dengan IBM Db2 sebagai platform database, berjalan di Amazon Web Services (AWS) Cloud. Tujuannya adalah untuk memberikan solusi berbiaya rendah untuk memberikan kelangsungan bisnis jika terjadi pemadaman.

Pola menggunakan pendekatan cahaya pilot. Dengan menerapkan pilot light DR di AWS, Anda dapat mengurangi waktu henti dan menjaga kelangsungan bisnis. Pendekatan pilot light berfokus pada pengaturan lingkungan DR minimal di AWS, termasuk sistem SAP dan database Db2 siaga, yang disinkronkan dengan lingkungan produksi.

Solusi ini dapat diskalakan. Anda dapat memperluasnya ke lingkungan pemulihan bencana skala penuh sesuai kebutuhan.

Prasyarat dan batasan

Prasyarat

- Instans SAP yang berjalan pada instans Amazon Elastic Compute Cloud (Amazon EC2)
- Database IBM Db2
- Sistem operasi yang didukung oleh SAP Product Availability Matrix (PAM)
- Nama host database fisik yang berbeda untuk host database produksi dan siaga
- Bucket Amazon Simple Storage Service (Amazon S3) di setiap Wilayah AWS dengan Replikasi Lintas Wilayah (CRR) diaktifkan

Versi produk

- IBM Db2 Database versi 11.5.7 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- Amazon EC2

- Amazon Simple Storage Service (Amazon S3)
- Amazon Virtual Private Cloud (Pengintipan VPC)
- Amazon Route 53
- IBM Db2 Pemulihan Bencana Ketersediaan Tinggi (HADR)

Arsitektur target

Arsitektur ini mengimplementasikan solusi DR untuk beban kerja SAP dengan Db2 sebagai platform database. Basis data produksi diterapkan di AWS Region 1 dan database siaga diterapkan di Wilayah kedua. Database siaga disebut sebagai sistem DR. Db2 Database mendukung beberapa database siaga (hingga tiga). Ini menggunakan Db2 HADR untuk menyiapkan database DR dan mengotomatiskan pengiriman log antara basis data produksi dan siaga.

Jika terjadi bencana yang membuat Wilayah 1 tidak tersedia, database siaga di Wilayah DR mengambil alih peran basis data produksi. Server aplikasi SAP dapat dibangun terlebih dahulu atau dengan menggunakan AWS Elastic Disaster Recovery atau Amazon Machine Image (AMI) untuk memenuhi persyaratan recovery time objektif (RTO). Pola ini menggunakan AMI.

Db2 HADR mengimplementasikan pengaturan siaga produksi, di mana produksi bertindak sebagai server utama, dan semua pengguna terhubung dengannya. Semua transaksi ditulis ke file log, yang ditransfer ke server siaga dengan menggunakan TCP/IP. Server siaga memperbarui database lokalnya dengan meneruskan catatan log yang ditransfer, yang membantu memastikan bahwa itu tetap sinkron dengan server produksi.

Pengintip VPC digunakan agar instans di Wilayah produksi dan Wilayah DR dapat berkomunikasi satu sama lain. Amazon Route 53 merutekan pengguna akhir ke aplikasi internet.

1. Buat AMI dari server aplikasi di Wilayah 1 dan salin AMI ke Wilayah 2. Gunakan AMI untuk meluncurkan server di Wilayah 2 jika terjadi bencana.
2. Siapkan replikasi Db2 HADR antara database produksi (di Wilayah 1) dan database siaga (di Wilayah 2).
3. Ubah jenis EC2 instance agar sesuai dengan instance produksi jika terjadi bencana.
4. Di Wilayah 1, `LOGARCHMETH1` diatur ke`db2remote: S3 path`.
5. Di Wilayah 2, `LOGARCHMETH1` diatur ke`db2remote: S3 path`.
6. Replikasi Lintas Wilayah dilakukan antara bucket S3.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan. Pola ini menggunakan VPC peering.

Praktik terbaik

- Jaringan memainkan peran kunci dalam menentukan mode replikasi HADR. Untuk DR di seluruh Wilayah AWS, kami menyarankan Anda menggunakan mode Db2 HADR ASYNC atau SUPERASYNC.
- Untuk informasi selengkapnya tentang mode replikasi untuk Db2 HADR, lihat dokumentasi IBM.
- Anda dapat menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI) Interface (AWS CLI) untuk membuat AMI baru dari sistem SAP yang ada. Anda kemudian dapat menggunakan AMI untuk memulihkan sistem SAP yang ada atau untuk membuat klon.
- AWS Systems Manager Automation dapat membantu tugas pemeliharaan dan penerapan umum EC2 instans dan sumber daya AWS lainnya.
- AWS menyediakan beberapa layanan asli untuk memantau dan mengelola infrastruktur dan aplikasi Anda di AWS. Layanan seperti Amazon CloudWatch dan AWS masing-masing CloudTrail dapat digunakan untuk memantau infrastruktur dan operasi API yang mendasarinya. Untuk detail selengkapnya, lihat SAP on AWS — IBM Db2 HADR with Pacemaker.

Epik

Siapkan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa sistem dan log.	1. Konfirmasikan bahwa SAP produksi pada sistem Db2 sudah diatur. 2. Konfirmasikan bahwa cadangan log diaktifkan dan dikonfigurasi untuk menyimpan log di bucket S3. Ini dapat diperiksa oleh parameter Db2. `LOGARCHMETH1` 3. Buat AMI dari server aplikasi tambahan.	Administrator AWS, administrator SAP Basis

Mengatur server dan replikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server SAP dan database.	1. Untuk menerapkan infrastruktur untuk Wilayah DR, gunakan CloudFormation skrip AWS atau gunakan AMI dari instans produksi. Sebagai bagian dari pendekatan pilot light, Anda dapat menggunakan EC2 instance yang lebih kecil dalam keluarga yang sama dengan instance produksi. Misalnya, jika jenis instance produksi	Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Andar6i.12xlarge , Anda dapat menggunak an tipe r6i.xlarge instance untuk build DR. Namun, pastikan Anda mengalokasikan kapasitas penyimpanan yang sama pada instans DR untuk memulihkan cadangan basis data produksi. 2. Buat titik pemasangan Amazon Elastic File System (Amazon EFS) untuk/ sapmnt/<SID>/ , dan pastikan bahwa itu diatur untuk direplikasi dari sistem utama. 3. Ambil cadangan database LENGKAP (online atau offline) dari sistem produksi. Anda akan menggunak an cadangan ini untuk membangun database DR. 4. Dalam sistem DR, gunakan metode penyalinan sistem SAP Software Provisioning Manager (SWPM) dengan Menggunakan salinan sistem dengan backup/ restore HA/DR tujuan membangun sistem DR SAP. 5. Ketika ditanya oleh SWPM, kembalikan database di	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	DR dengan cadangan yang Anda ambil dari produksi. Database DR akan berada dalam status tertunda rollforward. Status tertunda rollforward diatur secara default setelah cadangan penuh dipulihkan. Status tertunda rollforward menunjukkan bahwa database sedang dalam proses dipulihkan dan bahwa beberapa perubahan mungkin perlu diterapkan. Untuk informasi selengkapnya, lihat dokumentasi IBM.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa konfigurasi.	1. Untuk mengatur pengarsipan log untuk HADR, basis data produksi dan DR harus dapat mengambil log secara otomatis dari semua lokasi arsip log. Verifikasi bahwa `LOGARCHMETH1` parameter dalam database DR diatur ke lokasi yang sama seperti dalam database produksi. Jika lokasi yang sama tidak dapat diakses karena keterbatasan Regional, pastikan bahwa sistem DR dapat secara otomatis mengambil log dari sistem utama. 2. Untuk mengaktifkan TCP/IP port untuk pengaktifan replikasi database, modifikasi `/etc/services` dalam produksi dan host DR dengan menambahkan dua entri berikut. Dalam kode, `<SID>` mengacu pada ID Sistem (SID) dari database Db2 (misalnya,`PR1`). `<SID>_HADR_1 55001/tcp # DB2 HADR Port1`	Administrator AWS, administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<SID>_HADR_2 55002/tcp # DB2 HADR Port2 Konfirmasikan bahwa kedua port memungkinkan lalu lintas masuk dan keluar antara primer dan siaga. 3. Periksa `/etc/hosts` produksi dan host DR untuk mengonfirmasi bahwa nama host untuk host produksi dan siaga menunjuk ke alamat IP yang benar.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan replikasi dari DB produksi ke DR DB (menggunakan mode ASYNC).	1. Dalam database produksi, jalankan perintah berikut untuk memperbarui parameter.
	
	`db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_HOST HOST1`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_SVC <SID>_HADR_1`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_HOST HOST2`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_SVC <SID>_HADR_2`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_INST db2<sid>`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_TIMEOUT 120`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_SYNCMODE ASYNC`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_SPOOL_LIMIT 1000`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_PEER_WINDOW 240`
	`db2 UPDATE DB CFG FOR <SID> USING indexrec RESTART logindexbuild ON`
	
	2. Dalam database DR, jalankan perintah berikut
		Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	untuk memperbarui parameter.
	
	`db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_HOST HOST2`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_SVC <SID>_HADR_2`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_HOST HOST1`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_SVC <SID>_HADR_1`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_INST db2<sid>`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_TIMEOUT 120`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_SYNCMODE ASYNC`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_SPOOL_LIMIT 1000`
	`db2 UPDATE DB CFG FOR <SID> USING HADR_PEER_WINDOW 240`
	`db2 UPDATE DB CFG FOR <SID> USING indexrec RESTART logindexbuild ON`
	
	Parameter ini diperlukan untuk memberikan informasi terkait HADR ke kedua database. Dalam database Db2, HADR	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	diaktifkan berdasarkan nilai untuk masing-masing parameter yang ditetapkan sebelumnya. Untuk informasi selengkapnya tentang parameter ini, lihat dokumentasi IBM.
	3. Mulai HADR pertama pada database siaga yang baru dibuat dengan menggunakan perintah berikut.
	`db2 deactivate db <SID>`
	`db2 start hadr on db <SID> as standby`
	4. Mulai HADR pada database produksi dengan menggunakan perintah berikut.
	`db2 deactivate db <SID>`
	`db2 start hadr on db <SID> as primary`
	5. Periksa apakah database Db2 produksi dan siaga sinkron dan pengiriman log sedang berlangsung.
	Untuk memantau status replikasi HADR, gunakan perintah berikutdb2pd.
	`db2pd -d <SID> -hadr`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang pemantauan HADR, lihat dokumentasi IBM.	

Uji tugas failover DR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rencanakan downtime bisnis produksi untuk tes DR.	Pastikan Anda merencanakan downtime bisnis yang diperlukan pada lingkungan produksi untuk menguji skenario failover DR.	Administrator SAP Basis
Buat pengguna uji.	Buat pengguna uji (atau perubahan pengujian apa pun) yang dapat divalidasi di host DR untuk mengonfirmasi replikasi log setelah DR failover.	Administrator SAP Basis
Di konsol, hentikan EC2 instance produksi.	Shutdown yang tidak teratur dimulai pada langkah ini untuk meniru skenario bencana.	Administrator sistem AWS
Tingkatkan EC2 instance DR agar sesuai dengan persyaratan.	Di EC2 konsol, ubah jenis instance di Wilayah DR. 1. Hentikan instance: Jika instance sedang berjalan, Anda harus menghentikannya sebelum Anda dapat mengubah jenis instance-nya. Di EC2	SAP Dasar Admin

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	konsol, pilih instance, dan pilih Stop.
	2. Ubah jenis instans: Di EC2 konsol, pilih instance, dan pilih Actions, Instance Settings, Change Instance Type. Pilih jenis instance yang cocok dengan instance utama, dan pilih Apply.
	3. Mulai instance: Setelah perubahan jenis instance selesai, mulai instance dari EC2 konsol dengan memilih instance dan memilih Mulai.
	4. Untuk memulai database Db2, gunakan perintah berikut.
	`db2start`
	`db2 start HADR on db <SID> as standby`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Memulai pengambilalihan.	Dari sistem DR (`host2`), memulai proses pengambil alihan dan memunculkan database DR sebagai yang utama.
	`db2 takeover hadr on database <SID> by force`
	Secara opsional, Anda dapat mengatur parameter berikut untuk menyesuaikan alokasi memori database secara otomatis berdasarkan jenis instance. `INSTANCE_MEMORY` Nilai dapat diputuskan berdasarkan porsi memori khusus yang akan dialokasikan ke database Db2.
	`db2 update db cfg for <SID> using INSTANCE_MEMORY <FIXED VALUE> IMMEDIATE;`
	`db2 get db cfg for <SID> | grep -i DATABASE_MEMORY AUTOMATIC IMMEDIATE;`
	`db2 update db cfg for <SID> using self_tuning_mem ON IMMEDIATE;`
	Verifikasi perubahan dengan menggunakan perintah berikut.
		Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`db2 get db cfg for <SID> | grep -i MEMORY` `db2 get db cfg for <SID> | grep -i self_tuning_mem`	
Luncurkan server aplikasi untuk SAP di Wilayah DR.	Menggunakan AMI yang Anda buat dari sistem produksi, luncurkan server aplikasi tambahan baru di Wilayah DR.	Administrator SAP Basis
Lakukan validasi sebelum memulai aplikasi SAP.	1. Validasi `/etc/hosts` dan `/etc/fstab` entri. 2. Pasang `/sapmnt/<SID>/` pada sistem DR. 3. Validasi bahwa sistem file DR `/sapmnt/<SID>/` disinkronkan dengan produksi. `/sapmnt/<SID>/` 4. Masuk ke `<sid>adm` pengguna, jalankan`R3trans -d`, dan verifikasi output dalam `trans.log` file. `trans.log` File dihasilkan di lokasi yang sama di mana Anda menjalankan `R3trans -d` perintah.	Administrator AWS, administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulai aplikasi SAP pada sistem DR.	Mulai aplikasi SAP pada sistem DR dengan menggunakan `<sid>adm` pengguna. Gunakan kode berikut, yang XX mewakili nomor instans server SAP ABAP SAP Central Services (ASCS) Anda, dan YY mewakili nomor instans server aplikasi SAP Anda.
	`sapconrol -nr XX -function StartService <SID>`
	`sapconrol -nr XX -function StartSystem`
	`sapconrol -nr YY -function StartService <SID>`
	`sapconrol -nr YY -function StartSystem`
		Administrator SAP Basis

Lakukan validasi SAP.	Ini dilakukan sebagai tes DR untuk memberikan bukti atau untuk memeriksa keberhasilan replikasi data ke Wilayah DR.	Insinyur uji

Lakukan tugas failback DR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai produksi SAP dan server database.	Di konsol, mulai EC2 instance yang meng-host SAP dan database dalam sistem produksi.	Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulai database produksi dan atur HADR.	Masuk ke sistem produksi (`host1`) dan verifikasi bahwa DB dalam mode pemulihan dengan menggunakan perintah berikut.
	`db2start`
	`db2 start HADR on db P3V as standby`
	`db2 connect to <SID>`
	Verifikasi bahwa status HADR adalah`connected` . Status replikasi seharusnya. `peer`
	`db2pd -d <SID> -hadr`
	Jika database tidak konsisten dan tidak di dan `peer` status, backup `connected` dan restore mungkin diperlukan untuk membawa database (on`host1`) sinkron dengan database yang saat ini aktif (`host2`di Wilayah DR). Dalam hal ini, kembalikan cadangan DB dari database di Wilayah `host2` DR ke database di Wilayah `host1` produksi.
		Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Gagal kembali database ke Wilayah produksi.	Dalam business-as-usual skenario normal, langkah ini dilakukan dalam waktu henti yang dijadwalkan. Aplikasi yang berjalan pada sistem DR dihentikan, dan database gagal kembali ke Wilayah produksi (Wilayah 1) untuk melanjutkan operasi dari Wilayah produksi.
	1. Masuk ke server aplikasi SAP di Wilayah DR, dan hentikan aplikasi SAP.
	2. Lepaskan `/sapmnt/<SID>` dari sistem DR, pastikan bahwa perubahan direplikasi terbalik ke sistem produksi`/sapmnt/<SID>` .
	3. Masuk ke server database (`host1`) di Wilayah produksi, dan lakukan pengambilalihan.
	`db2 takeover hadr on database <SID>`
	4. Periksa status HADR: `HADR_ROLE` harus aktif `PRIMARY` `host1` dan aktif`StandBy`. `host2`
	`db2pd -d <SID> -hadr`
		Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan validasi sebelum memulai aplikasi SAP.	1. Validasi `/etc/hosts` dan `/etc/fstab` entri. 2. Pasang `/sapmnt/<SID>/` pada sistem produksi. 3. Pastikan itu sinkron dengan sistem DR`/sapmnt/<SID>/` . 4. Masuk ke `<sid>adm` pengguna, jalankan`R3trans -d`, dan verifikasi output dalam `trans.log` file. `trans.log` File dihasilkan di lokasi yang sama di mana Anda menjalankan `R3trans -d` perintah.	Administrator AWS, administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulai aplikasi SAP.	1. Mulai aplikasi SAP pada sistem produksi menggunakan `<sid>adm` pengguna. Gunakan kode berikut, yang XX mewakili nomor instans server SAP ASCS Anda, dan YY mewakili nomor instance server aplikasi SAP Anda.
	`sapconrol -nr XX -function StartService <SID>`
	`sapconrol -nr XX -function StartSystem`
	`sapconrol -nr YY -function StartService <SID>`
	`sapconrol -nr YY -function StartSystem`
	2. Untuk mengonfirmasi bahwa server aplikasi tersedia, masuk ke SAP dan lakukan pemeriksaan dengan menggunakan SICK dan SM51 transaksi.
		Administrator SAP Basis

Pemecahan masalah

Isu	Solusi
File log kunci dan perintah untuk memecahkan masalah terkait HADR	• `db2 get db cfg | grep -i hadr` • `db2pd -d sid -hadr`

Isu	Solusi
	• `Db2diag.log` (File ini umumnya terletak di dalam `db2dump` direktori, dan `db2dump` jalur ditentukan oleh parameter`DIAGPATH`.)
Catatan SAP untuk memecahkan masalah HADR di Db2 UDB	Lihat SAP Note 1154013 - DB6: Masalah DB di lingkungan HADR. (Anda memerlukan kredenal portal SAP untuk mengakses catatan ini.)

Sumber daya terkait

- Pendekatan pemulihan bencana untuk database Db2 di AWS (posting blog)
- SAP on AWS — IBM Db2 HADR dengan Pacemaker
- Langkah demi Langkah Prosedur untuk mengatur replikasi HADR antara database DB2
- Db2 HADR Wiki

Informasi tambahan

Dengan menggunakan pola ini, Anda dapat mengatur sistem pemulihan bencana untuk sistem SAP yang berjalan pada database Db2. Dalam situasi bencana, bisnis harus dapat melanjutkan dalam persyaratan tujuan waktu pemulihan yang ditentukan (RTO) dan tujuan titik pemulihan (RPO) Anda:

- RTO adalah penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan. Ini menentukan apa yang dianggap sebagai jendela waktu yang dapat diterima ketika layanan tidak tersedia.
- RPO adalah jumlah waktu maksimum yang dapat diterima sejak titik pemulihan data terakhir. Ini menentukan apa yang dianggap sebagai kehilangan data yang dapat diterima antara titik pemulihan terakhir dan gangguan layanan.

Untuk FAQs yang terkait dengan HADR, lihat catatan SAP #1612105 - DB6: FAQ tentang Db2 High Availability Disaster Recovery (HADR). (Anda memerlukan kredenal portal SAP untuk mengakses catatan ini.)

Siapkan CI/CD pipeline untuk migrasi database dengan menggunakan Terraform

Dr. Rahul Sharad Gaikwad, Ashish Bhatt, Aniket Dekate, Ruchika Modi, Tamilselvan P, Nadeem Rahaman, Aarti Rajput, dan Naveen Suthar, Amazon Web Services

Ringkasan

Pola ini adalah tentang membangun pipeline continuous integration and continuous deployment (CI/CD) untuk mengelola migrasi database dengan cara yang andal dan otomatis. Ini mencakup proses penyediaan infrastruktur yang diperlukan, memigrasi data, dan menyesuaikan perubahan skema dengan menggunakan Terraform, yang merupakan alat infrastruktur sebagai kode (IAc).

Secara khusus, pola menyiapkan CI/CD pipeline untuk memigrasikan database Microsoft SQL Server lokal ke Amazon Relational Database Service (Amazon RDS). AWS Anda juga dapat menggunakan pola ini untuk memigrasikan database SQL Server yang ada di mesin virtual (VM) atau di lingkungan cloud lain ke Amazon RDS.

Pola ini membahas tantangan berikut yang terkait dengan manajemen dan penyebaran basis data:

- Penerapan basis data manual memakan waktu, rawan kesalahan, dan tidak memiliki konsistensi di seluruh lingkungan.
- Mengkoordinasikan penyediaan infrastruktur, migrasi data, dan perubahan skema dapat menjadi kompleks dan sulit dikelola.
- Memastikan integritas data dan meminimalkan waktu henti selama pembaruan basis data sangat penting untuk sistem produksi.

Pola ini memberikan manfaat sebagai berikut:

- Merampingkan proses memperbarui dan menerapkan perubahan database dengan menerapkan CI/CD pipeline untuk migrasi database. Ini mengurangi risiko kesalahan, memastikan konsistensi di seluruh lingkungan, dan meminimalkan waktu henti.
- Membantu meningkatkan keandalan, efisiensi, dan kolaborasi. Memungkinkan waktu yang lebih cepat untuk memasarkan dan mengurangi waktu henti selama pembaruan basis data.
- Membantu Anda mengadopsi DevOps praktik modern untuk manajemen basis data, yang mengarah pada peningkatan kelincahan, keandalan, dan efisiensi dalam proses pengiriman perangkat lunak Anda.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Terraform 0.12 atau yang lebih baru diinstal pada mesin lokal Anda (untuk instruksi, lihat dokumentasi Terraform)
- AWS Penyedia Terraform versi 3.0.0 atau yang lebih baru dari HashiCorp (lihat GitHub repositori untuk penyedia ini)
- Kebijakan hak istimewa paling sedikit AWS Identity and Access Management (IAM) (lihat posting blog Teknik untuk menulis kebijakan IAM hak istimewa terkecil)

Arsitektur

Pola ini mengimplementasikan arsitektur berikut, yang menyediakan infrastruktur lengkap untuk proses migrasi database.

Dalam arsitektur ini:

- Database sumber adalah database SQL Server yang ada di tempat, di mesin virtual (VM), atau dihosting oleh penyedia cloud lain. Diagram mengasumsikan bahwa database sumber berada di pusat data lokal.
- Pusat data lokal dan AWS terhubung melalui VPN atau AWS Direct Connect koneksi. Ini menyediakan komunikasi yang aman antara database sumber dan AWS infrastruktur.
- Basis data target adalah database Amazon RDS yang di-host di dalam virtual private cloud (AWS VPC) dengan bantuan pipeline penyediaan database.
- AWS Database Migration Service (AWS DMS) mereplikasi database lokal Anda ke. AWS Hal ini digunakan untuk mengkonfigurasi replikasi database sumber ke database target.

Diagram berikut menunjukkan infrastruktur yang disiapkan dengan berbagai tingkat proses migrasi database, yang melibatkan penyediaan, AWS DMS penyiapan, dan validasi.

Dalam proses ini:

- Pipeline validasi memvalidasi semua pemeriksaan. Pipeline terintegrasi bergerak ke langkah berikutnya ketika semua validasi yang diperlukan selesai.
- Pipeline penyediaan DB terdiri dari berbagai AWS CodeBuild tahapan yang melakukan tindakan Terraform pada kode Terraform yang disediakan untuk database. Ketika langkah-langkah ini selesai, ia menyebarkan sumber daya di target Akun AWS.
- AWS DMS Pipeline terdiri dari berbagai CodeBuild tahapan yang melakukan pengujian dan kemudian menyediakan AWS DMS infrastruktur untuk melakukan migrasi dengan menggunakan IAc.

Alat

Layanan AWS dan alat-alat

- AWS CodeBuildadalah layanan integrasi berkelanjutan yang dikelola sepenuhnya yang mengkompilasi kode sumber, menjalankan pengujian, dan menghasilkan paket ready-to-deploy perangkat lunak.
- AWS CodePipelineadalah layanan pengiriman berkelanjutan yang dikelola sepenuhnya yang membantu Anda mengotomatiskan saluran pipa rilis Anda untuk pembaruan aplikasi dan infrastruktur yang cepat dan andal.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke dalam AWS Cloud atau di antara kombinasi pengaturan cloud dan lokal.

Layanan lainnya

- Terraform adalah alat IAc HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di DevOps Kerangka Migrasi GitHub Database menggunakan repositori sampel Terraform.

Praktik terbaik

- Terapkan pengujian otomatis untuk migrasi database Anda untuk memverifikasi kebenaran perubahan skema dan integritas data. Ini termasuk pengujian unit, tes integrasi, dan end-to-end tes.
- Terapkan strategi pencadangan dan pemulihan yang kuat untuk database Anda, terutama sebelum migrasi. Ini memastikan integritas data dan menyediakan opsi mundur jika terjadi kegagalan.
- Menerapkan strategi rollback yang kuat untuk mengembalikan perubahan database jika terjadi kegagalan atau masalah selama migrasi. Ini bisa melibatkan memutar kembali ke status database sebelumnya atau mengembalikan skrip migrasi individu.
- Siapkan mekanisme pemantauan dan pencatatan untuk melacak kemajuan dan status migrasi database. Ini membantu Anda mengidentifikasi dan menyelesaikan masalah dengan cepat.

Epik

Siapkan workstation lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan dan konfigurasikan Git di workstation lokal Anda.	Instal dan konfigurasikan Git di workstation lokal Anda dengan mengikuti instruksi dalam dokumentasi Git.	DevOps insinyur
Buat folder proyek dan tambahkan file dari GitHub repositori.	1. Buka GitHub repositori untuk pola ini. 2. Pilih Kode untuk melihat opsi kloning, dan salin URL yang disediakan di tab HTTPS. 3. Buat folder untuk proyek Anda di workstation Anda. 4. Buka terminal dan arahkan ke folder ini. 5. Kloning GitHub repositori:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`git clone <github-repository-url>`
	
	di `<github-repository-url>` mana URL yang Anda salin pada langkah 2.
	6. Saat kloning selesai, buka repositori kloning di folder proyek Anda:
	
	`cd <folder-name>/aws-terraform-db-migration-framework-samples`
	
	7. Buka proyek ini di lingkungan pengembangan terintegrasi (IDE) pilihan Anda.	

Menyediakan arsitektur target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui parameter yang diperlukan.	`ssm-parameters.sh` File menyimpan semua AWS Systems Manager parameter yang diperlukan. Anda dapat mengonfigurasi parameter ini dengan nilai kustom untuk proyek Anda. Di `setup/db-ssm-params` folder di workstation lokal Anda, buka `ssm-parameters.sh` file dan atur	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	parameter ini sebelum Anda menjalankan CI/CD pipeline.	
Inisialisasi konfigurasi Terraform.	Di `db-cicd-integration` folder, masukkan perintah berikut untuk menginisialisasi direktori kerja Anda yang berisi file konfigurasi Terraform: `terraform init`	DevOps insinyur
Pratinjau paket Terraform.	Untuk membuat paket Terraform, masukkan perintah berikut: `terraform plan -var-file="terraform.sample"` Terraform mengevaluasi file konfigurasi untuk menentukan status target untuk sumber daya yang dideklarasikan. Kemudian membandingkan status target dengan keadaan saat ini dan membuat rencana.	DevOps insinyur
Verifikasi rencananya.	Tinjau rencana dan konfirmasikan bahwa itu mengonfigurasi arsitektur yang diperlukan dalam target Akun AWS Anda.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan solusinya.	1. Masukkan perintah berikut untuk menerapkan rencana: `terraform apply -var-file="terraform.sample"` 2. Masukkan `yes` untuk mengonfirmasi. Terraform membuat, memperbarui, atau menghancurkan infrastruktur untuk mencapai status target yang dinyatakan dalam file konfigurasi. Untuk informasi lebih lanjut tentang urutan, lihat bagian Arsitektur dari pola ini.	DevOps insinyur

Verifikasi penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan.	Verifikasi status `db-cicd-integration` pipeline untuk mengonfirmasi bahwa migrasi database telah selesai. 1. Masuk ke Konsol Manajemen AWS, lalu buka AWS CodePipeline konsol. 2. Di panel navigasi, pilih Pipelines.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pilih `db-cicd-integration` pipa. 4. Validasi bahwa eksekusi pipeline telah selesai dengan sukses.	

Membersihkan infrastruktur setelah digunakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan infrastruktur.	1. Setelah proyek Anda selesai, bersihkan infrastruktur yang Anda buat dengan menggunakan perintah: `terraform destroy --var-file=terraform.sample` 2. Masukkan `yes` untuk mengonfirmasi.	DevOps insinyur

Sumber daya terkait

AWS dokumentasi

- Memulai dengan produk Terraform

Dokumentasi Terraform

- Instalasi Terraform
- Konfigurasi backend Terraform
- Dokumentasi Penyedia Terraform AWS

Mengatur cluster failover Microsoft SQL Server di Amazon EC2 menggunakan Windows FSx File Server

Sweta Krishna dan Ramesh Babu Donti, Amazon Web Services

Ringkasan

Microsoft SQL Server Standard edition dengan instance failover cluster (FCI) dapat memberikan alternatif yang lebih hemat biaya untuk SQL Server Enterprise. Menyiapkan SQL FCI memerlukan penyimpanan file bersama antar node, dan Amazon FSx untuk Windows File Server menyediakan penyimpanan terkelola penuh yang secara otomatis mereplikasi secara sinkron di seluruh Availability Zone. Amazon FSx mengurangi biaya penyimpanan dengan menggunakan deduplikasi data bawaan untuk berbagi file tujuan umum, yang menghilangkan kebutuhan untuk mempertahankan solusi pihak ketiga. Amazon FSx juga mendukung hal-hal berikut:

- Bayar hanya untuk apa yang Anda gunakan tanpa biaya atau komitmen di muka.
- Siapkan FCI secara manual dengan FSx penyimpanan bersama Anda.
- Gunakan FSx sebagai saksi berbagi file untuk cluster SQL Anda.
- Amazon FSx untuk Windows File Server mendukung Server Message Block (SMB) 3.0 untuk berbagi file yang tersedia secara terus menerus, sehingga cocok untuk penerapan SQL Server FCI.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Izin untuk membuat dan mengelola resource Amazon Virtual Private Cloud (Amazon VPC), instans Amazon Elastic Compute Cloud (EC2Amazon), grup keamanan, AWS Identity and Access Management dan peran (IAM).
- AWS Managed Microsoft AD atau Direktori Aktif lokal Anda sendiri.
- Pengguna domain Active Directory dengan izin yang diperlukan untuk menyiapkan cluster failover.
- Aturan grup keamanan untuk port SQL Server FCI dan Microsoft Active Directory untuk konektivitas hybrid yang aman.
- Akun layanan di Active Directory untuk SQL Server yang dikonfigurasi dengan izin yang sesuai di seluruh node SQL.

- Amazon FSx untuk Windows File Server dalam cluster failover.
- Binari instalasi SQL Server.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

- Amazon EC2 untuk Windows Server 2012 R2 atau yang lebih baru
- Amazon FSx untuk Windows File Server dengan semua versi Windows Server saat ini
- Amazon FSx untuk NetApp ONTAP sebagai alternatif untuk penyimpanan bersama
- SQL Server 2012/2016/2019/2022

Arsitektur

Tumpukan teknologi

- Amazon EC2
- Amazon FSx untuk Server File Windows
- Amazon VPC
- AWS Directory Service
- AWS Systems Manager
- IAM

Arsitektur target

Diagram berikut menunjukkan arsitektur tingkat tinggi Microsoft SQL Server FCI di Amazon menggunakan EC2 Amazon FSx untuk Windows File Server.

Infrastruktur jaringan

- Amazon VPC menyediakan wadah jaringan yang mencakup tiga Availability Zone.

- Subnet pribadi menyediakan subnet terisolasi di setiap Availability Zone untuk menyebarkan sumber daya.

Lapisan komputasi

- Amazon EC2 berisi node cluster SQL Server 1, digunakan di Availability Zone 1 sebagai bagian dari Windows Server Failover Cluster (WSFC).
- Amazon EC2 berisi node cluster SQL Server 2, digunakan di Availability Zone 2 sebagai bagian dari WSFC.
- Cluster WSFC menghubungkan kedua node SQL Server untuk kemampuan failover.

Lapisan penyimpanan untuk Amazon FSx untuk Windows File Server

FSx Penerapan multi-AZ (mencakup Availability Zone 1 dan 2)

- Sistem FSx file utama di Availability Zone 1 menghosting data SQL Server aktif dan file log.
- Sistem FSx file sekunder di Availability Zone 2 menyediakan kemampuan failover otomatis.
- Berbagi file SMB bersama (\\ fsx.domain\ sqlshare), dapat diakses oleh kedua node cluster untuk database SQL Server.

FSx Penyebaran AZ tunggal (in) AZ3

- Saksi server FSx file Amazon di Availability Zone 3 berfungsi sebagai saksi kuorum cluster.
- File share witness (`\\fsx.domain\witness`) mempertahankan kuorum cluster dan mencegah skenario split-brain.

Layanan direktori

- AWS Managed Microsoft AD menyediakan otentikasi Windows dan layanan domain yang diperlukan untuk fungsionalitas cluster.

Fitur ketersediaan tinggi

- Komponen multi-AZ memberikan toleransi kesalahan di seluruh Availability Zone.
- FSx server file siaga menyediakan failover otomatis jika server utama gagal.

- Saksi berbagi file menyediakan manajemen kuorum klaster di Availability Zone 3 untuk membantu memastikan operasi klaster yang tepat selama kegagalan.
- Domain terintegrasi dengan AWS Managed Microsoft AD otentikasi Windows yang mulus.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon FSx menyediakan sistem file yang mendukung protokol konektivitas standar industri dan menawarkan ketersediaan dan replikasi yang tinggi. Wilayah AWS
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS
- AWS Directory Service for Microsoft Active Directorymemungkinkan beban kerja dan sumber daya yang sadar direktori Anda AWS untuk menggunakan Microsoft Active Directory di file. AWS Cloud
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar.

Praktik terbaik

- Tempatkan instance database di subnet pribadi untuk melindunginya agar tidak dapat diakses publik dari internet sambil tetap memungkinkan mereka untuk terhubung Layanan AWS dan melakukan pembaruan.
- Untuk praktik terbaik umum, lihat Praktik Terbaik untuk Menerapkan Microsoft SQL Server di Amazon. EC2

- Untuk menggunakan PowerShell untuk mengelola Amazon FSx untuk Windows File Server, lihat Mengelola FSx untuk sistem file Windows.

Epik

Membuat dan mengkonfigurasi EC2 node Amazon untuk SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan nama dan tag.	1. Di Konsol Manajemen AWS, navigasikan ke Amazon EC2. 2. Pilih Luncurkan instans. 3. Tambahkan tag yang diperlukan.	DBA
Pilih AMI Windows.	Pilih Amazon Machine Image (AMI) untuk Windows yang memenuhi persyaratan SQL Server Anda.	DBA
Pilih jenis instance.	Pilih jenis EC2 instans Amazon yang memenuhi persyaratan Anda.	DBA
Gunakan key pair.	Anda dapat menggunakan key pair untuk terhubung dengan aman ke instans Anda. Pastikan Anda memiliki akses ke key pair yang dipilih sebelum meluncurkan instance.	DBA
Mengkonfigurasi pengaturan jaringan.	1. Pilih VPC yang Anda gunakan untuk mengkonfi gurasi Active Directory.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pilih VPC dan subnet pribadi. 3. Pilih grup keamanan yang ada. Jika Anda tidak memilikinya, pilih Buat grup keamanan.	
Konfigurasikan pengaturan jaringan lanjutan.	1. Pilih Pengaturan jaringan lanjutan. 2. Pilih IP sekunder. 3. Pilih kotak centang Tetapkan secara otomatis untuk menetapkan IP sekunder dari subnet.	DBA
Konfigurasikan penyimpanan.	Konfigurasikan total penyimpanan yang diperlukan dan pilih jenis penyimpanan yang diperlukan.	DBA
Konfigurasikan detail lanjutan dan luncurkan instance.	1. Pilih domain Active Directory dari daftar direktori. 2. Pilih peran IAM yang memiliki `AmazonSSMManagedInstanceCore` dan `AmazonSSMDirectoryServiceAccess` kebijakan terlampir. 3. (Opsional) Konfigurasikan opsi lain yang tersedia. 4. Luncurkan instance.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat simpul 2.	Ulangi langkah-langkah ini untuk membuat dan mengkonfigurasi node 2.	DBA

Instal dan konfigurasikan cluster failover Windows Server pada node 1 dan 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke node 1.	Masuk ke EC2 instance Windows Amazon sebagai administrator.	DBA
Instal fitur FCI pada node 1.	1. Dalam PowerShell, jalankan skrip berikut: `Install-WindowsFeature -Name Failover-Clustering -IncludeManagementTools` 1. Setelah instalasi, periksa dan instal pembaruan Windows yang luar biasa. 2. Mulai ulang instans.	DBA
Masuk ke node 2.	Masuk ke EC2 instance Windows Amazon sebagai administrator.	DBA
Instal fitur FCI pada node 2.	1. Dalam PowerShell, jalankan skrip berikut: `Install-WindowsFeature -Name Failover-`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Clustering -IncludeM anagementTools` 1. Setelah instalasi, periksa dan instal pembaruan Windows yang luar biasa. 2. Mulai ulang instans.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan node ke cluster.	1. Gunakan Remote Desktop untuk terhubung ke node 1 dan pilih Mulai manajer server. 2. Di menu Tools, pilih Failover Cluster Manager. 3. Pada panel pengelola klaster Failover di bawah Manajemen, pilih Buat klaster untuk membuka wizard Create cluster. 4. Pada halaman Before you begin, pilih Next. 5. Jika halaman Pilih server muncul, di bidang Masukkan nama, pilih Jelajahi dan cari server Anda, lalu pilih Tambah. Ulangi langkah ini untuk setiap server yang ingin Anda tambahkan dan kemudian pilih Berikutnya. 6. Pada Access point untuk mengelola halaman cluster, pilih cluster dan kemudian pilih Berikutnya. 7. Halaman Konfirmasi menunjukkan nama cluster, node, dan informasi domain lainnya. Tinjau detail ini dan kemudian pilih Berikutnya. 8. Pada halaman Ringkasan , verifikasi bahwa FCI	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	berhasil dibuat. Jika ada peringatan atau kesalahan , lihat output ringkasan atau pilih Lihat laporan. 9. Pilih Selesai. 10. Ketika cluster dikonfigurasi, sistem akan ditampilkan Anda telah menyelesaikan wizard membuat cluster. 11. Ulangi langkah-langkah ini untuk node 2.	
Bawa cluster online.	Untuk membuat cluster online, perbarui alamat IP statis dari kedua node: 1. Pilih nama objek cluster. 2. Buka menu konteks (klik kanan) dan pilih Properties. 3. Pilih subnet dan alamat IP sekunder yang tidak digunakan dari instans Anda EC2 . 4. Setelah memperbarui alamat IP sekunder, buka menu konteks (klik kanan) dan pilih Nama objek cluster. 5. Pilih Bawa online.	DBA
Validasi cluster.	Arahkan ke pengelola klaster Failover dan verifikasi bahwa sumber daya inti klaster sedang online.	DBA

Instal SQL Server pada node 1 dan 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke server.	Masuk ke EC2 instans Amazon sebagai administr ator.	DBA
Pasang binari SQL.	1. Mulai pusat instalasi SQL Server dan kemudian pilih Instalasi. 2. Pilih Instalasi cluster failover SQL Server Baru. 3. Ikuti petunjuknya hingga Setup file. 4. Di kotak dialog Aturan global, verifikasi bahwa semua pemeriksaan berhasil. 5. Di kotak dialog Instal aturan klaster failover, verifikasi bahwa semua pemeriksaan berhasil. Atasi peringatan apa pun sebelum melanjutk an. 6. Dalam kotak dialog Pemilihan fitur, pilih kotak centang untuk layanan mesin Database dan konektivitas alat klien. 7. Di kotak dialog Aturan fitur, verifikasi bahwa semua aturan disahkan. 8. Dalam kotak dialog konfigurasi Instans,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	masukkan nama jaringan SQL Server. 9. Dalam kelompok sumber daya Cluster, tentukan nama untuk grup sumber daya. Pilih grup dari daftar dropdown atau masukkan nama khusus. 10. Di kotak dialog Konfigurasi jaringan cluster, pilih IPv4dan hapus DHCP. Berikan alamat IP pribadi sekunder untuk setiap node. 11. Di kotak dialog konfigurasi Server: • Pada tab Akun layanan, berikan rincian yang diperlukan untuk nama akun Layanan SQL dan kredensional untuk SQL Server. • Verifikasi bahwa jenis Startup untuk agen dan mesin database diatur ke Manual. • Pilih kotak centang untuk Grant melakukan hak istimewa tugas pemeliharaan volume ke layanan mesin database SQL Server. Hal ini memungkinkan inisialis	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	asi file instan untuk SQL Server. 12. Dalam kotak dialog konfigurasi mesin Database: • Pada tab konfigurasi Server di bawah mode otentikasi Windows, verifikasi bahwa mode Otentikasi dipilih. • Pada tab Direktori data, tentukan lokasi FSx jalur Amazon untuk data, log, tempdb, dan file cadangan (misalnya, untuk `.mdf` file dan `\\fsx.domain\sqlshare\data` `\\fsx.domain\sqlshare\logs` untuk `.ldf` file). 13. Dalam kotak dialog Aturan konfigurasi fitur, verifikasi bahwa semua pemeriksaan berhasil dan kemudian pilih Berikutnya. 14. Dalam kotak dialog Siap untuk menginstal, verifikasi semua pengaturan konfigurasi. 15. Pilih Instal untuk melanjutkan penginstalan. 16. Dalam kotak dialog Lengkap, pilih Tutup.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan node 2 ke cluster failover.	1. Arahkan ke pusat instalasi SQL Server. 2. Pilih Instalasi. 3. Pilih Tambahkan node ke cluster failover SQL Server dan ikuti petunjuk hingga konfigurasi node Cluster. 4. Dalam kotak dialog Konfigurasi simpul cluster, centang IPv4dan hapus DHCP. 5. Tambahkan alamat IP untuk node 2. 6. Di bawah akun Layanan, verifikasi bahwa detailnya cocok untuk node 1 dan 2. 7. Berikan kredensil untuk akun layanan SQL Server yang sesuai. 8. Di kotak dialog Aturan fitur, verifikasi bahwa semua pemeriksaan berhasil. 9. Pilih Berikutnya dan ikuti petunjuk yang tersisa. 10. Pilih Instal untuk melanjutkan penginstalan. 11. Dalam kotak dialog Lengkap, pilih Tutup.	DBA

Konfigurasikan saksi berbagi FSx file Amazon

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan kuorum.	1. Gunakan RDP untuk terhubung ke instans Amazon EC2 Anda. 2. Arahkan ke pengelola cluster Failover. 3. Buka menu konteks (klik kanan) pada cluster dan pilih Tindakan lainnya. 4. Pilih Konfigurasikan pengaturan kuorum cluster. 5. Konfigurasikan pengaturan kuorum.	DBA
Ambil detail DNS.	Di FSx konsol Amazon, pilih Managed AD lalu Lampirkan. DNS harus memiliki format berikut: `\\example.example.net\share`	DBA
Konfigurasikan saksi berbagi file.	Pilih jalur berbagi FSx file Amazon dan kemudian Selesai.	DBA

Sumber daya terkait

AWS sumber daya

- Amazon FSx untuk Server File Windows (video)
- Menyelam jauh di Amazon FSx untuk Windows File Server (video)
- Cara menyebarkan cluster failover SQL Server dengan Amazon EBS Multi-Attach di Windows Server (posting blog)AWS

- Sederhanakan penerapan ketersediaan tinggi Microsoft SQL Server Anda menggunakan FSx Amazon untuk Windows File Server AWS (posting blog)
- Penerapan ketersediaan tinggi SQL Server menggunakan Amazon FSx untuk NetApp ONTAP (posting blog)AWS
- Menggunakan FSx untuk Windows File Server dengan Microsoft SQL Server
- Apa itu FSx untuk Windows File Server?

Sumber daya lainnya

- Buat cluster failover

Informasi tambahan

Mengkonfigurasi saksi berbagi file

Pastikan Anda terhubung ke sistem file dari kedua node dengan menambahkan aturan di grup FSx keamanan Amazon yang memungkinkan koneksi masuk. Port SMB harus diizinkan. Misalnya, jika nama DNS adalah`\\example.example.com\share`, gunakan`\\example.example.com\share`. Gunakan nilai yang sama untuk saksi berbagi file di klaster ketersediaan Selalu Aktif. Selesaikan langkah-langkah berikut untuk mengonfigurasi saksi berbagi file:

1. Gunakan RDP untuk terhubung ke instans Amazon EC2 Anda.
2. Arahkan ke pengelola cluster Failover.
3. Buka menu konteks (klik kanan) dan pilih Tindakan lainnya.
4. Pilih Konfigurasikan pengaturan kuorum cluster.
5. Pilih Berikutnya.
6. Pilih Konfigurasi kuorum dan konfigurasikan saksi berbagi file.
7. Berikan nama DNS.
8. Tinjau ringkasan dan kemudian pilih Selesai. Saksi berbagi file harus online di bagian sumber daya inti Cluster.

Menyiapkan HA/DR arsitektur untuk Oracle E-Business Suite di Amazon RDS Custom dengan database siaga aktif

Simon Cunningham, Jaydeep Nandy, dan Nitin Saxena, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana Anda dapat merancang solusi Oracle E-Business Anda di Amazon Relational Database Service (Amazon RDS) Kustom untuk ketersediaan tinggi (HA) dan pemulihan bencana (DR) dengan menyiapkan database replika baca Amazon RDS Custom di Zona Ketersediaan Amazon Web Services (AWS) lainnya dan mengubahnya menjadi database siaga aktif. Pembuatan replika baca Kustom Amazon RDS sepenuhnya otomatis melalui AWS Management Console.

Pola ini tidak membahas langkah-langkah untuk menambahkan tingkatan aplikasi tambahan dan sistem file bersama, yang juga dapat menjadi bagian dari HA/DR arsitektur. Untuk informasi selengkapnya tentang topik tersebut, lihat Catatan Dukungan Oracle berikut: 1375769.1, 1375670.1, dan 1383621.1 (bagian 5, Opsi Kloning Lanjutan). (Akses memerlukan akun Dukungan Oracle.)

Untuk memigrasikan sistem E-Business Suite ke arsitektur single-tier, Single-AZ di Amazon Web Services (AWS), lihat pola Migrate Oracle E-Business Suite ke Amazon RDS Custom.

Oracle E-Business Suite adalah solusi Enterprise Resource Planning (ERP) untuk mengotomatisasi proses di seluruh perusahaan seperti keuangan, sumber daya manusia, rantai pasokan, dan manufaktur. Ini memiliki arsitektur tiga tingkat: klien, aplikasi, dan database. Sebelumnya, Anda harus menjalankan database E-Business Suite pada instans Amazon Elastic Compute Cloud (Amazon EC2) yang dikelola sendiri, tetapi sekarang Anda dapat memanfaatkan Amazon RDS Custom.

Prasyarat dan batasan

Prasyarat

- Instalasi E-Business Suite yang ada di Amazon RDS Custom; lihat pola Migrasi Oracle E-Business Suite ke Amazon RDS Custom
- Jika Anda ingin mengubah replika baca menjadi read-only dan menggunakannya untuk menurunkan pelaporan ke standby, lisensi database Oracle Active Data Guard (lihat Daftar Harga Komersil Teknologi Oracle)

Batasan

- Batasan dan konfigurasi yang tidak didukung untuk database Oracle di Amazon RDS Custom
- Keterbatasan yang terkait dengan Amazon RDS Custom untuk replika baca Oracle

Versi produk

Untuk versi Oracle Database dan kelas instans yang didukung oleh Amazon RDS Custom, lihat Persyaratan dan batasan untuk Amazon RDS Custom for Oracle.

Arsitektur

Diagram berikut menggambarkan arsitektur representatif untuk E-Business Suite di AWS yang mencakup beberapa Availability Zone dan tingkatan aplikasi dalam penyiapan. active/passive Basis data menggunakan instans Amazon RDS Custom DB dan replika baca Amazon RDS Custom. Replika baca menggunakan Active Data Guard untuk mereplikasi ke Availability Zone lainnya. Anda juga dapat menggunakan replika baca untuk membongkar lalu lintas baca di database utama dan untuk tujuan pelaporan.

Untuk informasi selengkapnya, lihat Bekerja dengan replika baca untuk Amazon RDS Custom for Oracle dalam dokumentasi Amazon RDS.

Replika baca Kustom Amazon RDS dibuat secara default saat dipasang. Namun, jika Anda ingin menurunkan beberapa beban kerja hanya-baca ke database siaga untuk mengurangi beban pada database utama Anda, Anda dapat secara manual mengubah mode replika yang dipasang menjadi hanya-baca dengan mengikuti langkah-langkah di bagian Epics. Kasus penggunaan umum untuk ini adalah menjalankan laporan Anda dari database siaga. Mengubah ke read-only memerlukan lisensi database siaga aktif.

Saat Anda membuat replika baca di AWS, sistem menggunakan broker Oracle Data Guard di bawah sampulnya. Konfigurasi ini dibuat secara otomatis dan diatur dalam mode Kinerja Maksimum sebagai berikut:

```
DGMGRL> show configuration
Configuration - rds_dg
  Protection Mode: MaxPerformance
  Members:
  vis_a - Primary database
    vis_b - Physical standby database
Fast-Start Failover: DISABLED
```

```
Configuration Status:
SUCCESS   (status updated 58 seconds ago)
```

Alat

Layanan AWS

- Amazon RDS Custom for Oracle adalah layanan database terkelola untuk aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi dan lingkungan database yang mendasarinya. Ini mengotomatiskan tugas dan operasi administrasi database sambil memungkinkan Anda, sebagai administrator database, untuk mengakses dan menyesuaikan lingkungan database dan sistem operasi Anda.

Alat lainnya

- Oracle Data Guard adalah alat yang membantu Anda membuat dan mengelola database siaga Oracle. Pola ini menggunakan Oracle Data Guard untuk menyiapkan database siaga aktif di Amazon RDS Custom.

Epik

Membuat replika baca

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat replika baca instans Amazon RDS Custom DB.	Untuk membuat replika baca, ikuti petunjuk dalam dokumentasi Amazon RDS dan gunakan instans Amazon RDS Custom DB yang Anda buat (lihat bagian Prasyarat) sebagai database sumber. Secara default, replika baca Kustom Amazon RDS dibuat sebagai siaga fisik dan dalam keadaan terpasang. Ini disengaja untuk memastika	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n kepatuhan dengan lisensi Oracle Active Data Guard. Ikuti langkah selanjutnya untuk mengonversi replika baca ke mode hanya-baca.	

Ubah replika baca menjadi siaga aktif hanya-baca

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Connect ke Amazon RDS Custom read replika.	Gunakan perintah berikut untuk mengonversi database siaga fisik Anda ke database siaga aktif.
	
	Important
	Perintah ini memerlukan lisensi siaga aktif Oracle. Untuk mendapatk an lisensi, hubungi perwakilan Oracle Anda.
	
	`$ sudo su - rdsdb`
	`-bash-4.2$ sql`
	`SQL> select process,s`
	`tatus,sequence# from v`
	`$managed_standby;`
	
	`PROCESS STATUS`
	`SEQUENCE#`
	`--------- ------------`
	`----------`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>ARCH CLOSING
	3956
	ARCH CONNECTED
	0
	ARCH CLOSING
	3955
	ARCH CLOSING
	3957
	RFS IDLE
	0
	RFS IDLE
	3958
	MRP0 APPLYING_LOG
	3958
	SQL> select name,
	database_role,
	open_mode from v
	$database;
	
	NAME DATABASE_
	ROLE OPEN_MODE
	--------- ---------
	------- ---------
	----------- 	
	VIS PHYSICAL
	STANDBY MOUNTED
	SQL> alter database
	recover managed standby
	database cancel;
	Database altered.
	Open the standby
	database
	SQL> alter database
	open;
	Database altered.
	SQL> select name,
	database_role,
	open_mode from v
	$database;
	
	NAME DATABASE_
	ROLE OPEN_MODE</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--------- --------- ------- --------- ----------- VIS PHYSICAL STANDBY READ ONLY`	
Mulai pemulihan media dengan log real-time berlaku.	Untuk mengaktifkan fitur penerapan log waktu nyata, gunakan perintah berikut. Ini mengonversi dan memvalida si siaga (replika baca) sebagai database siaga aktif, sehingga Anda dapat menghubungkan dan menjalankan kueri hanya-baca. `SQL> alter database recover managed standby database using current logfile disconnect from session; Database altered`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa status database.	Untuk memeriksa status database, gunakan perintah berikut.
	`SQL> select name, database_role, open_mode from v$database;`
	`NAME DATABASE_ROLE OPEN_MODE`
	`--------- ---------- ---------- VIS PHYSICAL STANDBY READ ONLY WITH APPLY`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa mode redo apply.	Untuk memeriksa redo apply mode, gunakan perintah berikut.
	
	`SQL> select process,s tatus,sequence# from v $managed_standby;`
	`PROCESS STATUS SEQUENCE#`
	`--------- ------------ ----------`
	`ARCH CLOSING 3956`
	`ARCH CONNECTED 0`
	`ARCH CLOSING 3955`
	`ARCH CLOSING 3957`
	`RFS IDLE 0`
	`RFS IDLE 3958`
	`MRP0 APPLYING_LOG 3958`
	
	`SQL> select open_mode from v$database;`
	`OPEN_MODE`
	`--------------------`
	`READ ONLY WITH APPLY`
		DBA

Sumber daya terkait

- Migrasikan Oracle E-Business Suite ke Amazon RDS Custom (AWS Prescriptive Guidance)
- Bekerja dengan Amazon RDS Custom (dokumentasi Amazon RDS)
- Bekerja dengan replika baca untuk Amazon RDS Custom untuk Oracle (dokumentasi Amazon RDS)

- Amazon RDS Kustom untuk Oracle - Kemampuan Kontrol Baru di Lingkungan Database (blog AWS News)
- Memigrasi Oracle E-Business Suite di AWS (whitepaper AWS)
- Arsitektur Oracle E-Business Suite di AWS (whitepaper AWS)

Streaming data dari IBM Db2, SAP, Sybase, dan database lainnya ke MongoDB Atlas AWS

Battulga Purevragchaa dan Igor Alekseev, Amazon Web Services

Babu Srinivasan, MongoDB

Ringkasan

Pola ini menjelaskan langkah-langkah untuk memigrasikan data dari IBM Db2 dan database lain seperti database mainframe dan Sybase ke MongoDB Atlas di file. AWS Cloud Ini digunakan AWS Glueuntuk membantu mempercepat migrasi data ke MongoDB Atlas.

Pola ini menyertai panduan Migrasi ke MongoDB Atlas di situs web Prescriptive Guidance. AWS AWS Ini memberikan langkah-langkah implementasi untuk salah satu skenario migrasi yang dibahas dalam panduan itu. Untuk skenario migrasi tambahan, lihat pola berikut di situs web Panduan AWS Preskriptif:

- Migrasikan lingkungan MongoDB yang dihosting sendiri ke MongoDB Atlas di AWS
- Migrasikan database relasional ke MongoDB Atlas di AWS

Pola ini ditujukan untuk AWS Managed Services Partners dan AWS pengguna.

Prasyarat dan batasan

Prasyarat

- Database sumber seperti SAP, Sybase, IBM Db2, dan lainnya untuk bermigrasi ke MongoDB Atlas.
- Keakraban dengan database seperti SAP, Sybase, IBM Db2, MongoDB Atlas, dan. Layanan AWS

Versi produk

- MongoDB versi 5.0 atau yang lebih baru.

Arsitektur

Diagram berikut menggambarkan pemuatan data batch dan streaming data dengan menggunakan AWS Glue Studio, Amazon Kinesis Data Streams, dan MongoDB Atlas.

Arsitektur referensi ini digunakan AWS Glue Studio untuk membuat pipeline ekstrak, transformasi, dan beban (ETL) untuk memigrasikan data ke MongoDB Atlas. An Perayap AWS Glue terintegrasi dengan MongoDB Atlas untuk memfasilitasi tata kelola data. Data dapat di-porting dalam batch atau dialirkan ke MongoDB Atlas dengan menggunakan Amazon Kinesis Data Streams.

Pemuatan data batch

Untuk informasi selengkapnya tentang migrasi data batch, lihat posting AWS blog Menulis pekerjaan ETL Anda untuk MongoDB Atlas dengan. AWS Glue

Streaming data

Untuk arsitektur referensi MongoDB Atlas yang mendukung skenario penggunaan yang berbeda, lihat Migrasi ke MongoDB Atlas di situs web Panduan Preskriptif. AWS AWS

Alat

- AWS Glueadalah layanan ETL yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- Amazon Kinesis Data Streams membantu Anda mengumpulkan dan memproses aliran besar catatan data secara real time.
- MongoDB Atlas adalah database yang dikelola sepenuhnya sebagai layanan (dBaaS) untuk menyebarkan dan mengelola database MongoDB di cloud.

Praktik terbaik

Untuk panduan, lihat Panduan Praktik Terbaik untuk MongoDB di repositori MongoDB. GitHub

Epik

Penemuan dan penilaian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan ukuran cluster.	Perkirakan ukuran set kerja dengan menggunakan	MongoDB DBA, Arsitek aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi dari `db.stats( )` untuk total ruang indeks. Asumsikan bahwa persentase ruang data Anda akan sering diakses. Atau, Anda dapat memperkirakan kebutuhan memori Anda berdasarkan asumsi Anda. Tugas ini harus memakan waktu sekitar satu minggu. Untuk informasi lebih lanjut dan contoh untuk ini dan cerita lainnya dalam epik ini, lihat tautan di bagian Sumber daya terkait.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perkirakan kebutuhan bandwidth jaringan.	Untuk memperkirakan kebutuhan bandwidth jaringan Anda, kalikan ukuran dokumen rata-rata dengan jumlah dokumen yang disajikan per detik. Pertimbangkan lalu lintas maksimum yang akan ditanggung oleh node mana pun di cluster Anda sebagai dasarnya. Untuk menghitung kecepatan transfer data hilir dari klaster Anda ke aplikasi klien, gunakan jumlah total dokumen yang dikembalikan selama periode waktu tertentu. Jika aplikasi Anda membaca dari node sekunder, bagilah jumlah total dokumen ini dengan jumlah node yang dapat melayani operasi baca. Untuk menemukan ukuran dokumen rata-rata untuk database, gunakan `db.stats().avgObjSize` perintah. Tugas ini biasanya akan memakan waktu satu hari.	MongoDB DBA
Pilih tingkat Atlas.	Ikuti petunjuk dalam dokumentasi MongoDB untuk memilih tingkat cluster Atlas yang benar.	MongoDB DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rencanakan untuk cutover.	Rencanakan pemotongan aplikasi.	MongoDB DBA, Arsitek aplikasi

Siapkan lingkungan MongoDB Atlas baru di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster MongoDB Atlas baru di. AWS	Di MongoDB Atlas, pilih Build a Cluster, dan AWS pilih sebagai penyedia cloud.	MongoDB DBA
Pilih Wilayah AWS dan konfigurasi cluster global.	Pilih dari daftar yang tersedia Wilayah AWS untuk cluster Atlas Anda. Konfigurasikan cluster global jika diperlukan.	MongoDB DBA
Pilih tingkat cluster.	Pilih tingkat klaster pilihan Anda. Pemilihan tingkat Anda menentukan faktor-faktor seperti memori, penyimpanan, dan spesifikasi IOPS.	MongoDB DBA
Konfigurasikan pengaturan cluster tambahan.	Konfigurasikan pengaturan cluster tambahan seperti versi MongoDB, cadangan, dan opsi enkripsi. Untuk informasi selengkapnya tentang opsi ini, lihat bagian Sumber daya terkait.	MongoDB DBA

Konfigurasikan keamanan dan kepatuhan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan daftar akses.	Untuk terhubung ke cluster Atlas, Anda harus menambahkan entri ke daftar akses proyek. Atlas menggunakan Transport Layer Security (TLS) /Secure Sockets Layer (SSL) untuk mengenkripsi koneksi ke virtual private cloud (VPC) untuk database Anda. Untuk mengatur daftar akses untuk proyek dan untuk informasi lebih lanjut tentang cerita dalam epik ini, lihat tautan di bagian Sumber daya terkait.	MongoDB DBA
Otentikasi dan otorisasi pengguna.	Anda harus membuat dan mengotentikasi pengguna database yang akan mengakses cluster MongoDB Atlas. Untuk mengakses cluster dalam proyek, pengguna harus menjadi bagian dari proyek itu, dan mereka dapat menjadi bagian dari beberapa proyek. Anda juga dapat mengaktifkan otorisasi dengan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat Mengatur Otentikasi dengan IAM di dokumentasi MongoDB.	MongoDB DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran khusus.	(Opsional) Atlas mendukung pembuatan peran khusus jika hak istimewa pengguna database Atlas bawaan tidak mencakup rangkaian hak istimewa yang Anda inginkan.	MongoDB DBA
Siapkan pengintip VPC.	(Opsional) Atlas mendukung peering VPC dengan AWS lainnya. VPCs	MongoDB DBA
Siapkan AWS PrivateLink titik akhir.	(Opsional) Anda dapat mengatur titik akhir pribadi AWS dengan menggunakan AWS PrivateLink.	MongoDB DBA
Aktifkan otentikasi dua faktor.	(Opsional) Atlas mendukung otentikasi dua faktor (2FA) untuk membantu pengguna mengontrol akses ke akun Atlas mereka.	MongoDB DBA
Siapkan otentikasi dan otorisasi pengguna dengan LDAP.	(Opsional) Atlas mendukung melakukan otentikasi dan otorisasi pengguna dengan Lightweight Directory Access Protocol (LDAP).	MongoDB DBA
Siapkan AWS akses terpadu.	(Opsional) Beberapa fitur Atlas, termasuk Atlas Data Lake dan enkripsi saat istirahat menggunakan manajemen kunci pelanggan , menggunakan peran IAM untuk otentikasi.	MongoDB DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan enkripsi saat istirahat dengan menggunakan AWS KMS.	(Opsional) Atlas mendukung penggunaan AWS Key Management Service (AWS KMS) untuk mengenkripsi mesin penyimpanan dan cadangan penyedia cloud.	MongoDB DBA
Mengatur CSFLE.	(Opsional) Atlas mendukung enkripsi tingkat bidang sisi klien (CSFLE), termasuk enkripsi bidang otomatis.	MongoDB DBA

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan replika target Anda yang ditetapkan di MongoDB Atlas.	Luncurkan replika target Anda yang ditetapkan di MongoDB Atlas. Di Atlas Live Migration Service, pilih Saya siap untuk bermigrasi.	MongoDB DBA
Membangun koneksi AWS Glue dengan MongoDB Atlas.	Gunakan Perayap AWS Glue untuk terhubung AWS Glue dengan MongoDB Atlas (database target). Langkah ini membantu mempersia pkan lingkungan target untuk migrasi. Lihat informasi yang lebih lengkap dalam dokumentasi AWS Glue.	MongoDB DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membangun koneksi AWS Glue dengan database sumber atau aliran sumber.	Ini membantu mempersia pkan lingkungan target untuk migrasi.	MongoDB DBA
Mengatur transformasi data.	Konfigurasikan logika transformasi untuk memigrasi kan data dari skema terstrukt ur lama ke skema fleksibel MongoDB.	MongoDB DBA
Migrasikan data.	Jadwalkan migrasi di AWS Glue Studio.	MongoDB DBA

Konfigurasikan integrasi operasional

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke cluster.	Connect ke cluster MongoDB Atlas.	Pengembang aplikasi
Berinteraksi dengan data.	Berinteraksi dengan data cluster.	Pengembang aplikasi
Pantau cluster.	Pantau cluster MongoDB Atlas Anda.	MongoDB DBA
Cadangkan dan pulihkan data.	Cadangkan dan pulihkan data cluster.	MongoDB DBA

Pemecahan masalah

Isu	Solusi
Jika Anda mengalami masalah	Lihat Pemecahan Masalah di repositori MongoDB Atlas Resources. CloudFormation

Sumber daya terkait

Semua link berikut, kecuali dinyatakan lain, pergi ke halaman web dalam dokumentasi MongoDB.

Panduan migrasi

- Bermigrasi ke MongoDB AWS Atlas on (Panduan Preskriptif)AWS

Penemuan dan penilaian

- Memori
- Contoh ukuran dengan kumpulan data sampel Atlas
- Contoh ukuran untuk aplikasi seluler
- Lalu Lintas Jaringan
- Penskalaan Otomatis Cluster
- Template ukuran Atlas

Mengkonfigurasi keamanan dan kepatuhan

- Konfigurasikan Entri Daftar Akses IP
- Konfigurasikan Pengguna Database
- Konfigurasikan Akses ke UI Atlas
- Konfigurasikan Peran Database Kustom
- Konfigurasikan Pengguna Database
- Siapkan Koneksi Peering Jaringan
- Pelajari Tentang Titik Akhir Pribadi di Atlas
- Kelola Opsi Otentikasi Multi-Faktor Anda

- Mengatur Otentikasi dan Otorisasi Pengguna dengan LDAP
- Danau Atlas Data
- Enkripsi saat Istirahat menggunakan Manajemen Kunci Pelanggan
- Metode untuk mengambil peran (dokumentasi IAM)
- Enkripsi Tingkat Bidang Sisi Klien
- Enkripsi Otomatis
- Kontrol Keamanan MongoDB Atlas
- Pusat Kepercayaan MongoDB
- Konfigurasikan Fitur Keamanan untuk Cluster

Menyiapkan lingkungan MongoDB Atlas baru di AWS

- Penyedia dan Wilayah Cloud
- Kelola Global Cluster
- Pilih Tingkat Cluster
- Konfigurasikan Pengaturan Tambahan
- Memulai dengan Atlas
- Konfigurasikan Akses ke UI Atlas

Migrasi data

- Migrasi atau Impor Data

Cluster pemantauan

- Pantau Cluster Anda

Mengintegrasikan operasi

- Connect ke Cluster
- Berinteraksi dengan Data Anda
- Pantau Cluster Anda
- Backu Up, Restore, dan Arsipkan Data

GitHub repositori

- Streaming data ke MongoDB Atlas menggunakan AWS Glue

Peran transisi untuk PeopleSoft aplikasi Oracle di Amazon RDS Custom for Oracle

sampath kathirvel, Amazon Web Services

Ringkasan

Untuk menjalankan solusi Oracle PeopleSoft Enterprise Resource Planning (ERP) di Amazon Web Services (AWS), Anda dapat menggunakan Amazon Relational Database Service (Amazon RDS) atau Amazon RDS Custom for Oracle, yang mendukung aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi (OS) dan lingkungan database yang mendasarinya. Untuk faktor kunci yang perlu dipertimbangkan saat merencanakan migrasi, lihat Strategi migrasi database Oracle di AWS Prescriptive Guidance.

Pola ini berfokus pada langkah-langkah untuk melakukan peralihan Oracle Data Guard, atau transisi peran, untuk database PeopleSoft aplikasi yang berjalan di Amazon RDS Custom sebagai database utama dengan database replika baca. Pola ini mencakup langkah-langkah untuk mengonfigurasi fast-start failover (FSFO). Selama proses ini, database dalam konfigurasi Oracle Data Guard terus berfungsi dalam peran baru mereka. Kasus penggunaan umum untuk peralihan Oracle Data Guard adalah latihan pemulihan bencana (DR), aktivitas pemeliharaan terjadwal pada database, dan patch bergulir Standby-First Patch Apply. Untuk informasi selengkapnya, lihat posting blog Mengurangi waktu henti patching database di Amazon RDS Custom.

Prasyarat dan batasan

Prasyarat

- Penyelesaian Add HA ke Oracle PeopleSoft di Amazon RDS Custom dengan menggunakan pola replika baca.

Keterbatasan

- Batasan dan konfigurasi yang tidak didukung untuk RDS Custom for Oracle
- Keterbatasan yang terkait dengan Amazon RDS Custom untuk replika baca Oracle

Versi produk

- Untuk versi Oracle Database yang didukung oleh Amazon RDS Custom, lihat RDS Custom for Oracle.
- Untuk kelas instans Oracle Database yang didukung oleh Amazon RDS Custom, lihat dukungan kelas instans DB untuk RDS Custom for Oracle.

Arsitektur

Tumpukan teknologi

- Amazon RDS Custom for Oracle

Arsitektur target

Diagram berikut menunjukkan instans Amazon RDS Custom DB dan replika baca Amazon RDS Custom. Oracle Data Guard menyediakan transisi peran selama failover untuk DR.

Untuk arsitektur representatif yang menggunakan Oracle PeopleSoft di AWS, lihat Menyiapkan PeopleSoft arsitektur yang sangat tersedia di AWS.

Alat

Layanan AWS

- Amazon RDS Custom for Oracle adalah layanan database terkelola untuk aplikasi lama, kustom, dan paket yang memerlukan akses ke OS dan lingkungan database yang mendasarinya.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Dalam pola ini, Anda mengambil kata sandi pengguna database dari Secrets Manager untuk `RDS_DATAGUARD` dengan nama `do-not-delete-rds-custom-+<<RDS Resource ID>>+-dg` rahasia.

Layanan lainnya

- Oracle Data Guard membantu Anda membuat, memelihara, mengelola, dan memantau database siaga. Pola ini menggunakan Oracle Data Guard Maximum Performance untuk peran transisi (Oracle Data Guard switchover).

Praktik terbaik

Untuk penerapan produksi Anda, sebaiknya luncurkan instance pengamat di Availability Zone ketiga, terpisah dari node replika primer dan baca.

Epik

Memulai transisi peran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jeda otomatisasi database untuk primer dan replika.	Meskipun kerangka kerja otomatisasi Kustom RDS tidak mengganggu proses transisi peran, itu adalah praktik yang baik untuk menjeda otomatisasi selama peralihan Oracle Data Guard. Untuk menjeda dan melanjutkan otomatisasi basis data Kustom RDS, ikuti petunjuk di Menjeda dan melanjutkan otomatisasi Kustom RDS.	Administrator awan, DBA
Periksa status Oracle Data Guard.	Untuk memeriksa status Oracle Data Guard, masuk ke database utama. Pola ini mencakup kode untuk menggunakan database kontainer multitenant (CDB) atau instance non-CDB. Non-CDB `-bash-4.2$ dgmgrl RDS_DATAGUARD@RDS_CUSTOM_ORCL_A`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	DGMGRL for Linux: Release 19.0.0.0.0 - Production on Mon Nov 28 20:55:50 2022 Version 19.10.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "ORCL_A" Connected as SYSDG. DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 59 seconds ago) DGMGRL> CDB CDB-bash-4.2$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_A DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 18 06:13:07 2023	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_A " Connected as SYSDG. DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 52 seconds ago) DGMGRL>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi peran instance.	Buka AWS Management Console, dan navigasikan ke konsol Amazon RDS. Di bagian Replikasi database, pada tab Konektivitas & keamanan, verifikasi peran instance untuk primer dan replika. Peran utama harus sesuai dengan database utama Oracle Data Guard, dan peran replika harus sesuai dengan database siaga fisik Oracle Data Guard.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lakukan peralihan.	Untuk melakukan switchover, sambungkan ke DGMGRL dari simpul utama.
	
	Non-CDB
	
	`DGMGRL> switchover to orcl_d;`
	`Performing switchover NOW, please wait...`
	`Operation requires a connection to database "orcl_d"`
	`Connecting ...`
	`Connected to "ORCL_D"`
	`Connected as SYSDG.`
	`New primary database "orcl_d" is opening...`
	`Operation requires start up of instance "ORCL" on database "orcl_a"`
	`Starting instance "ORCL"...`
	`Connected to an idle instance.`
	`ORACLE instance started.`
	`Connected to "ORCL_A"`
	`Database mounted.`
	`Database opened.`
	`Connected to "ORCL_A"`
	`Switchover succeeded, new primary is "orcl_d"`
	`DGMGRL>`
	
	CDB
	
	`DGMGRL> switchover to rdscdb_b`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`Performing switchover NOW, please wait...`
	`New primary database "rdscdb_b" is opening...`
	`Operation requires start up of instance "RDSCDB" on database "rdscdb_a"`
	`Starting instance "RDSCDB"...`
	`Connected to an idle instance.`
	`ORACLE instance started.`
	`Connected to "RDSCDB_A"`
	`Database mounted.`
	`Database opened.`
	`Connected to "RDSCDB_A"`
	`Switchover succeeded, new primary is "rdscdb_b"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi koneksi Oracle Data Guard.	Setelah switchover, verifikasi koneksi Oracle Data Guard dari node utama ke. `DGMGRL`
	
	Non-CDB
	
	`DGMGRL> show configuration;`
	`Configuration - rds_dg`
	`Protection Mode: MaxAvailability`
	`Members:`
	`orcl_d - Primary database`
	`orcl_a - Physical standby database`
	`Fast-Start Failover: Disabled`
	`Configuration Status:`
	`SUCCESS (status updated 60 seconds ago)`
	`DGMGRL>`
	
	`DGMGRL> show configuration lag;`
	`Configuration - rds_dg`
	`Protection Mode: MaxAvailability`
	`Members:`
	`orcl_d - Primary database`
	`orcl_a - Physical standby database`
	`Transport Lag: 0 seconds (computed 0 seconds ago)`
	`Apply Lag: 0 seconds (computed 0 seconds ago)`
	`Fast-Start Failover: Disabled`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Configuration Status: SUCCESS (status updated 44 seconds ago) DGMGRL> CDB DGMGRL> show configura tion DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_b - Primary database rdscdb_a - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 52 seconds ago) DGMGRL> DGMGRL> show configura tion lag Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_b - Primary database rdscdb_a - Physical standby database Transport Lag: 0 seconds (computed 0 seconds ago)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`Apply Lag: 0 seconds (computed 0 seconds ago)`
	`Fast-Start Failover: Disabled`
	`Configuration Status: SUCCESS (status updated 53 seconds ago)`
	`DGMGRL>`	

Verifikasi peran instans di konsol Amazon RDS.	Setelah Anda menjalankan sakelar peran, konsol Amazon RDS menampilkan peran baru di bawah bagian Replikasi pada tab Konektivitas & Keamanan di bawah Database. Mungkin perlu beberapa menit agar status Replikasi diperbarui dari kosong ke Replikasi.	DBA

Konfigurasikan FSFO

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Setel ulang peralihan.	Atur switchover kembali ke simpul utama.	DBA
Instal dan mulai pengamat.	Proses pengamat adalah komponen `DGMGRL` klien, biasanya berjalan di mesin yang berbeda dari database primer dan siaga. Instalasi ORACLE HOME untuk pengamat dapat berupa	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	instalasi Oracle Client Administrator, atau Anda dapat menginstal Oracle Database Enterprise Edition atau Personal Edition. Untuk informasi selengkapnya tentang penginstalan observer untuk rilis database Anda, lihat Menginstal dan Memulai Observer. Untuk mengonfig urasi ketersediaan tinggi untuk proses pengamat, Anda mungkin ingin melakukan hal berikut: • Aktifkan pemulihan otomatis EC2 instans untuk EC2 instance yang menjalankan pengamat Anda. Anda perlu mengotomatiskan proses startup pengamat sebagai bagian dari startup OS. • Menerapkan pengamat dalam EC2 instance dan konfigurasikan grup Auto EC2 Scaling Amazon dengan ukuran satu (1). Jika terjadi kegagalan EC2 instance, grup penskalaa n otomatis secara otomatis memutar EC2 instance lain. Untuk Oracle 12c Release 2 dan yang lebih baru, Anda dapat menyebarkan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	hingga tiga pengamat. Satu pengamat adalah pengamat utama, dan sisanya adalah pengamat cadangan. Ketika pengamat utama gagal, salah satu pengamat cadangan mengambil peran utama.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Connect ke DGMGRL dari host pengamat.	Host pengamat dikonfigurasi dengan `tnsnames.ora` entri untuk konektivitas basis data primer dan siaga. Anda dapat mengaktifkan FSFO dengan mode perlindungan kinerja maksimum selama kehilanga n data berada dalam FastStart FailoverLagLimitkonfigurasi (nilai dalam detik), Namun, Anda harus menggunakan mode perlindungan ketersedi aan maksimum untuk bekerja untuk mencapai nol kehilanga n data (RPO = 0).
	
	Non-CDB
	
	`DGMGRL> show configura tion;`
	`Configuration - rds_dg`
	`Protection Mode:`
	`MaxAvailability`
	`Members:`
	`orcl_a - Primary`
	`database`
	`orcl_d - Physical`
	`standby database`
	`Fast-Start Failover:`
	`Disabled`
	`Configuration Status:`
	`SUCCESS (status updated`
	`58 seconds ago)`
	`DGMGRL> show configura tion lag`
	`Configuration - rds_dg`
	`Protection Mode:`
	`MaxAvailability`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

```
Members:
orcl_a - Primary
 database
orcl_d - Physical
 standby database
Transport Lag: 0
 seconds (computed 1
 second ago)
Apply Lag: 0 seconds
 (computed 1 second ago)
Fast-Start Failover:
 Disabled
Configuration Status:
SUCCESS (status updated
 5 seconds ago)
DGMGRL>
```

CDB

```
-bash-4.2$ dgmgrl
 C##RDS_DATAGUARD@R
DS_CUSTOM_RDSCDB_A
DGMGRL for Linux:
 Release 19.0.0.0.0 -
 Production on Wed Jan
 18 06:55:09 2023
Version 19.16.0.0.0
Copyright (c) 1982,
 2019, Oracle and/or
 its affiliates.  All
 rights reserved.
Welcome to DGMGRL, type
 "help" for informati
on.
Password:
Connected to "RDSCDB_A
"
Connected as SYSDG.
DGMGRL> show configura
tion
Configuration - rds_dg
```

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> Protection Mode:
	 MaxAvailability
	  Members:
	  rdscdb_a - Primary
	 database
	    rdscdb_b - Physical
	 standby database
	Fast-Start Failover:
	 Disabled
	Configuration Status:
	SUCCESS (status
	 updated 18 seconds ago)
	DGMGRL></pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah database siaga menjadi target failover.	Connect dari node primer atau node observer ke satu database siaga. (Meskipun onfiguration Anda dapat memiliki database siaga mulitiple, Anda hanya perlu terhubung ke satu saat ini.) Non-CDB `DGMGRL> edit database orcl_a set property FastStartFailoverTarget='orcl_d'; Property "faststartfailovertarget" updated DGMGRL> edit database orcl_d set property FastStartFailoverTarget='orcl_a'; Property "faststartfailovertarget" updated DGMGRL> show database orcl_a FastStartFailoverTarget; FastStartFailoverTarget = 'orcl_d' DGMGRL> show database orcl_d FastStartFailoverTarget; FastStartFailoverTarget = 'orcl_a' DGMGRL>` CDB	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`DGMGRL> edit database orcl_a set property FastStartFailoverT arget='rdscdb_b'; Object "orcl_a" was not found DGMGRL> edit database rdscdb_a set property FastStartFailoverT arget='rdscdb_b'; Property "faststar tfailovertarget" updated DGMGRL> edit database rdscdb_b set property FastStartFailoverT arget='rdscdb_a'; Property "faststar tfailovertarget" updated DGMGRL> show database rdscdb_a FastStart FailoverTarget; FastStartFailoverT arget = 'rdscdb_b' DGMGRL> show database rdscdb_b FastStart FailoverTarget; FastStartFailoverT arget = 'rdscdb_a' DGMGRL>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan FastStart FailoverThreshold untuk koneksi ke DGMGRL.	Nilai default adalah 30 detik di Oracle 19c, dan nilai minimum adalah 6 detik. Nilai yang lebih rendah berpotensi mempersingkat tujuan waktu pemulihan (RTO) selama failover. Nilai yang lebih tinggi membantu mengurangi kemungkinan kesalahan sementara failover yang tidak perlu pada database utama.
	
	Kerangka kerja otomatisasi RDS Custom for Oracle memantau kesehatan database dan melakukan tindakan korektif setiap beberapa detik. Oleh karena itu, kami merekomendasikan pengaturan FastStartFailoverThreshold ke nilai yang lebih tinggi dari 10 detik. Contoh berikut mengkonfigurasi nilai ambang pada 35 detik.
	
	Non-CBD atau CDB
	
	`DGMGRL> edit configuration set property FastStartFailoverThreshold=35;`
	`Property "faststartfailoverthreshold" updated`
	`DGMGRL> show configuration FastStartFailoverThreshold;`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`FastStartFailover Threshold = '35' DGMGRL>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Aktifkan FSFO dengan menghubungkan ke DGMGRL dari simpul primer atau pengamat.	Jika database tidak mengaktifkan Flashback Database, pesan peringatan akan `ORA-16827` muncul. Database flashback opsional membantu secara otomatis mengembalikan database primer yang gagal ke titik waktu sebelum failover jika properti FastStartFailoverAutoReinstatekonfigurasi disetel ke TRUE (yang merupakan default).
	
	Non-CDB
	
	`DGMGRL> enable fast_start failover;`
	`Warning: ORA-16827: Flashback Database is disabled`
	`Enabled in Zero Data Loss Mode.`
	`DGMGRL>`
	`DGMGRL> show configuration`
	`Configuration - rds_dg`
	`Protection Mode: MaxAvailability`
	`Members:`
	`orcl_a - Primary database`
	`Warning: ORA-16819: fast-start failover observer not started`
	`orcl_d - (*) Physical standby database`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Warning: ORA-16819: fast-start failover observer not started Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 29 seconds ago) DGMGRL> CDB DGMGRL> enable fast_star t failover; Warning: ORA-16827: Flashback Database is disabled Enabled in Zero Data Loss Mode. DGMGRL> show configura tion; Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database Warning: ORA-16819 : fast-start failover observer not started rdscdb_b - (*) Physical standby database Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 11 seconds ago)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`DGMGRL>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulai pengamat untuk pemantauan FSFO, dan verifikasi statusnya.	Anda dapat memulai pengamat sebelum atau setelah Anda mengaktifkan FSFO. Jika FSFO sudah diaktifkan, pengamat segera mulai memantau status dan koneksi ke basis data siaga utama dan target. Jika FSFO tidak diaktifkan, pengamat tidak memulai pemantauan sampai setelah FSFO diaktifka n.
	
	Ketika Anda memulai pengamat, konfigurasi DB utama akan ditampilkan tanpa pesan kesalahan, sebagaimana dibuktikan oleh perintah sebelumnya. `show configuration`
	
	Non-CDB
	
	`DGMGRL> start observer;`
	`[W000 2022-12-0`
	`1T06:16:51.271+00:00]`
	`FSFO target standby is`
	`orcl_d`
	`Observer 'ip-10-0-`
	`1-89' started`
	`[W000 2022-12-0`
	`1T06:16:51.352+00:00]`
	`Observer trace level is`
	`set to USER`
	
	`DGMGRL> show configura`
	`tion`
	`Configuration - rds_dg`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`Protection Mode: MaxAvailability`
	`Members:`
	`orcl_a - Primary database`
	`orcl_d - (*) Physical standby database`
	`Fast-Start Failover: Enabled in Zero Data Loss Mode`
	`Configuration Status:`
	`SUCCESS (status updated 56 seconds ago)`
	`DGMGRL>`
	
	`DGMGRL> show observer`
	`Configuration - rds_dg`
	`Primary: orcl_a`
	`Active Target: orcl_d`
	`Observer "ip-10-0-1-89" - Master`
	`Host Name: ip-10-0-1-89`
	`Last Ping to Primary: 1 second ago`
	`Last Ping to Target: 1 second ago`
	`DGMGRL>`
	
	CDB
	
	`DGMGRL> start observer;`
	`Succeeded in opening the observer file "/home/oracle/fsfo_ip-10-0-1-56.dat".`
	`[W000 2023-01-18T07:31:32.589+00:00] FSFO target standby is rdscdb_b`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`Observer 'ip-10-0-1-56' started`
	`The observer log file is '/home/oracle/observer_ip-10-0-1-56.log'.`
	
	`DGMGRL> show configuration`
	`Configuration - rds_dg`
	`  Protection Mode: MaxAvailability`
	`  Members:`
	`  rdscdb_a - Primary database`
	`    rdscdb_b - (*) Physical standby database`
	`Fast-Start Failover: Enabled in Zero Data Loss Mode`
	`Configuration Status:`
	`SUCCESS   (status updated 12 seconds ago)`
	`DGMGRL>`
	
	`DGMGRL> show observer;`
	`Configuration - rds_dg`
	`  Primary: rdscdb_a`
	`  Active Target: rdscdb_b`
	`Observer "ip-10-0-1-56" - Master`
	`  Host Name: ip-10-0-1-56`
	`  Last Ping to Primary: 1 second ago`
	`  Last Ping to Target: 2 seconds ago`
	`DGMGRL>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi failover.	Dalam skenario ini, tes failover dapat dilakukan dengan menghentikan EC2 instance utama secara manual. Sebelum menghentikan EC2 instance, gunakan `tail` perintah untuk memantau file log pengamat berdasarkan konfigurasi Anda. Gunakan `DGMGRL` untuk masuk ke database siaga `orcl_d` dengan penggunaRDS_DATAGUARD , dan periksa status Oracle Data Guard. Ini harus menunjukkan bahwa `orcl_d` adalah basis data utama yang baru. **Note** Dalam skenario pengujian failover ini, `orcl_d` adalah database non-CDB. Sebelum failover, database flashback telah diaktifkan. `orcl_a` Setelah database primer sebelumnya kembali online dan dimulai dalam `MOUNT` keadaan, pengamat mengembalikannya ke database siaga baru. Database yang dipulihkan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bertindak sebagai target FSFO untuk database utama baru. Anda dapat memverifikasi detailnya di log pengamat. `DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_d - Primary database Warning: ORA-16824 : multiple warnings, including fast-star t failover-related warnings, detected for the database orcl_a - (*) Physical standby database (disabled) ORA-16661: the standby database needs to be reinstated Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 25 seconds ago) DGMGRL>`  Berikut ini menunjukkan contoh output dio`observer. log` .  `$ tail -f /tmp/obse rver.log`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Unable to connect to database using rds_custom_orcl_a [W000 2023-01-18T07:50:32.589+00:00] Primary database cannot be reached. [W000 2023-01-18T07:50:32.589+00:00] Fast-Start Failover threshold has expired. [W000 2023-01-18T07:50:32.590+00:00] Try to connect to the standby. [W000 2023-01-18T07:50:32.590+00:00] Making a last connection attempt to primary database before proceeding with Fast-Start Failover. [W000 2023-01-18T07:50:32.591+00:00] Check if the standby is ready for failover. [S002 2023-01-18T07:50:32.591+00:00] Fast-Start Failover started... 2023-01-18T07:50:32.591+00:00 Initiating Fast-Start Failover to database "orcl_d"... [S002 2023-01-18T07:50:32.592+00:00] Initiating Fast-start Failover. Performing failover NOW, please wait...	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Failover succeeded, new primary is "orcl_d" 2023-01-18T07:55:32.101+00:00 [S002 2023-01-18T07:55:32.591+00:00] Fast-Start Failover finished... [W000 2023-01-18T07:55:32.591+00:00] Failover succeeded. Restart pinging. [W000 2023-01-18T07:55:32.603+00:00] Primary database has changed to orcl_d. [W000 2023-01-18T07:55:33.618+00:00] Try to connect to the primary. [W000 2023-01-18T07:55:33.622+00:00] Try to connect to the primary rds_custom_orcl_d. [W000 2023-01-18T07:55:33.634+00:00] The standby orcl_a needs to be reinstated [W000 2023-01-18T07:55:33.654+00:00] Try to connect to the new standby orcl_a. [W000 2023-01-18T07:55:33.654+00:00] Connection to the primary restored! [W000 2023-01-18T07:55:35.654+00:00] Disconnecting from database rds_custom_orcl_d.`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`[W000 2023-01-18T07:55:57.701+00:00] Try to connect to the new standby orcl_a. ORA-12170: TNS:Connect timeout occurred`	

Konfigurasikan konektivitas antara aplikasi Oracle Peoplesoft dan database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan mulai layanan di database utama.	Anda dapat menghindari perubahan konfigurasi aplikasi selama transisi peran dengan menggunakan entri TNS yang berisi titik akhir basis data primer dan siaga dalam konfigurasi. Anda dapat menentukan dua layanan basis data berbasis peran untuk mendukung keduanya read/write dan beban kerja hanya-baca. Dalam contoh berikut, `orcl_rw` adalah read/write layanan yang aktif pada database utama. `orcl_ro`adalah layanan hanya-baca dan aktif pada database siaga yang telah dibuka dalam mode hanya-baca. `SQL> select name,open_mode from v$database; NAME OPEN_MODE`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--------- --------- ----------- ORCL READ WRITE SQL> exec dbms_serv ice.create_service ('orcl_rw','orcl_r w'); PL/SQL procedure successfully completed . SQL> exec dbms_serv ice.create_service ('orcl_ro','orcl_r o'); PL/SQL procedure successfully completed . SQL> exec dbms_serv ice.start_service('orcl_rw'); PL/SQL procedure successfully completed . SQL>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai layanan di database siaga.	Untuk memulai layanan dalam database siaga read-only, gunakan kode berikut. `SQL> select name,open _mode from v$database; NAME OPEN_MODE --------- --------- ----------- ORCL READ ONLY WITH APPLY SQL> exec dbms_serv ice.start_service( 'orcl_ro'); PL/SQL procedure successfully completed . SQL>`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Otomatiskan memulai layanan saat DB utama dimulai ulang.	Untuk memulai layanan secara otomatis di database utama saat dimulai ulang, gunakan kode berikut. `SQL> CREATE OR REPLACE TRIGGER TrgDgServices after startup on database DECLARE db_role VARCHAR(30); db_open_mode VARCHAR(30); BEGIN SELECT DATABASE_ROLE, OPEN_MODE INTO db_role, db_open_mode FROM V$DATABASE; IF db_role = 'PRIMARY' THEN DBMS_SERV 2 ICE.START_SERVICE('orcl_rw'); END IF; IF db_role = 'PHYSICAL STANDBY' AND db_open_mode LIKE 'READ ONLY%' THEN DBMS_SERVICE.START_SERVICE('orcl_ro'); END IF; END; / Trigger created. SQL>`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan koneksi antara database read-only read/write dan read-only.	Anda dapat menggunakan contoh konfigurasi aplikasi berikut untuk koneksi read/write dan hanya-baca. `ORCL_RW = (DESCRIPTION = (CONNECT_TIMEOUT= 120)(RETRY_COUNT=20)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3) (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST=devpsftdb.******.us-west-2.rds.amazonaws.com) (PORT=1521)) (ADDRESS = (PROTOCOL = TCP)(HOST=psftread.******.us-west-2.rds.amazonaws.com) (PORT=1521))) (CONNECT_DATA=(SERVICE_NAME = orcl_rw))) ORCL_RO = (DESCRIPTION = (CONNECT_TIMEOUT= 120)(RETRY_COUNT=20)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3) (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST=devpsftdb.******.us-west-2.rds.amazonaws.com) (PORT=1521))`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`(ADDRESS = (PROTOCOL = TCP)(HOST=psftread.******.us-west-2.rds.amazonaws.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME = orcl_ro))`	

Sumber daya terkait

- Mengaktifkan Ketersediaan Tinggi dengan Data Guard di Amazon RDS Custom for Oracle (Panduan Teknis AWS)
- Mengonfigurasi Amazon RDS sebagai Oracle Database PeopleSoft (whitepaper AWS)
- Panduan Oracle Data Guard Broker (dokumentasi referensi Oracle)
- Konsep dan Administrasi Data Guard (dokumentasi referensi Oracle)
- Persyaratan Konfigurasi FAN dan FCF Khusus Oracle Data Guard (dokumentasi referensi Oracle)

Membongkar data dari klaster Amazon Redshift di seluruh akun ke Amazon S3

Andrew Kamel, Amazon Web Services

Ringkasan

Saat Anda menguji aplikasi, akan sangat membantu jika memiliki data produksi di lingkungan pengujian Anda. Menggunakan data produksi dapat memberi Anda penilaian yang lebih akurat tentang aplikasi yang sedang Anda kembangkan.

Pola ini mengekstrak data dari klaster Amazon Redshift di lingkungan produksi ke bucket Amazon Simple Storage Service (Amazon S3) di lingkungan pengembangan di Amazon Web Services ().AWS

Pola langkah-langkah melalui pengaturan akun DEV dan PROD, termasuk yang berikut:

- Sumber daya yang dibutuhkan
- AWS Identity and Access Management Peran (IAM)
- Penyesuaian jaringan untuk subnet, grup keamanan, dan virtual private cloud (VPC) untuk mendukung koneksi Amazon Redshift
- Contoh AWS Lambda fungsi dengan runtime Python untuk menguji arsitektur

Untuk memberikan akses ke klaster Amazon Redshift, pola tersebut digunakan AWS Secrets Manager untuk menyimpan kredensil yang relevan. Manfaatnya adalah memiliki semua informasi yang diperlukan untuk terhubung langsung ke cluster Amazon Redshift tanpa perlu mengetahui di mana cluster Amazon Redshift berada. Selain itu, Anda dapat memantau penggunaan rahasia.

Rahasia yang disimpan di Secrets Manager mencakup host cluster Amazon Redshift, nama database, port, dan kredenal yang relevan.

Untuk informasi tentang pertimbangan keamanan saat menggunakan pola ini, lihat bagian Praktik terbaik.

Prasyarat dan batasan

Prasyarat

- Cluster Amazon Redshift yang berjalan di akun PROD
- Bucket S3 yang dibuat di akun DEV

- VPC mengintip antara akun DEV dan PROD, dengan tabel rute yang disesuaikan
- Nama host DNS dan resolusi DNS diaktifkan untuk kedua peered VPCs

Batasan

- Bergantung pada jumlah data yang ingin Anda kueri, fungsi Lambda mungkin habis waktu.

  Jika proses Anda membutuhkan waktu lebih lama daripada batas waktu maksimum Lambda (15 menit), gunakan pendekatan asinkron untuk kode Lambda Anda. Contoh kode untuk pola ini menggunakan pustaka psycopg2 untuk Python, yang saat ini tidak mendukung pemrosesan asinkron.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan arsitektur target, dengan akun DEV dan PROD.

Diagram menunjukkan alur kerja berikut:

1. Fungsi Lambda di akun DEV mengasumsikan peran IAM yang diperlukan untuk mengakses kredensil Amazon Redshift di Secrets Manager di akun PROD.

   Fungsi Lambda kemudian mengambil rahasia cluster Amazon Redshift.
2. Fungsi Lambda di akun DEV menggunakan informasi untuk terhubung ke cluster Amazon Redshift di akun PROD melalui peered. VPCs

   Fungsi Lambda kemudian mengirimkan perintah unload untuk menanyakan cluster Amazon Redshift di akun PROD.
3. Cluster Amazon Redshift di akun PROD mengasumsikan peran IAM yang relevan untuk mengakses bucket S3 di akun DEV.

   Cluster Amazon Redshift membongkar data kueri ke bucket S3 di akun DEV.

Meminta data dari Amazon Redshift

Diagram berikut menunjukkan peran yang digunakan untuk mengambil kredenal Amazon Redshift dan terhubung ke cluster Amazon Redshift. Alur kerja diprakarsai oleh fungsi Lambda.

Diagram menunjukkan alur kerja berikut:

1. `CrossAccount-SM-Read-Role`Dalam akun DEV mengasumsikan `SM-Read-Role` di akun PROD.
2. `SM-Read-Role`Peran menggunakan kebijakan terlampir untuk mengambil rahasia dari Secrets Manager.
3. Kredensialnya digunakan untuk mengakses kluster Amazon Redshift.

Mengunggah data ke Amazon S3

Diagram berikut menunjukkan proses baca-tulis lintas akun untuk mengekstrak data dan mengunggahnya ke Amazon S3. Alur kerja diprakarsai oleh fungsi Lambda. Pola ini merantai peran IAM di Amazon Redshift. Perintah unload yang berasal dari cluster Amazon Redshift mengasumsikan, dan kemudian `CrossAccount-S3-Write-Role` mengasumsikan. `S3-Write-Role` Rantai peran ini memberi Amazon Redshift akses ke Amazon S3.

Alur kerja mencakup langkah-langkah berikut:

1. `CrossAccount-SM-Read-Role`Dalam akun DEV mengasumsikan `SM-Read-Role` di akun PROD.
2. Ini `SM-Read-Role` mengambil kredensi Amazon Redshift dari Secrets Manager.
3. Fungsi Lambda terhubung ke cluster Amazon Redshift dan mengirimkan kueri.
4. Cluster Amazon Redshift mengasumsikan. `CrossAccount-S3-Write-Role`
5. `CrossAccount-S3-Write-Role`Mengasumsikan `S3-Write-Role` di akun DEV.
6. Hasil kueri diturunkan ke bucket S3 di akun DEV.

Alat

Layanan AWS

- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Redshift adalah layanan gudang data skala petabyte terkelola di AWS Cloud.
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub unload-redshift-to-s3-python.

Praktik terbaik

Penafian keamanan

Sebelum Anda menerapkan solusi ini, pertimbangkan rekomendasi keamanan penting berikut:

- Ingatlah bahwa menghubungkan akun pengembangan dan produksi dapat meningkatkan cakupan dan menurunkan postur keamanan secara keseluruhan. Kami merekomendasikan untuk menerapkan solusi ini hanya sementara, mengekstrak bagian data yang diperlukan dan kemudian segera menghancurkan sumber daya yang digunakan. Untuk menghancurkan sumber daya, Anda harus menghapus fungsi Lambda, menghapus peran dan kebijakan IAM yang dibuat untuk solusi ini, dan mencabut akses jaringan apa pun yang diberikan di antara akun.
- Konsultasikan dengan tim keamanan dan kepatuhan Anda sebelum menyalin data apa pun dari produksi ke lingkungan pengembangan. Informasi identitas pribadi (PII), Informasi kesehatan yang dilindungi (PHI), dan data rahasia atau teregulasi lainnya umumnya tidak boleh disalin dengan cara ini. Salin hanya informasi non-rahasia yang tersedia untuk umum (misalnya, data stok publik dari frontend toko). Pertimbangkan untuk mentokenisasi atau menganonimkan data, atau menghasilkan data uji sintetis, alih-alih menggunakan data produksi bila memungkinkan. Salah satu prinsip AWS keamanan adalah menjauhkan orang dari data. Dengan kata lain, pengembang tidak boleh melakukan operasi di akun produksi.
- Batasi akses ke fungsi Lambda di akun pengembangan karena dapat membaca data dari cluster Amazon Redshift di lingkungan produksi.

- Untuk menghindari gangguan lingkungan produksi, terapkan rekomendasi berikut:
  - Gunakan akun pengembangan khusus yang terpisah untuk kegiatan pengujian dan pengembangan.
  - Menerapkan kontrol akses jaringan yang ketat dan membatasi lalu lintas antar akun hanya untuk apa yang diperlukan.
  - Memantau dan mengaudit akses ke lingkungan produksi dan sumber data.
  - Menerapkan kontrol akses dengan hak istimewa paling rendah untuk semua sumber daya dan layanan yang terlibat.
  - Tinjau dan putar kredenal secara teratur, seperti AWS Secrets Manager rahasia dan kunci akses peran IAM.
- Lihat dokumentasi keamanan berikut untuk layanan yang digunakan dalam artikel ini:
  - AWS Lambda keamanan
  - Keamanan Amazon Redshift
  - Keamanan Amazon S3
  - AWS Secrets Manager keamanan
  - Praktik terbaik keamanan IAM

Keamanan adalah prioritas utama saat mengakses data dan sumber daya produksi. Selalu ikuti praktik terbaik, terapkan kontrol akses dengan hak istimewa, dan tinjau dan perbarui langkah-langkah keamanan Anda secara teratur.

Epik

Data kueri dari Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat rahasia untuk cluster Amazon Redshift.	Untuk membuat rahasia untuk cluster Amazon Redshift, lakukan hal berikut: 1. Di akun PROD, masuk ke Konsol Manajemen AWS, dan buka konsol Secrets Manager di https://	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	console.aws.amazon.com/secretsmanager/. 2. Pilih Simpan Rahasia baru. 3. Pilih Kredensial untuk gudang data Amazon Redshift. 4. Untuk nama Pengguna dan Kata Sandi, masukkan nilai untuk instans Anda, dan konfirmasikan atau pilih nilai untuk kunci Enkripsi. 5. Pilih gudang data Amazon Redshift yang akan diakses rahasia Anda. 6. Masukkan `Redshift-Creds-Secret` nama rahasianya. 7. Selesaikan langkah-langkah pembuatan yang tersisa dengan pilihan default, lalu pilih Store. 8. Lihat rahasia Anda, dan catat nilai Rahasia ARN yang dihasilkan untuk mengidentifikasi rahasia.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran untuk mengakses Secrets Manager.	Untuk membuat peran, lakukan hal berikut: 1. Di akun PROD, buka konsol IAM di. https://console.aws.amazon.com/iam/ 2. Pilih Policies (Kebijakan). 3. Pilih Buat kebijakan. 4. Pilih tab JSON, lalu masukkan kebijakan IAM seperti berikut ini: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "secretsmanager:GetResourcePolicy", "secretsmanager:GetSecretValue", "secretsmanager:DescribeSecret", "secretsmanager:ListSecretVersionIds" ], "Resource": [ "<Redshift-Creds-Secret-ARN>"`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>]
	        },
	        {
	            "Effect":
	 "Allow",
	            "Action":
	 "secretsmanager:Li
	stSecrets",
	            "Resource
	": "*"
	        }
	    ]
	}</pre>
	Ganti `Redshift-Creds-Secret-ARN` dengan Amazon Resource Name (ARN) dari rahasia Secrets Manager yang berisi informasi dan kredensil untuk cluster Amazon Redshift.	

Mengunggah data ke Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran untuk mengakses bucket S3.	Untuk membuat peran untuk mengakses bucket S3, lakukan hal berikut: 1. Di akun DEV, buka konsol IAM. 2. Pilih Policies (Kebijakan). 3. Pilih Buat kebijakan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pilih tab JSON, lalu masukkan kebijakan IAM seperti berikut ini: `{ "Version": "2012-10-17", "Statement": [{ "Sid": "kmsstmt", "Effect": "Allow", "Action": ["kms:Decrypt", "kms:Encrypt", "kms:GenerateDataKey"], "Resource": ["<kms-key-arn>"] }, { "Sid": "s3stmt", "Effect": "Allow", "Action": ["s3:PutObject", "s3:Get*", "s3:List*"], "Resource": ["arn:aws:s3:::mybucket",`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"arn:aws:
	s3:::mybucket/*"
	]
	}
	]
	}</pre>
	Ganti `mybucket` dengan nama bucket S3 yang ingin Anda akses. Juga, jika bucket S3 dienkripsi, ganti `kms-key-arn` dengan ARN dari kunci AWS Key Management Service (AWS KMS) yang digunakan untuk mengenkripsi bucket S3. Jika tidak, Anda tidak memerlukan AWS KMS bagian dalam kebijakan.
	5. Pilih Kebijakan tinjauan, masukkan `S3-Write-Policy` sebagai nama kebijakan, lalu pilih Buat kebijakan.
	6. Di panel navigasi, pilih Peran.
	7. Pilih Buat peran.
	8. Untuk peran entitas tepercaya, pilih Kebijakan kepercayaan khusus.
	9. Pilih Berikutnya: Izin, lalu pilih kebijakan S3-Write-Policy yang Anda buat.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	10. Masukkan `S3-Write-Role` sebagai nama peran, lalu pilih Buat peran.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran Amazon Redshift.	Untuk membuat peran Amazon Redshift, lakukan hal berikut: 1. Di akun PROD, buka konsol IAM. 2. Pilih Policies (Kebijakan). 3. Pilih Buat kebijakan. 4. Pilih tab JSON, lalu masukkan kebijakan IAM seperti berikut ini: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "CrossAccountPolicy", "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "S3-Write-Role-ARN" } ] }` Ganti `S3-Write-Role-ARN` dengan ARN untuk `S3-Write-Role` di akun DEV. 5. Pilih Kebijakan tinjauan, masukkan `S3-Write-`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Role-Assume-Policy` sebagai nama kebijakan, lalu pilih Buat kebijakan. 6. Di panel navigasi, pilih Peran, lalu pilih Buat peran. 7. Pilih layanan AWS sebagai jenis entitas tepercaya Anda, lalu pilih Redshift, Redshift - Customizable. 8. Pilih Berikutnya: Izin, lalu pilih kebijakan S3-Write-Role-Assume-Policy yang Anda buat. 9. Masukkan `CrossAccount-S3-Write-Role` sebagai nama peran, lalu pilih Buat peran. 10. Kaitkan peran IAM dengan cluster Amazon Redshift Anda.	

Deploy fungsi Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan fungsi Lambda.	Untuk menerapkan fungsi Lambda di VPC peered, lakukan hal berikut: 1. Buka konsol Lambda di. https://console.aws.amazon.com/lambda/ 2. Pilih Fungsi.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pilih Buat fungsi. 4. Di bawah Informasi Dasar, untuk Nama fungsi, masukkan nama untuk fungsi Anda. 5. Untuk Runtime, pilih Python 3.8. 6. Perluas Ubah peran eksekusi default, lalu lakukan hal berikut: a. Pilih Gunakan peran yang ada. b. Untuk peran yang ada, pilih peran Lambda CrossAccount-RM-baca-peran yang Anda buat sebelumnya. 7. Perluas Pengaturan lanjutan, dan lakukan hal berikut: a. Pilih kotak centang Aktifkan VPC. b. Untuk VPC, pilih VPC peered di akun DEV. c. Untuk Subnet, pilih subnet pribadi. d. Untuk grup Keamanan, pilih grup keamanan default. 8. Pilih Buat fungsi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	9. Tambahkan pustaka psycopg2 sebagai lapisan ke fungsi Lambda. **Note** Anda dapat menggunakan layer yang sudah diterapkan dari repositori psycopg2-lambda-layer. Pastikan untuk menggunakan URL berdasark an runtime Anda Wilayah AWS dan Python.	

Uji arsitekturnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Impor sumber daya yang dibutuhkan.	Untuk mengimpor sumber daya yang diperlukan, jalankan perintah berikut:
	
	`import ast`
	`import boto3`
	`import psycopg2`
	`import base64`
	`from botocore.`
	`exceptions import`
	` ClientError`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan fungsi handler Lambda.	Fungsi Lambda menggunakan AWS Security Token Service (AWS STS) untuk akses lintas akun dan manajemen kredensi sementara. Fungsi ini menggunakan operasi AssumeRole API untuk sementara mengasumsikan izin peran `sm_read_role` IAM.
	
	Untuk menjalankan fungsi Lambda, gunakan kode contoh berikut:
	
	`def lambda_handler(eve nt, context):`
	`  sts_client =  boto3.client('sts')`
	
	`  # Secrets Manager  Configurations`
	`  secret_name =  "redshift_creds"`
	`  sm_region = "eu- west-1"`
	`  sm_read_role =  "arn:aws:iam::PROD _ACCOUNT_NUMBER:role/ SM-Read-Role"`
	
	`  # S3 Bucket  Configurations`
	`  s3_bucket_path =  "s3://mybucket/"`
	`  s3_bucket_region =  "eu-west-1"`
	`  s3_write_role =  "arn:aws:iam::DEV_`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`ACCOUNT_NUMBER:role/S3-Write-Role"`
	
	`    # Redshift Configurations`
	`    sql_query = "select * from category"`
	`    redshift_db = "dev"`
	`    redshift_s3_write_role = "arn:aws:iam::PROD_ACCOUNT_NUMBER:role/CrossAccount-S3-Write-Role"`
	
	`    chained_s3_write_role = "%s,%s" % (redshift_s3_write_role, s3_write_role)`
	
	`    assumed_role_object = sts_client.assume_role(`
	`        RoleArn=sm_read_role,`
	`        RoleSessionName="CrossAccountRoleAssumption",`
	`        ExternalId="YOUR_EXTERNAL_ID",`
	`    )`
	`    credentials = assumed_role_object['Credentials']`
	
	`    secret_dict = ast.literal_eval(get_secret(credentials, secret_name, sm_region))`
	`    execute_query(secret_dict, sql_query, s3_bucket_path,`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>chained_s3_write_role, s3_bucket_region, redshift_db)
	
	    return {
	        'statusCode': 200
	    }</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Dapatkan rahasianya.	Untuk mendapatkan rahasia Amazon Redshift, gunakan kode contoh berikut:
	
	`def get_secret(credentials, secret_name, sm_region):`
	`  # Create a Secrets Manager client`
	`  session = boto3.session.Session()`
	`  sm_client = session.client(`
	`  service_name='secretsmanager',`
	`  aws_access_key_id=credentials['AccessKeyId'],`
	`  aws_secret_access_key=credentials['SecretAccessKey'],`
	`  aws_session_token=credentials['SessionToken'],`
	`  region_name=sm_region`
	`  )`
	
	`  try:`
	`  get_secret_value_response = sm_client.get_secret_value(`
	`  SecretId=secret_name`
	`  )`
	`  except ClientError as e:`
	`  print(e)`
	`  raise e`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> else:
	        if 'SecretSt
	ring' in get_secre
	t_value_response:
	            return
	 get_secret_value_r
	esponse['SecretStr
	ing']
	        else:
	            return
	 base64.b64decode(g
	et_secret_value_re
	sponse['SecretBina
	ry'])</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan perintah bongkar muat.	Untuk membongkar data ke bucket S3, gunakan kode contoh berikut.
	
	`def execute_query(secret_dict, sql_query, s3_bucket_path, chained_s3_write_role, s3_bucket_region, redshift_db):`
	`  conn_string = "dbname='%s' port='%s' user='%s' password='%s' host='%s'" \`
	`  % (redshift_db,`
	`  secret_dict["port"],`
	`  secret_dict["username"],`
	`  secret_dict["password"],`
	`  secret_dict["host"])`
	`  con = psycopg2.connect(conn_string)`
	`  unload_command = "UNLOAD ('{}') TO '{}' IAM_ROLE '{}' DELIMITER '|' REGION '{}';" \`
	`  .format(sql_query,`
	`  s3_bucket_path + str(datetime.datetime.now()) + ".csv",`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>chained_s3_write_role,
	
	s3_bucket_region)
	
	# Opening a cursor and run query
	cur = con.cursor()
	cur.execute(unload_command)
	
	print(cur.fetchone())
	cur.close()
	con.close()</pre>	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus fungsi Lambda.	Untuk menghindari biaya yang tidak direncanakan, hapus sumber daya dan koneksi antara akun DEV dan PROD. Untuk menghapus fungsi Lambda, lakukan hal berikut: 1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/. 2. Cari dan pilih fungsi Lambda yang Anda buat. 3. Pilih Tindakan, lalu pilih Hapus.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Konfirmasi penghapusan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus peran dan kebijakan IAM.	Hapus peran dan kebijakan IAM dari akun DEV dan PROD. Di akun DEV, lakukan hal berikut: 1. Buka konsol IAM. 2. Hapus peran berikut: • `S3-Write-Role` • `CrossAccount-RM-Read-Role` (Peran Lambda) 3. Hapus kebijakan terkait: • `S3-Write-Policy` • CrossAccount Kebijakan untuk mengasumsikan peran akun PROD Di akun PROD, lakukan hal berikut: 1. Buka konsol IAM. 2. Hapus peran berikut: • `SM-Read-Role` • `CrossAccount-S3-Write-Role` 3. Hapus kebijakan terkait: • CrossAccount Kebijakan untuk mengakses Secrets Manager	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `S3-Write-Role-Assume-Policy`	
Hapus rahasia di Secrets Manager.	Untuk menghapus rahasia, lakukan hal berikut: 1. Di akun PROD, buka konsol Secrets Manager. 2. Cari dan pilih rahasia bernama`Redshift-Creds-Secret` . 3. Pilih Tindakan, lalu pilih Hapus rahasia. 4. Konfirmasi penghapusan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus peering VPC dan aturan grup keamanan.	Untuk menghapus aturan VPC peering dan grup keamanan, lakukan hal berikut: 1. Di akun PROD, buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/. 2. Arahkan ke Grup Keamanan. 3. Temukan grup keamanan yang digunakan oleh cluster Amazon Redshift. 4. Edit aturan masuk, dan hapus aturan yang mengizinkan koneksi dari VPC Lambda akun DEV. 5. Arahkan ke VPC Peering Connections, dan hapus koneksi peering.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus data dari bucket S3.	Untuk menghapus data dari Amazon S3, lakukan hal berikut: 1. Di akun DEV, buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/ 2. Temukan bucket yang Anda gunakan untuk penyimpanan data. 3. Hapus objek di dalam ember, atau hapus seluruh bucket jika tidak diperlukan lagi.	DevOps insinyur
Bersihkan AWS KMS kunci.	Jika Anda membuat AWS KMS kunci khusus untuk enkripsi, lakukan hal berikut: 1. Buka AWS KMS konsol di https://console.aws.amazon.com/kms/. 2. Temukan kunci apa pun yang dibuat untuk pola ini. 3. Jadwalkan kunci untuk dihapus. (Ada masa tunggu wajib untuk penghapusan kunci).	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau dan hapus CloudWatch log Amazon.	Untuk menghapus CloudWatch log, lakukan hal berikut: 1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/. 2. Periksa grup log apa pun yang dibuat oleh fungsi Lambda Anda atau cluster Amazon Redshift. 3. Hapus grup log ini jika tidak lagi diperlukan.	DevOps insinyur

Sumber daya terkait

- CloudWatch Dokumentasi Amazon
- Dokumentasi IAM
- Dokumentasi Lambda
- Dokumentasi Amazon Redshift
- Dokumentasi Amazon S3
- AWS Secrets Manager dokumentasi
- AWS prinsip keamanan

Informasi tambahan

Setelah Anda membongkar data dari Amazon Redshift ke Amazon S3, Anda dapat menganalisisnya dengan menggunakan Amazon Athena.

Amazon Athena adalah layanan kueri data besar yang bermanfaat ketika Anda perlu mengakses data dalam jumlah besar. Anda dapat menggunakan Athena tanpa harus menyediakan server atau database. Athena mendukung kueri yang kompleks, dan Anda dapat menjalankannya pada objek yang berbeda.

Seperti kebanyakan Layanan AWS, manfaat utama menggunakan Athena adalah memberikan fleksibilitas yang besar dalam cara Anda menjalankan kueri tanpa kerumitan tambahan. Saat Anda menggunakan Athena, Anda dapat menanyakan tipe data yang berbeda, seperti CSV dan JSON, di Amazon S3 tanpa mengubah tipe data. Anda dapat meminta data dari berbagai sumber, termasuk di luar AWS. Athena mengurangi kompleksitas karena Anda tidak perlu mengelola server. Athena membaca data langsung dari Amazon S3 tanpa memuat atau mengubah data sebelum Anda menjalankan kueri.

Pola migrasi database berdasarkan beban kerja

Topik

- IBM
- Microsoft
- N/A
- Sumber terbuka
- Oracle
- SAP

IBM

- Migrasikan database Db2 dari Amazon ke EC2 Aurora MySQL yang kompatibel dengan menggunakan AWS DMS
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan menggunakan pengiriman log untuk mengurangi waktu pemadaman
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan pemulihan bencana ketersediaan tinggi
- Bermigrasi dari IBM Db2 di Amazon EC2 ke Aurora PostgreSQL yang kompatibel dengan menggunakan AWS DMS dan AWS SCT
- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2
- Streaming data dari IBM Db2, SAP, Sybase, dan database lainnya ke MongoDB Atlas AWS

Microsoft

- Mempercepat penemuan dan migrasi beban kerja Microsoft ke AWS
- Mengakses tabel Microsoft SQL Server lokal dari Microsoft SQL Server di Amazon EC2 menggunakan server tertaut
- Menilai kinerja kueri untuk memigrasi database SQL Server ke MongoDB Atlas di AWS
- Mengotomatiskan tugas database dalam edisi SQL Server Express yang berjalan di Amazon EC2 dengan menggunakan AWS Lambda dan Penjadwal Tugas
- Ubah aplikasi Python dan Perl untuk mendukung migrasi database dari Microsoft SQL Server ke Amazon Aurora PostgreSQL Edisi yang kompatibel
- Konfigurasikan perutean hanya-baca dalam grup ketersediaan Selalu Aktif di SQL Server di AWS
- Konfigurasikan otentikasi Windows untuk Amazon RDS untuk Microsoft SQL Server menggunakan AWS Managed Microsoft AD
- Buat CloudFormation templat AWS untuk tugas AWS DMS menggunakan Microsoft Excel dan Python
- Menerapkan instance cluster failover SQL Server di Amazon EC2 dan Amazon FSx dengan menggunakan Terraform
- Ekspor database Microsoft SQL Server ke Amazon S3 dengan menggunakan AWS DMS
- Ekspor Amazon RDS untuk tabel SQL Server ke bucket S3 dengan menggunakan AWS DMS
- Menerapkan SHA1 hashing untuk data PII saat bermigrasi dari SQL Server ke PostgreSQL
- Menyerap dan memigrasikan instans EC2 Windows ke akun AWS Managed Services
- Mengatur cluster failover Microsoft SQL Server di Amazon EC2 menggunakan Windows FSx File Server
- Memigrasikan antrian pesan dari Microsoft Azure Service Bus ke Amazon SQS
- Migrasi database Microsoft SQL Server dari Amazon EC2 ke Amazon DocumentDB dengan menggunakan AWS DMS
- Migrasikan database Microsoft SQL Server ke Aurora MySQL dengan menggunakan AWS DMS dan AWS SCT
- Migrasi aplikasi.NET dari Microsoft Azure App Service ke AWS Elastic Beanstalk
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan server tertaut

- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan metode pencadangan dan pemulihan asli
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan AWS DMS
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT
- Memigrasi database Microsoft SQL Server lokal ke Microsoft SQL Server di Amazon yang menjalankan Linux EC2
- Migrasi data dari Microsoft Azure Blob ke Amazon S3 dengan menggunakan Rclone
- Migrasi aplikasi yang dihosting IIS ke Amazon dengan menggunakan appcmd.exe EC2
- Migrasi grup ketersediaan Microsoft SQL Server Always On menggunakan AWS Application Migration Service
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2 menggunakan Layanan Migrasi Aplikasi
- Migrasi database relasional ke MongoDB Atlas di AWS
- Migrasikan SQL Server ke AWS menggunakan grup ketersediaan terdistribusi
- Migrasikan sertifikat SSL Windows ke Application Load Balancer menggunakan ACM
- Menghosting ulang beban kerja lokal di AWS Cloud: daftar periksa migrasi
- Mengatasi kesalahan koneksi setelah memigrasikan Microsoft SQL Server ke AWS Cloud
- Mengirim pemberitahuan untuk instans database Amazon RDS for SQL Server dengan menggunakan server SMTP lokal dan Database Mail
- Siapkan CI/CD pipeline untuk migrasi database dengan menggunakan Terraform
- Menyiapkan infrastruktur Multi-AZ untuk SQL Server Always On FCI dengan menggunakan Amazon FSx

N/A

- Membuat proses persetujuan untuk permintaan firewall selama migrasi rehost ke AWS
- Enkripsi Amazon RDS yang ada untuk instans DB Amazon RDS for PostgreSQL
- Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB
- Menerapkan pemulihan bencana lintas wilayah dengan AWS DMS dan Amazon Aurora

Sumber terbuka

- Secara otomatis menghasilkan model PynamoDB dan fungsi CRUD untuk Amazon DynamoDB dengan menggunakan aplikasi Python
- Connect dengan menggunakan terowongan SSH di pgAdmin
- Buat pengguna dan peran aplikasi di Aurora PostgreSQL yang kompatibel
- Aktifkan koneksi terenkripsi untuk instans PostgreSQL DB di Amazon RDS
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dengan dan menggunakan dan AWS SCT AWS DMS AWS CLI CloudFormation
- Memigrasi database MariaDB lokal ke Amazon RDS for MariaDB menggunakan alat bawaan
- Memigrasikan database MySQL lokal ke Amazon EC2
- Memigrasikan database MySQL lokal ke Amazon RDS for MySQL
- Memigrasikan database MySQL lokal ke Aurora MySQL
- Memigrasikan database PostgreSQL lokal ke Aurora PostgreSQL
- Memigrasikan database Couchbase Server ke Amazon EC2
- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2 dengan Auto Scaling
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Oracle dan AWS DMS SharePlex
- Bermigrasi dari PostgreSQL di Amazon ke Amazon RDS untuk PostgreSQL menggunakan pglogical EC2
- Migrasikan aplikasi Java lokal ke AWS menggunakan AWS App2Container
- Migrasikan database MySQL lokal ke Aurora MySQL menggunakan Percona, Amazon EFS, dan Amazon S3 XtraBackup
- Migrasikan tabel eksternal Oracle ke Amazon Aurora PostgreSQL yang kompatibel
- Migrasikan fungsi dan prosedur Oracle yang memiliki lebih dari 100 argumen ke PostgreSQL
- Migrasikan beban kerja Redis ke Redis Enterprise Cloud di AWS
- Pantau Amazon Aurora untuk instance tanpa enkripsi
- Mulai ulang AWS Replication Agent secara otomatis tanpa menonaktifkan SELinux setelah me-reboot server sumber RHEL
- Jadwalkan pekerjaan untuk Amazon RDS untuk PostgreSQL dan Aurora PostgreSQL dengan menggunakan Lambda dan Secrets Manager

- Transportasi database PostgreSQL antara dua instans Amazon RDS DB menggunakan pg_transport
- Membongkar data dari klaster Amazon Redshift di seluruh akun ke Amazon S3

Oracle

- Tambahkan HA ke Oracle PeopleSoft di Amazon RDS Custom dengan menggunakan replika baca
- Mengkonversi kueri JSON Oracle ke database PostgreSQL SQL
- Mengkonversi VARCHAR2 (1) tipe data untuk Oracle ke tipe data Boolean untuk Amazon Aurora PostgreSQL
- Meniru Oracle DR dengan menggunakan database global Aurora yang kompatibel dengan PostgreSQL
- Meniru array PL/SQL asosiatif Oracle di Amazon Aurora PostgreSQL dan Amazon RDS untuk PostgreSQL
- Perkirakan ukuran mesin Amazon RDS untuk database Oracle dengan menggunakan laporan AWR
- Menangani blok anonim dalam pernyataan SQL Dinamis di Aurora PostgreSQL
- Bermigrasi secara bertahap dari Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL menggunakan Oracle SQL Developer dan AWS SCT
- Muat file BLOB ke TEXT dengan menggunakan pengkodean file di Aurora PostgreSQL yang kompatibel
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dalam mode SSL dengan menggunakan AWS DMS
- Migrasikan database Amazon RDS for Oracle ke database Akun AWS lain Wilayah AWS dan gunakan untuk replikasi yang sedang berlangsung AWS DMS
- Memigrasikan database Oracle lokal ke Amazon menggunakan Oracle Data EC2 Pump
- Memigrasi database Oracle lokal ke Amazon OpenSearch Service menggunakan Logstash
- Memigrasikan database Oracle lokal ke Amazon RDS for MySQL menggunakan AWS DMS dan AWS SCT
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle dengan menggunakan Impor Oracle Data Pump langsung melalui tautan database
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle menggunakan Oracle Data Pump
- Memigrasikan database Oracle lokal ke Amazon RDS for PostgreSQL menggunakan Oracle bystander dan AWS DMS
- Memigrasi database Oracle lokal ke Oracle di Amazon EC2

- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS untuk MariaDB menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS for Oracle menggunakan AWS DMS
- Migrasikan database Oracle ke Amazon DynamoDB menggunakan AWS DMS
- Migrasikan database Oracle ke Amazon RDS for Oracle dengan menggunakan adaptor file datar Oracle GoldenGate
- Migrasikan Database Oracle ke Amazon Redshift menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle ke Aurora PostgreSQL menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle JD Edwards EnterpriseOne ke AWS dengan menggunakan Oracle Data Pump dan AWS DMS
- Migrasikan tabel partisi Oracle ke PostgreSQL menggunakan AWS DMS
- Migrasikan PeopleSoft database Oracle ke AWS dengan menggunakan AWS DMS
- Memigrasi data dari database Oracle lokal ke Aurora PostgreSQL
- Bermigrasi dari Amazon RDS for Oracle ke Amazon RDS untuk MySQL
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk PostgreSQL menggunakan tampilan terwujud dan AWS DMS
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Amazon RDS for PostGresQL dan AWS DMS SharePlex
- Migrasi dari Oracle Database ke Amazon RDS for PostgreSQL dengan menggunakan Oracle GoldenGate
- Bermigrasi dari Oracle di Amazon EC2 ke Amazon RDS untuk MySQL menggunakan AWS DMS dan AWS SCT
- Bermigrasi dari Oracle WebLogic ke Apache Tomcat (ToMee) di Amazon ECS
- Migrasikan indeks berbasis fungsi dari Oracle ke PostgreSQL
- Migrasi aplikasi lama dari Oracle Pro*C ke ECPG
- Migrasikan nilai Oracle CLOB ke baris individual di PostgreSQL di AWS
- Migrasikan kode kesalahan Oracle Database ke database yang kompatibel dengan Amazon Aurora PostgreSQL
- Migrasikan fungsi asli Oracle ke PostgreSQL menggunakan ekstensi
- Migrasi Oracle ke PeopleSoft Amazon RDS Kustom
- Migrasikan fungsionalitas Oracle ROWID ke PostgreSQL di AWS

- Migrasikan paket pragma Oracle SERIALLY_REUSABLE ke PostgreSQL
- Migrasikan kolom virtual yang dihasilkan dari Oracle ke PostgreSQL
- Memantau GoldenGate log Oracle dengan menggunakan Amazon CloudWatch
- Menganalisis dependensi objek untuk migrasi database sebagian dari Oracle ke PostgreSQL
- Replatform Oracle Database Enterprise Edition ke Edisi Standar 2 di Amazon RDS for Oracle
- Menyiapkan HA/DR arsitektur untuk Oracle E-Business Suite di Amazon RDS Custom dengan database siaga aktif
- Siapkan fungsionalitas Oracle UTL_FILE pada Aurora PostgreSQL yang kompatibel
- Peran transisi untuk PeopleSoft aplikasi Oracle di Amazon RDS Custom for Oracle
- Validasi objek database setelah bermigrasi dari Oracle ke Amazon Aurora PostgreSQL

SAP

- Secara otomatis mencadangkan database SAP HANA menggunakan Systems Manager dan EventBridge
- Memigrasikan database SAP ASE lokal ke Amazon EC2
- Bermigrasi dari SAP ASE ke Amazon RDS for SQL Server menggunakan AWS DMS
- Migrasikan SAP ASE di Amazon EC2 ke Amazon Aurora PostgreSQL yang kompatibel dengan menggunakan AWS SCT dan AWS DMS
- Migrasikan SAP HANA ke AWS menggunakan SAP HSR dengan nama host yang sama
- Siapkan pemulihan bencana untuk SAP di IBM Db2 di AWS

Lebih banyak pola

- Akses database Amazon Neptunus dari wadah Amazon EKS
- Akses, kueri, dan gabungkan tabel Amazon DynamoDB menggunakan Athena
- Izinkan EC2 instance menulis akses ke bucket S3 di akun AMS
- Analisis dan visualisasikan data JSON bersarang dengan Amazon Athena dan Amazon Quick Sight
- Mengotomatiskan pencadangan untuk Amazon RDS untuk instans DB Amazon RDS for PostgreSQL dengan menggunakan AWS Batch
- Secara otomatis mengarsipkan item ke Amazon S3 menggunakan DynamoDB TTL
- Secara otomatis memulihkan instans dan klaster Amazon RDS DB yang tidak terenkripsi
- Secara otomatis berhenti dan memulai instans Amazon RDS DB menggunakan Windows AWS Systems Manager Pemeliharaan
- Bangun AWS landing zone yang mencakup MongoDB Atlas
- Membangun program COBOL Db2 dengan menggunakan dan AWS Mainframe Modernization AWS CodeBuild
- Membangun mesh data perusahaan dengan Amazon DataZone, AWS CDK, dan AWS CloudFormation
- Ubah aplikasi Python dan Perl untuk mendukung migrasi database dari Microsoft SQL Server ke Amazon Aurora PostgreSQL Edisi yang kompatibel
- Mengonversi dan membongkar data EBCDIC ke ASCII di AWS dengan menggunakan Python
- Mengonversi fitur temporal Teradata NORMALIZE ke Amazon Redshift SQL
- Mengonversi fitur Teradata RESET WHEN ke Amazon Redshift SQL
- Mengkonversi VARCHAR2 (1) tipe data untuk Oracle ke tipe data Boolean untuk Amazon Aurora PostgreSQL
- Buat pengguna dan peran aplikasi di Aurora PostgreSQL yang kompatibel
- Buat CloudFormation templat AWS untuk tugas AWS DMS menggunakan Microsoft Excel dan Python
- Mengirimkan data DynamoDB ke Amazon S3 menggunakan Kinesis Data Streams dan Firehose dengan AWS CDK
- Terapkan cluster Cassandra di Amazon EC2 dengan statis IPs pribadi untuk menghindari penyeimbangan kembali

- Menerapkan cluster CockroachDB di Amazon EKS dengan menggunakan Terraform
- Kembangkan asisten berbasis obrolan AI generatif tingkat lanjut dengan menggunakan RAG dan prompt ReAct
- Meniru Oracle DR dengan menggunakan database global Aurora yang kompatibel dengan PostgreSQL
- Aktifkan pengarsipan DB2 log langsung ke Amazon S3 dalam database IBM Db2
- Aktifkan enkripsi data transparan di Amazon RDS for SQL Server
- Ekspor database Microsoft SQL Server ke Amazon S3 dengan menggunakan AWS DMS
- Menerapkan SHA1 hashing untuk data PII saat bermigrasi dari SQL Server ke PostgreSQL
- Bermigrasi secara bertahap dari Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL menggunakan Oracle SQL Developer dan AWS SCT
- Muat file BLOB ke TEXT dengan menggunakan pengkodean file di Aurora PostgreSQL yang kompatibel
- Kelola kredensil menggunakan AWS Secrets Manager
- Migrasikan database Db2 dari Amazon ke EC2 Aurora MySQL yang kompatibel dengan menggunakan AWS DMS
- Migrasi database Microsoft SQL Server dari Amazon EC2 ke Amazon DocumentDB dengan menggunakan AWS DMS
- Migrasikan database Microsoft SQL Server ke Aurora MySQL dengan menggunakan AWS DMS dan AWS SCT
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dalam mode SSL dengan menggunakan AWS DMS
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dengan dan menggunakan dan AWS SCT AWS DMS AWS CLI CloudFormation
- Migrasikan instans Amazon RDS DB ke VPC atau akun lain
- Migrasikan database Amazon RDS for Oracle ke database Akun AWS lain Wilayah AWS dan gunakan untuk replikasi yang sedang berlangsung AWS DMS
- Migrasi cluster Amazon Redshift ke Wilayah AWS di Tiongkok
- Memigrasi database MariaDB lokal ke Amazon RDS for MariaDB menggunakan alat bawaan
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server

- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan server tertaut
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan metode pencadangan dan pemulihan asli
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan AWS DMS
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT
- Memigrasi database Microsoft SQL Server lokal ke Microsoft SQL Server di Amazon yang menjalankan Linux EC2
- Memigrasikan database MySQL lokal ke Amazon RDS for MySQL
- Memigrasikan database MySQL lokal ke Aurora MySQL
- Memigrasikan database Oracle lokal ke Amazon menggunakan Oracle Data EC2 Pump
- Memigrasi database Oracle lokal ke Amazon OpenSearch Service menggunakan Logstash
- Memigrasikan database Oracle lokal ke Amazon RDS for MySQL menggunakan AWS DMS dan AWS SCT
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle dengan menggunakan Impor Oracle Data Pump langsung melalui tautan database
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle menggunakan Oracle Data Pump
- Memigrasikan database Oracle lokal ke Amazon RDS for PostgreSQL menggunakan Oracle bystander dan AWS DMS
- Memigrasi database Oracle lokal ke Oracle di Amazon EC2
- Memigrasikan database PostgreSQL lokal ke Aurora PostgreSQL
- Memigrasikan database SAP ASE lokal ke Amazon EC2
- Memigrasi database ThoughtSpot Falcon lokal ke Amazon Redshift
- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS untuk MariaDB menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS for Oracle menggunakan AWS DMS
- Migrasikan database Oracle ke Amazon RDS for Oracle dengan menggunakan adaptor file datar Oracle GoldenGate
- Migrasikan Database Oracle ke Amazon Redshift menggunakan AWS DMS dan AWS SCT

- Migrasikan database Oracle ke Aurora PostgreSQL menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle JD Edwards EnterpriseOne ke AWS dengan menggunakan Oracle Data Pump dan AWS DMS
- Migrasikan tabel partisi Oracle ke PostgreSQL menggunakan AWS DMS
- Migrasikan PeopleSoft database Oracle ke AWS dengan menggunakan AWS DMS
- Memigrasikan database Couchbase Server ke Amazon EC2
- Memigrasi data dari database Oracle lokal ke Aurora PostgreSQL
- Migrasi data ke AWS Cloud dengan menggunakan Starburst
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan menggunakan pengiriman log untuk mengurangi waktu pemadaman
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan pemulihan bencana ketersediaan tinggi
- Bermigrasi dari Amazon RDS for Oracle ke Amazon RDS untuk MySQL
- Migrasi dari Couchbase Server ke Couchbase Capella di AWS
- Bermigrasi dari IBM Db2 di Amazon EC2 ke Aurora PostgreSQL yang kompatibel dengan menggunakan AWS DMS dan AWS SCT
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk PostgreSQL menggunakan tampilan terwujud dan AWS DMS
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Amazon RDS for PostGresQL dan AWS DMS SharePlex
- Migrasi dari Oracle Database ke Amazon RDS for PostgreSQL dengan menggunakan Oracle GoldenGate
- Bermigrasi dari Oracle di Amazon EC2 ke Amazon RDS untuk MySQL menggunakan AWS DMS dan AWS SCT
- Bermigrasi dari PostgreSQL di Amazon ke Amazon RDS untuk PostgreSQL menggunakan pglogical EC2
- Bermigrasi dari SAP ASE ke Amazon RDS for SQL Server menggunakan AWS DMS
- Migrasikan indeks berbasis fungsi dari Oracle ke PostgreSQL
- Migrasi aplikasi lama dari Oracle Pro*C ke ECPG
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2 menggunakan Layanan Migrasi Aplikasi
- Memigrasikan beban kerja Cloudera lokal ke Platform Data Cloudera di AWS

- Migrasikan database MySQL lokal ke Aurora MySQL menggunakan Percona, Amazon EFS, dan Amazon S3 XtraBackup
- Migrasikan Oracle Business Intelligence 12c ke AWS Cloud dari server lokal
- Migrasikan nilai Oracle CLOB ke baris individual di PostgreSQL di AWS
- Migrasikan kode kesalahan Oracle Database ke database yang kompatibel dengan Amazon Aurora PostgreSQL
- Migrasikan tabel eksternal Oracle ke Amazon Aurora PostgreSQL yang kompatibel
- Migrasikan fungsi asli Oracle ke PostgreSQL menggunakan ekstensi
- Migrasi Oracle ke PeopleSoft Amazon RDS Kustom
- Migrasikan fungsionalitas Oracle ROWID ke PostgreSQL di AWS
- Migrasikan paket pragma Oracle SERIALLY_REUSABLE ke PostgreSQL
- Migrasikan SAP ASE di Amazon EC2 ke Amazon Aurora PostgreSQL yang kompatibel dengan menggunakan AWS SCT dan AWS DMS
- Migrasikan kolom virtual yang dihasilkan dari Oracle ke PostgreSQL
- Siapkan ruang data minimum yang layak untuk berbagi data antar organisasi
- Pantau ElastiCache kluster Amazon untuk enkripsi saat istirahat
- Kueri tabel Amazon DynamoDB dengan SQL dengan menggunakan Amazon Athena
- Mengamankan dan merampingkan akses pengguna dalam database federasi Db2 di AWS dengan menggunakan konteks tepercaya
- Siapkan PeopleSoft arsitektur yang sangat tersedia di AWS
- Siapkan fungsionalitas Oracle UTL_FILE pada Aurora PostgreSQL yang kompatibel
- Merampingkan penerapan PostgreSQL di Amazon EKS dengan menggunakan PGO
- Transfer z/OS data Db2 skala besar ke Amazon S3 dalam file CSV
- Transportasi database PostgreSQL antara dua instans Amazon RDS DB menggunakan pg_transport
- Validasi objek database setelah bermigrasi dari Oracle ke Amazon Aurora PostgreSQL

Penyimpanan & cadangan

Topik

- Izinkan EC2 instance menulis akses ke bucket S3 di akun AMS
- Mengotomatiskan konsumsi aliran data ke dalam database Snowflake dengan menggunakan Snowflake Snowpipe, Amazon S3, Amazon SNS, dan Amazon Data Firehose
- Secara otomatis mengenkripsi volume Amazon EBS yang ada dan baru
- Cadangkan server Sun SPARC di emulator Stromasys Charon-SSP di AWS Cloud
- Cadangkan dan arsipkan data ke Amazon S3 dengan Veeam Backup & Replication
- Aktifkan pengarsipan DB2 log langsung ke Amazon S3 dalam database IBM Db2
- Memigrasikan data dari lingkungan Hadoop lokal ke Amazon S3 menggunakan AWS untuk Amazon S3 DistCp PrivateLink
- Lebih banyak pola

Izinkan EC2 instance menulis akses ke bucket S3 di akun AMS

Mansi Suratwala, Amazon Web Services

Ringkasan

AWS Managed Services (AMS) membantu Anda mengoperasikan AWS infrastruktur dengan lebih efisien dan aman. Akun AMS memiliki pagar keamanan untuk administrasi standar sumber daya Anda. AWS Salah satu pagar pembatas adalah bahwa profil instans Amazon Elastic Compute Cloud (Amazon EC2) default tidak mengizinkan akses tulis ke bucket Amazon Simple Storage Service (Amazon S3). Namun, organisasi Anda mungkin memiliki beberapa bucket S3 dan memerlukan kontrol lebih besar atas akses menurut EC2 instans. Misalnya, Anda mungkin ingin menyimpan cadangan database dari EC2 instance di bucket S3.

Pola ini menjelaskan cara menggunakan request for change (RFCs) agar EC2 instans Anda dapat menulis akses ke bucket S3 di akun AMS Anda. RFC adalah permintaan yang dibuat oleh Anda atau AMS untuk membuat perubahan di lingkungan terkelola Anda dan yang mencakup ID tipe perubahan (CT) untuk operasi tertentu.

Prasyarat dan batasan

Prasyarat

- Akun AMS Advanced. Untuk informasi selengkapnya tentang hal ini, lihat rencana operasi AMS di dokumentasi AMS.
- Akses ke `customer-mc-user-role` peran AWS Identity and Access Management (IAM) untuk RFCs dikirimkan.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi dengan EC2 instance di akun AMS Anda.
- Pemahaman tentang cara membuat dan mengirimkan RFCs di AMS. Untuk informasi selengkapnya tentang hal ini, lihat Apa itu jenis perubahan AMS? dalam dokumentasi AMS.
- Pemahaman tentang jenis perubahan manual dan otomatis (CTs). Untuk informasi selengkapnya tentang ini, lihat Otomatis dan manual CTs dalam dokumentasi AMS.

Arsitektur

Tumpukan teknologi

- AMS
- AWS CLI
- Amazon EC2
- Amazon S3
- IAM

Alat

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Managed Services (AMS) membantu Anda mengoperasikan infrastruktur AWS dengan lebih efisien dan aman.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

Epik

Buat bucket S3 dengan RFC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 dengan menggunakan RFC otomatis.	1. Masuk ke akun AMS Anda, pilih halaman Pilih ubah jenis, pilih RFCs, lalu pilih Buat RFC. 2. Kirimkan RFC otomatis Create S3 Bucket.	Administrator sistem AWS, pengembang AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Pastikan Anda merekam nama bucket S3.	

Buat profil instans IAM dan kaitkan dengan instance EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim RFC manual untuk membuat peran IAM.	Saat akun AMS di-onboard, profil instans IAM default yang bernama `customer-mc-ec2-instance-profile` dibuat dan dikaitkan dengan setiap EC2 instance di akun AMS Anda. Namun, profil instance tidak memiliki izin menulis ke bucket S3 Anda. Untuk menambahkan izin tulis, kirimkan manual Create IAM Resource RFC untuk membuat peran IAM yang memiliki tiga kebijakan berikut:`customer_ec2_instance_` ,, dan. `customer_deny_policy` `customer_ec2_s3_integration_policy`	Administrator sistem AWS, pengembang AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important `customer_deny_policy` Kebijakan `customer_ec2_instance_` dan kebijakan sudah ada di akun AMS Anda. Namun, Anda perlu membuat `customer_ec2_s3_integration_policy` dengan menggunakan kebijakan sampel berikut: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "", "Effect": "Allow", "Principal": { "Service": "ec2.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>Role Permissions:
	{
	     "Version":
	 "2012-10-17",
	     "Statement": [
	        {
	             "Action": [
	
	 "s3:ListBucket",
	
	 "s3:GetBucketLocat
	ion"
	            ],
	             "Resource
	": "arn:aws:s3:::",
	             "Effect":
	 "Allow"
	        },
	        {
	             "Action": [
	
	 "s3:GetObject",
	
	 "s3:PutObject",
	
	 "s3:ListMultipartU
	ploadParts",
	
	 "s3:AbortMultipart
	Upload"
	            ],
	             "Resource
	": "arn:aws:s3:::/*",
	             "Effect":
	 "Allow"
	        }
	    ]
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim RFC manual untuk mengganti profil instans IAM.	Kirimkan RFC manual untuk mengaitkan EC2 instance target dengan profil instans IAM yang baru.	Administrator sistem AWS, pengembang AWS
Uji operasi penyalinan ke bucket S3.	Uji operasi salin ke bucket S3 dengan menjalankan perintah berikut di AWS CLI: `aws s3 cp test.txt s3://<S3 bucket>/test2.txt`	Administrator sistem AWS, pengembang AWS

Sumber daya terkait

- Buat profil instans IAM untuk instans Amazon EC2 Anda
- Membuat bucket S3 (menggunakan konsol Amazon S3 AWS SDKs,, atau) AWS CLI

Mengotomatiskan konsumsi aliran data ke dalam database Snowflake dengan menggunakan Snowflake Snowpipe, Amazon S3, Amazon SNS, dan Amazon Data Firehose

Bikash Chandra Rout, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana Anda dapat menggunakan layanan di Amazon Web Services (AWS) Cloud untuk memproses aliran data yang berkelanjutan dan memuatnya ke database Snowflake. Pola ini menggunakan Amazon Data Firehose untuk mengirimkan data ke Amazon Simple Storage Service (Amazon S3), Amazon Simple Notification Service (Amazon SNS) untuk mengirim notifikasi saat data baru diterima, dan Snowflake Snowpipe untuk memuat data ke database Snowflake.

Dengan mengikuti pola ini, Anda dapat terus menghasilkan data yang tersedia untuk analisis dalam hitungan detik, menghindari beberapa `COPY` perintah manual, dan memiliki dukungan penuh untuk data semi-terstruktur saat dimuat.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Sumber data yang terus mengirim data ke aliran pengiriman Firehose.
- Bucket S3 yang sudah ada yang menerima data dari aliran pengiriman Firehose.
- Akun Snowflake yang aktif.

Batasan

- Snowflake Snowpipe tidak terhubung langsung ke Firehose.

Arsitektur

Tumpukan teknologi

- Amazon Data Firehose
- Amazon SNS

- Amazon S3
- Pipa Salju Kepingan Salju
- Database kepingan salju

Alat

- Amazon Data Firehose adalah layanan yang dikelola sepenuhnya untuk mengirimkan data streaming real-time ke tujuan seperti Amazon S3, Amazon Redshift, OpenSearch Amazon Service, Splunk, dan titik akhir HTTP kustom atau titik akhir HTTP apa pun yang dimiliki oleh penyedia layanan pihak ketiga yang didukung.
- Amazon Simple Storage Service (Amazon S3) adalah penyimpanan untuk internet.
- Amazon Simple Notification Service (Amazon SNS) mengoordinasikan dan mengelola penyampaian atau pengiriman pesan ke titik akhir atau klien yang berlangganan .
- Snowflake — Snowflake adalah gudang data analitik yang disediakan sebagai (SaaS). Software-as-a-Service
- Snowflake Snowpipe — Snowpipe memuat data dari file segera setelah tersedia dalam tahap Snowflake.

Epik

Siapkan Snowflake Snowpipe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file CSV di Snowflake.	Masuk ke Snowflake dan jalankan `CREATE FILE FORMAT` perintah untuk membuat file CSV dengan pembatas bidang tertentu. Untuk informasi lebih lanjut tentang ini dan perintah Snowflake lainnya, lihat bagian Informasi tambahan.	Developer
Buat panggung Snowflake eksternal.	Jalankan `CREATE STAGE` perintah untuk membuat	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tahap Snowflake eksternal yang mereferensikan file CSV yang Anda buat sebelumnya. Penting: Anda memerlukan URL untuk bucket S3, kunci AWS akses, dan kunci akses AWS rahasia Anda. Jalankan `SHOW STAGES` perintah untuk memverifikasi bahwa tahap Snowflake dibuat.	
Buat tabel target Snowflake.	Jalankan `CREATE TABLE` perintah untuk membuat tabel Snowflake.	Developer
Buat pipa.	Jalankan `CREATE PIPE` perintah; pastikan itu `auto_ingest=true` ada di perintah. Jalankan `SHOW PIPES` perintah untuk memverifikasi bahwa pipa dibuat. Salin dan simpan nilai `notification_channel` kolom. Nilai ini akan digunakan untuk mengonfigurasi pemberitahuan acara Amazon S3.	Developer

Konfigurasikan bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan siklus hidup 30 hari untuk bucket S3.	Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3. Pilih bucket	Administrator Sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	S3 yang berisi data dari Firehose. Kemudian pilih tab Manajemen di bucket S3 dan pilih Tambahkan aturan siklus hidup. Masukkan nama aturan Anda di kotak dialog Aturan Siklus Hidup, dan konfigura sikan aturan siklus hidup 30 hari untuk bucket Anda. Untuk bantuan tentang ini dan cerita lainnya, lihat bagian Sumber daya terkait.	
Buat kebijakan IAM untuk bucket S3.	Buka konsol AWS Identity and Access Managemen t (IAM) dan pilih Kebijakan . Pilih Buat kebijakan, dan pilih tab JSON. Salin dan tempel kebijakan dari bagian Informasi tambahan ke bidang JSON. Kebijakan ini akan memberikan `PutObject` dan `DeleteObject` izin, serta `GetObject` `GetObjectVersion` , dan `ListBucket` izin. Pilih Kebijakan ulasan, masukkan nama kebijakan, lalu pilih Buat kebijakan.	Administrator Sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan kebijakan ke peran IAM.	Buka konsol IAM, pilih Peran, lalu pilih Buat peran. Pilih akun AWS lain sebagai entitas tepercaya. Masukkan Akun AWS ID Anda, dan pilih Memerlukan ID eksternal . Masukkan ID placeholder yang akan Anda ubah nanti. Pilih Berikutnya, dan tetapkan kebijakan IAM yang Anda buat sebelumnya. Kemudian buat peran IAM.	Administrator Sistem, Pengembang
Salin Nama Sumber Daya Amazon (ARN) untuk peran IAM.	Buka konsol IAM, dan pilih Peran. Pilih peran IAM yang Anda buat sebelumnya, lalu salin dan simpan ARN Peran.	Administrator Sistem, Pengembang

Siapkan integrasi penyimpanan di Snowflake

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat integrasi penyimpanan di Snowflake.	Masuk ke Snowflake dan jalankan perintah. `CREATE STORAGE INTEGRATION` Ini akan mengubah hubungan tepercaya, memberikan akses ke Snowflake, dan memberikan ID eksternal untuk tahap Snowflake Anda.	Administrator Sistem, Pengembang
Ambil peran IAM untuk akun Snowflake Anda.	Jalankan `DESC INTEGRATION` perintah untuk mengambil ARN untuk peran IAM.	Administrator Sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important `<integration_name>` adalah nama integrasi penyimpanan Snowflake yang Anda buat sebelumnya.	
Rekam dua nilai kolom.	Salin dan simpan nilai untuk `storage_aws_external_id` kolom `storage_aws_iam_user_arn` dan.	Administrator Sistem, Pengembang

Izinkan Snowflake Snowpipe mengakses bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah kebijakan peran IAM.	Buka konsol IAM dan pilih Peran. Pilih peran IAM yang Anda buat sebelumnya dan pilih tab Trust relationship. Pilih Edit trust relationship (Edit Hubungan Kepercayaan). Ganti `snowflake_external_id` dengan `storage_aws_external_id` nilai yang Anda salin sebelumnya. Ganti `snowflake_user_arn` dengan `storage_aws_iam_user_arn` nilai yang Anda salin sebelumny	Administrator Sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	a. Kemudian pilih Perbarui kebijakan kepercayaan.	

Aktifkan dan konfigurasikan notifikasi SNS untuk bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan notifikasi acara untuk bucket S3.	Buka konsol Amazon S3 dan pilih bucket Anda. Pilih Properti, dan di bawah Pengaturan lanjutan s, pilih Acara. Pilih Tambahkan pemberitahuan, dan masukkan nama untuk acara ini. Jika Anda tidak memasukkan nama, pengident ifikasi unik global (GUID) akan digunakan.	Administrator Sistem, Pengembang
Konfigurasikan notifikasi Amazon SNS untuk bucket S3.	Di bawah Peristiwa, pilih ObjectCreate (Semua), lalu pilih SQS Queue di daftar dropdown Kirim ke. Dalam daftar SNS, pilih Add SQS queue ARN, dan paste `notification_chann el` nilai yang Anda salin sebelumnya. Lalu, pilih Simpan.	Administrator Sistem, Pengembang
Berlangganan antrian Snowflake SQS ke topik SNS.	Berlangganan antrian Snowflake SQS ke topik SNS yang Anda buat. Untuk bantuan dengan langkah ini,	Administrator Sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lihat bagian Sumber daya terkait.	

Periksa integrasi tahap Snowflake

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa dan uji Snowpipe.	Masuk ke Snowflake dan buka panggung Snowflake. Jatuhkan file ke dalam ember S3 Anda dan periksa apakah tabel Snowflake memuatnya . Amazon S3 akan mengirim notifikasi SNS ke Snowpipe saat objek baru muncul di bucket S3.	Administrator Sistem, Pengembang

Sumber daya terkait

- Mengelola siklus hidup penyimpanan
- Berlangganan Antrian Snowflake SQS ke Topik Amazon SNS

Informasi tambahan

Buat format file:

```
CREATE FILE FORMAT <name>
TYPE = 'CSV'
FIELD_DELIMITER = '|'
SKIP_HEADER = 1;
```

Buat tahap eksternal:

```
externalStageParams (for Amazon S3) ::=
```

```
  URL = 's3://[//]'

  [ { STORAGE_INTEGRATION =  } | { CREDENTIALS = ( {  { AWS_KEY_ID = `` AWS_SECRET_KEY
 = `` [ AWS_TOKEN = `` ] } | AWS_ROLE = ``  } ) ) }` ]
  [ ENCRYPTION = ( [ TYPE = 'AWS_CSE' ] [ MASTER_KEY = '' ] |
                   [ TYPE = 'AWS_SSE_S3' ] |
                   [ TYPE = 'AWS_SSE_KMS' [ KMS_KEY_ID = '' ] |
                   [ TYPE = NONE ] )
```

Buat tabel:

```
CREATE [ OR REPLACE ] [ { [ LOCAL | GLOBAL ] TEMP[ORARY] | VOLATILE } | TRANSIENT ]
 TABLE [ IF NOT EXISTS ]
  <table_name>
    ( <col_name> <col_type> [ { DEFAULT <expr>
                                | { AUTOINCREMENT | IDENTITY } [ ( <start_num> ,
 <step_num> ) | START <num> INCREMENT <num> ] } ]
                                 /* AUTOINCREMENT / IDENTITY supported only for numeric
 data types (NUMBER, INT, etc.) */
                               [ inlineConstraint ]
      [ , <col_name> <col_type> ... ]
      [ , outoflineConstraint ]
      [ , ... ] )
  [ CLUSTER BY ( <expr> [ , <expr> , ... ] ) ]
  [ STAGE_FILE_FORMAT = ( { FORMAT_NAME = '<file_format_name>'
                           | TYPE = { CSV | JSON | AVRO | ORC | PARQUET | XML }
 [ formatTypeOptions ] } ) ]
  [ STAGE_COPY_OPTIONS = ( copyOptions ) ]
  [ DATA_RETENTION_TIME_IN_DAYS = <num> ]
  [ COPY GRANTS ]
  [ COMMENT = '<string_literal>' ]
```

Tampilkan tahapan:

```
SHOW STAGES;
```

Buat pipa:

```
CREATE [ OR REPLACE ] PIPE [ IF NOT EXISTS ]
  [ AUTO_INGEST = [ TRUE | FALSE ] ]
  [ AWS_SNS_TOPIC =  ]
  [ INTEGRATION = '' ]
```

```
  [ COMMENT = '' ]
  AS
```

Tampilkan pipa:

```
SHOW PIPES [ LIKE '<pattern>' ]
           [ IN { ACCOUNT | [ DATABASE ] <db_name> | [ SCHEMA ] <schema_name> } ]
```

Buat integrasi penyimpanan:

```
CREATE STORAGE INTEGRATION <integration_name>
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = S3
  ENABLED = TRUE
  STORAGE_AWS_ROLE_ARN = '<iam_role>'
  STORAGE_ALLOWED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/')
  [ STORAGE_BLOCKED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/') ]
```

Contoh:

```
create storage integration s3_int
  type = external_stage
  storage_provider = s3
  enabled = true
  storage_aws_role_arn = 'arn:aws:iam::001234567890:role/myrole'
  storage_allowed_locations = ('s3://amzn-s3-demo-bucket1/mypath1/', 's3://amzn-s3-
demo-bucket2/mypath2/')
  storage_blocked_locations = ('s3://amzn-s3-demo-bucket1/mypath1/sensitivedata/',
 's3://amzn-s3-demo-bucket2/mypath2/sensitivedata/');
```

Untuk informasi selengkapnya tentang langkah ini, lihat Mengonfigurasi integrasi penyimpanan Snowflake untuk mengakses Amazon S3 dari dokumentasi Snowflake.

Jelaskan integrasi:

```
DESC INTEGRATION <integration_name>;
```

Kebijakan bucket S3:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "s3:PutObject",
              "s3:GetObject",
              "s3:GetObjectVersion",
              "s3:DeleteObject",
              "s3:DeleteObjectVersion"
            ],
            "Resource": "arn:aws:s3:::/*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::",
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "/*"
                    ]
                }
            }
        }
    ]
}
```

Secara otomatis mengenkripsi volume Amazon EBS yang ada dan baru

Tony DeMarco dan Josh Joy, Amazon Web Services

Ringkasan

Enkripsi volume Amazon Elastic Block Store (Amazon EBS) penting untuk strategi perlindungan data organisasi. Ini adalah langkah penting dalam membangun lingkungan yang dirancang dengan baik. Meskipun tidak ada cara langsung untuk mengenkripsi volume atau snapshot EBS yang tidak terenkripsi yang ada, Anda dapat mengenkripsi mereka dengan membuat volume atau snapshot baru. Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya EBS di dokumentasi Amazon EC2 . Pola ini menyediakan kontrol preventif dan detektif untuk mengenkripsi volume EBS Anda, baik yang baru maupun yang sudah ada. Dalam pola ini, Anda mengonfigurasi pengaturan akun, membuat proses remediasi otomatis, dan menerapkan kontrol akses.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi di macOS, Linux, atau Windows
- jq, diinstal dan dikonfigurasi di macOS, Linux, atau Windows
- Izin AWS Identity and Access Management (IAM) disediakan untuk memiliki akses baca dan tulis ke AWS, CloudFormation Amazon Elastic Compute Cloud (EC2Amazon), AWS Systems Manager, AWS Config, dan AWS Key Management Service (AWS KMS)
- AWS Organizations dikonfigurasi dengan semua fitur diaktifkan, persyaratan untuk kebijakan kontrol layanan
- AWS Config diaktifkan di akun target

Batasan

- Di akun AWS target Anda, tidak boleh ada aturan AWS Config bernama encrypted-volume. Solusi ini menerapkan aturan dengan nama ini. Aturan yang sudah ada sebelumnya dengan nama ini dapat menyebabkan penerapan gagal dan mengakibatkan biaya yang tidak perlu terkait dengan pemrosesan aturan yang sama lebih dari sekali.
- Solusi ini mengenkripsi semua volume EBS dengan kunci AWS KMS yang sama.

- Jika Anda mengaktifkan enkripsi volume EBS untuk akun, pengaturan ini khusus Wilayah. Jika Anda mengaktifkannya untuk Wilayah AWS, Anda tidak dapat menonaktifkannya untuk volume atau snapshot individual di Wilayah tersebut. Untuk informasi selengkapnya, lihat Enkripsi secara default di EC2 dokumentasi Amazon.
- Saat Anda memulihkan volume EBS yang sudah ada dan tidak terenkripsi, pastikan EC2 instans tidak digunakan. Otomatisasi ini mematikan instance untuk melepaskan volume yang tidak terenkripsi dan melampirkan yang terenkripsi. Ada downtime saat remediasi sedang berlangsung. Jika ini adalah bagian penting dari infrastruktur untuk organisasi Anda, pastikan bahwa konfigurasi ketersediaan tinggi manual atau otomatis tersedia agar tidak memengaruhi ketersediaan aplikasi apa pun yang berjalan pada instance. Kami menyarankan Anda memulihkan sumber daya penting hanya selama jendela pemeliharaan standar.

Arsitektur

Alur kerja otomatisasi

1. AWS Config mendeteksi volume EBS yang tidak terenkripsi.
2. Administrator menggunakan AWS Config untuk mengirim perintah remediasi ke Systems Manager.
3. Otomatisasi Systems Manager mengambil snapshot dari volume EBS yang tidak terenkripsi.
4. Otomatisasi Systems Manager menggunakan AWS KMS untuk membuat salinan snapshot terenkripsi.
5. Otomatisasi Systems Manager melakukan hal berikut:
   a. Menghentikan EC2 instance yang terpengaruh jika sedang berjalan
   b. Melampirkan salinan volume yang baru dan terenkripsi ke instance EC2
   c. Mengembalikan EC2 instance ke keadaan semula

Alat

Layanan AWS

- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) menyediakan akses langsung ke antarmuka pemrograman aplikasi publik APIs () layanan AWS. Anda dapat menjelajahi kemampuan layanan dengan AWS CLI dan mengembangkan skrip shell untuk mengelola sumber daya Anda. Selain perintah setara API tingkat rendah, beberapa layanan

AWS menyediakan penyesuaian untuk AWS CLI. Kustomisasi dapat mencakup perintah tingkat tinggi yang menyederhanakan penggunaan layanan dengan API yang kompleks.

- AWS CloudFormation — AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda. Anda membuat templat yang menjelaskan semua sumber daya AWS yang Anda inginkan (seperti EC2 instans Amazon), serta menyediakan serta CloudFormation mengonfigurasi sumber daya tersebut untuk Anda.
- AWS Config — AWS Config memberikan tampilan terperinci tentang konfigurasi sumber daya AWS di akun AWS Anda. Ini mencakup bagaimana sumber daya terkait satu sama lain dan bagaimana sumber daya tersebut dikonfigurasi di masa lalu sehingga Anda dapat melihat bagaimana konfigurasi dan hubungan berubah dari waktu ke waktu.
- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang dapat diubah ukurannya yang Anda gunakan untuk membangun dan meng-host sistem perangkat lunak Anda.
- AWS KMS — AWS Key Management Service (AWS KMS) adalah layanan enkripsi dan manajemen kunci yang diskalakan untuk cloud. Kunci dan fungsionalitas AWS KMS digunakan oleh layanan AWS lainnya, dan Anda dapat menggunakannya untuk melindungi data di lingkungan AWS Anda.
- AWS Organizations — AWS Organizations adalah layanan manajemen akun yang memungkinkan Anda mengkonsolidasikan beberapa akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Systems Manager Automation — Systems Manager Automation menyederhanakan tugas pemeliharaan dan penerapan umum untuk EC2 instans Amazon dan sumber daya AWS lainnya.

Layanan lainnya

- jq - jq adalah prosesor JSON command-line yang ringan dan fleksibel. Anda menggunakan alat ini untuk mengekstrak informasi kunci dari output AWS CLI.

Kode

- Kode untuk pola ini tersedia dalam Remediasi Volume EBS yang tidak terenkripsi GitHub secara otomatis menggunakan repositori kunci KMS pelanggan.

Epik

Mengotomatiskan remediasi volume yang tidak terenkripsi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh skrip dan CloudFormation templat.	Unduh skrip shell, file JSON, dan CloudFormation templat dari Volume EBS yang tidak terenkripsi GitHub secara otomatis menggunakan repositori kunci KMS pelanggan.	Administrator AWS, AWS Umum
Identifikasi administrator untuk kunci AWS KMS.	1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/. 2. Identifikasi pengguna atau peran yang akan menjadi administrator kunci AWS KMS. Jika pengguna atau peran baru perlu dibuat untuk tujuan ini, buat sekarang. Untuk informasi selengkapnya, lihat Identitas IAM dalam dokumentasi IAM. Otomatisasi ini membuat kunci AWS KMS baru. 3. Setelah diidentifikasi, salin Nama Sumber Daya Amazon (ARN) pengguna atau peran. Untuk informasi selengkapnya, lihat IAM ARNs dalam dokumenta	Administrator AWS, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	si IAM. Anda menggunakan ARN ini di langkah berikutnya.	
Menyebarkan template Stack1 CloudFormation .	1. Buka CloudFormation konsol AWS di https://console.aws.amazon.com/cloudformation/. 2. Di CloudFormation, gunakan `Stack1.yaml` template. Perhatikan detail penerapan berikut: • Berikan tumpukan nama yang jelas dan deskriptif. Perhatikan nama tumpukan karena Anda memerlukan nilai ini di langkah berikutnya. • Tempelkan ARN administrator kunci ke satu-satunya bidang parameter di Stack1. Pengguna atau peran ini menjadi administrator untuk kunci AWS KMS yang dibuat oleh tumpukan. Untuk informasi selengkapnya tentang penerapan CloudFormation templat, lihat Bekerja dengan CloudFormation templat AWS dalam CloudFormation dokumentasi.	Administrator AWS, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan template Stack2 CloudFormation .	Di CloudFormation, gunakan `Stack2.yaml` template. Perhatikan detail penerapan berikut: • Berikan tumpukan nama yang jelas dan deskriptif. • Untuk satu-satunya parameter Stack2, masukkan nama tumpukan yang Anda buat pada langkah sebelumnya. Ini memungkinkan Stack2 untuk mereferensikan kunci dan peran AWS KMS baru yang digunakan oleh tumpukan pada langkah sebelumnya.	Administrator AWS, AWS Umum
Buat volume yang tidak terenkripsi untuk pengujian.	Buat EC2 instance dengan volume EBS yang tidak terenkripsi. Untuk petunjukn ya, lihat Membuat volume Amazon EBS di EC2 dokumentasi Amazon. Jenis instance tidak masalah, dan akses ke instance tidak diperlukan. Anda dapat membuat instance t2.micro untuk tetap berada di tingkat gratis, dan Anda tidak perlu membuat key pair.	Administrator AWS, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aturan AWS Config.	1. Buka konsol AWS Config di. https://console.aws.amazon.com/config/ Pada halaman Aturan, pilih aturan volume terenkripsi. 2. Konfirmasikan bahwa instance pengujian baru yang tidak terenkripsi muncul di daftar sumber daya yang tidak sesuai. Jika volume tidak segera muncul, tunggu beberapa menit dan segarkan hasilnya. Aturan AWS Config mendeteksi perubahan sumber daya segera setelah instance dan volume dibuat. 3. Pilih sumber daya, lalu pilih Remediate. Anda dapat melihat kemajuan dan status remediasi di Systems Manager sebagai berikut: 1. Buka konsol AWS Systems Manager di https://console.aws.amazon.com/systems-manager/. 2. Pada panel navigasi, pilih Otomatisasi.	Administrator AWS, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pilih tautan ID Eksekusi untuk melihat langkah-l angkah dan statusnya.	
Konfigurasikan akun tambahan atau Wilayah AWS.	Jika diperlukan untuk kasus penggunaan Anda, ulangi epik ini untuk akun tambahan atau Wilayah AWS.	Administrator AWS, AWS Umum

Aktifkan enkripsi volume EBS tingkat akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip aktifkan.	1. Dalam bash shell, gunakan `cd` perintah untuk menavigasi ke repositori kloning. 2. Masukkan perintah berikut untuk menjalankan `enable-ebs-encrypt ion-for-account` skrip. `./Bash/enable-ebs- encryption-for-acc ount.sh`	Administrator AWS, AWS Umum, bash
Konfirmasikan pengaturan diperbarui.	1. Buka EC2 konsol Amazon di https://console.aw s.amazon.com/ec2/. 2. Di sisi kanan layar, di bawah Pengaturan, pilih Perlindungan dan keamanan data.	Administrator AWS, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di bawah bagian enkripsi EBS, konfirmasikan bahwa Selalu enkripsi volume EBS baru diaktifkan dan bahwa kunci enkripsi Default diatur ke ARN yang Anda tentukan sebelumnya. **Note** Jika pengaturan Always encrypt new EBS volume dimatikan atau kunci masih disetel alias/aws/ebs, konfirmasikan bahwa Anda masuk ke akun yang sama dan Wilayah AWS tempat Anda menjalankan skrip shell, dan periksa shell Anda untuk pesan kesalahan.	
Konfigurasikan akun tambahan atau Wilayah AWS.	Jika diperlukan untuk kasus penggunaan Anda, ulangi epik ini untuk akun tambahan atau Wilayah AWS.	Administrator AWS, AWS Umum

Mencegah pembuatan instance yang tidak terenkripsi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan kontrol layanan.	1. Buka konsol AWS Organizations di https://console.aws.amazon.com/organizations/v2/. 2. Buat kebijakan kontrol layanan baru. Untuk informasi selengkapnya, lihat Membuat kebijakan kontrol layanan di dokumentasi AWS Organizations. 3. Tambahkan konten `DenyUnencryptedEC2.json` ke kebijakan dan simpan. Anda mengunduh file JSON ini dari GitHub repositori di epik pertama. 4. Lampirkan kebijakan ini ke akar organisasi atau unit organisasi yang diperlukan (OUs). Untuk informasi selengkapnya, lihat Melampirkan dan melepaskan kebijakan kontrol layanan dalam dokumentasi AWS Organizations.	Administrator AWS, AWS Umum

Sumber daya terkait

Dokumentasi layanan AWS

- AWS CLI
- AWS Config
- AWS CloudFormation
- Amazon EC2
- AWS KMS
- AWS Organizations
- AWS Systems Manager Automation

Sumber daya lainnya

- jq manual (situs web jq)
- jq unduh () GitHub

Cadangkan server Sun SPARC di emulator Stromasys Charon-SSP di AWS Cloud

Kevin Yung dan Rohit Darji, Amazon Web Services

Luis Ramos, Stromasy

Ringkasan

Pola ini menyediakan empat opsi untuk mencadangkan server Sun Microsystems SPARC Anda setelah migrasi dari lingkungan lokal ke Amazon Web Services () Cloud.AWS Opsi pencadangan ini membantu Anda menerapkan rencana cadangan yang memenuhi tujuan titik pemulihan (RPO) organisasi Anda dan tujuan waktu pemulihan (RTO), menggunakan pendekatan otomatis, dan menurunkan biaya operasional Anda secara keseluruhan. Pola ini memberikan gambaran umum dari empat opsi cadangan dan langkah-langkah untuk mengimplementasikannya.

Jika Anda menggunakan server Sun SPARC yang dihosting sebagai tamu di emulator Stromasys Charon-SSP, Anda dapat menggunakan salah satu dari tiga opsi cadangan berikut:

- Opsi Backup 1: Stromasys virtual tape - Gunakan fitur pita virtual Charon-SSP untuk menyiapkan fasilitas cadangan di server Sun SPARC dan mengarsipkan file cadangan Anda ke Amazon Simple Storage Service (Amazon S3) menggunakan Otomasi.AWS Systems Manager
- Opsi cadangan 2: Snapshot Stromasys - Gunakan fitur snapshot Charon-SSP untuk menyiapkan fasilitas cadangan untuk server tamu Sun SPARC di Charon-SSP.
- Opsi cadangan 3: Snapshot volume Amazon Elastic Block Store (Amazon EBS) - Jika Anda meng-host emulator Charon-SSP di Amazon Elastic Compute Cloud (Amazon EC2), Anda dapat menggunakan snapshot volume Amazon EBS untuk membuat cadangan untuk sistem file Sun SPARC.

Jika Anda menggunakan server Sun SPARC yang dihosting sebagai tamu di perangkat keras dan Charon-SSP di EC2 Amazon, Anda dapat menggunakan opsi cadangan berikut:

- Opsi Backup 4: AWS Storage Gateway Virtual Tape Library (VTL) — Gunakan aplikasi cadangan dengan Storage Gateway VTL Tape Gateway untuk mencadangkan server Sun SPARC.

Jika Anda menggunakan server Sun SPARC yang dihosting sebagai zona bermerek di server Sun SPARC, Anda dapat menggunakan opsi cadangan 1, 2, dan 4.

Stromasys menyediakan perangkat lunak dan layanan untuk meniru sistem kritis SPARC, Alpha, VAX, dan PA-RISC lama. Untuk informasi lebih lanjut tentang migrasi ke AWS Cloud menggunakan emulasi Stromasys, lihat Rehosting SPARC, Alpha, atau sistem warisan lainnya ke dengan Stromasys di Blog. AWS AWS

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Server Sun SPARC yang ada.
- Lisensi yang ada untuk Charon-SSP. Lisensi untuk Charon-SSP tersedia dari AWS Marketplace dan lisensi untuk Stromasys Virtual Environment (VE) tersedia dari Stromasys. Untuk informasi lebih lanjut, hubungi penjualan Stromasys.
- Keakraban dengan server Sun SPARC dan cadangan Linux.
- Keakraban dengan teknologi emulasi Charon-SSP. Untuk informasi lebih lanjut tentang ini, lihat Emulasi server lama Stromasys di dokumentasi Stromasys.
- Jika Anda ingin menggunakan fasilitas rekaman virtual atau aplikasi cadangan untuk sistem file server Sun SPARC Anda, Anda harus membuat dan mengonfigurasi fasilitas cadangan untuk sistem file server Sun SPARC.
- Pemahaman tentang RPO dan RTO. Untuk informasi lebih lanjut tentang ini, lihat Tujuan pemulihan bencana dari whitepaper Reliability Pillar dalam dokumentasi Well-Architected AWS Framework.
- Untuk menggunakan opsi Backup 4, Anda harus memiliki yang berikut:
  - Aplikasi backup berbasis perangkat lunak yang mendukung Storage Gateway VTL Tape Gateway. Untuk informasi selengkapnya tentang ini, lihat Bekerja dengan perangkat VTL dalam dokumentasi. AWS Storage Gateway
  - Direktur Bacula atau aplikasi cadangan serupa, diinstal dan dikonfigurasi. Untuk informasi lebih lanjut tentang ini, lihat dokumentasi Direktur Bacula.

Tabel berikut memberikan informasi tentang empat opsi cadangan dalam pola ini.

Opsi Backup	Mencapai konsistensi crash?	Mencapai konsistensi aplikasi?	Solusi alat cadangan virtual?	Kasus penggunaan yang khas

Opsi 1 - Pita virtual Stromasys	Ya Anda dapat mengotomatiskan snapshot sistem file Sun SPARC untuk mencadangkan data dalam rekaman virtual. Misalnya, Anda dapat menggunakan snapshot UFS atau ZFS.	Ya Opsi pencadangan ini memerlukan skrip otomatis untuk membersihkan transaksi dalam penerbangan, mengonfigurasi mode offline hanya-baca atau sementara selama snapshot sistem file, atau mengambil dump data aplikasi. Anda mungkin juga memerlukan waktu henti aplikasi atau mode hanya-baca.	Ya	Pencadangan sistem file server Sun SPARC dengan file.tar atau .zip Pencadangan data aplikasi

Opsi 2 - Snapshot Stromasys	Ya Anda harus mengkonfigurasi Charon-SSP Manager atau menggunakan argumen startup baris perintah untuk mengaktifkan fitur ini. Anda juga harus menjalankan perintah Linux untuk meminta emulator Charon-SSP untuk menyimpan status server tamu Sun SPARC ke dalam file snapshot. **Important** Anda harus mematikan server tamu Sun SPARC.	Ya Opsi cadangan ini membuat snapshot dari server tamu yang ditiru, termasuk disk virtual dan dump memori. **Important** Anda harus mematikan server tamu Sun SPARC selama snapshot.	Tidak	Cuplikan server Sun SPARC Pencadangan data aplikasi

Opsi 3 - Snapshot volume Amazon EBS	Ya Anda dapat menggunakan AWS Backup untuk mengotomatiskan snapshot Amazon EBS.	Ya Opsi pencadangan ini memerlukan skrip otomatis untuk membersihkan transaksi dalam penerbangan dan mengonfigurasi penghentian instans Amazon hanya-baca atau sementara EC2 selama snapshot volume Amazon EBS. **Important** Opsi pencadangan ini mungkin memerlukan waktu henti aplikasi atau mode hanya-baca untuk mencapai konsisten	Tidak	Snapshot sistem file server Sun SPARC Pencadangan data aplikasi

si aplikasi.

Opsi 4 - AWS Storage Gateway VTL	Ya Anda dapat secara otomatis mencadangkan data cadangan sistem file Sun SPARC ke VTL dengan menggunakan agen cadangan.	Ya Opsi pencadangan ini memerlukan skrip otomatis untuk membersihkan transaksi dalam penerbangan dan mengonfigurasi mode offline hanya-baca atau sementara selama snapshot sistem file atau dump data aplikasi. **Important** Opsi pencadangan ini mungkin memerlukan waktu henti aplikasi atau mode hanya-baca.	Ya	Armada besar cadangan sistem file server Sun SPARC Pencadangan data aplikasi

Batasan

- Anda dapat menggunakan pendekatan pola ini untuk mencadangkan server Sun SPARC individual, tetapi Anda juga dapat menggunakan opsi cadangan ini untuk data bersama jika Anda memiliki aplikasi yang berjalan di cluster.

Alat

Opsi Backup 1: Pita virtual Stromasys

- Emulator Stromasys Charon-SSP menciptakan replika virtual dari perangkat keras SPARC asli di dalam sistem komputer standar 64-bit x86 yang kompatibel. Ini menjalankan kode biner SPARC asli, termasuk sistem operasi (OSs) seperti SunOS atau Solaris, produk berlapis mereka, dan aplikasi.
- Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang dapat diubah ukurannya yang Anda gunakan untuk membangun dan meng-host sistem perangkat lunak Anda.
- Amazon Elastic File System (Amazon EFS) menyediakan sistem file yang sederhana, tanpa server, set-and-forget elastis untuk digunakan dengan Layanan AWS dan sumber daya lokal.
- Amazon Simple Storage Service (Amazon S3) adalah penyimpanan untuk internet.
- AWS Systems Manager Otomatisasi menyederhanakan tugas pemeliharaan dan penyebaran umum EC2 instans Amazon dan sumber daya lainnya. AWS

Opsi Backup 2: Snapshot Stromasys

- Emulator Stromasys Charon-SSP menciptakan replika virtual dari perangkat keras SPARC asli di dalam sistem komputer standar 64-bit x86 yang kompatibel. Ini menjalankan kode biner SPARC asli, termasuk OSs seperti SunOS atau Solaris, produk berlapis mereka, dan aplikasi.
- Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang dapat diubah ukurannya yang Anda gunakan untuk membangun dan meng-host sistem perangkat lunak Anda.
- Amazon Elastic File System (Amazon EFS) menyediakan sistem file yang sederhana, tanpa server, set-and-forget elastis untuk digunakan dengan Layanan AWS dan sumber daya lokal.
- Amazon Simple Storage Service (Amazon S3) adalah penyimpanan untuk internet.
- AWS Systems Manager Otomatisasi menyederhanakan tugas pemeliharaan dan penyebaran umum EC2 instans Amazon dan sumber daya lainnya. AWS

Opsi Backup 3: Snapshot volume Amazon EBS

- Emulator emulator Stromasys Charon-SSP menciptakan replika virtual dari perangkat keras SPARC asli di dalam sistem komputer standar 64-bit x86 yang kompatibel. Ini menjalankan kode biner SPARC asli, termasuk OSs seperti SunOS atau Solaris, produk berlapis mereka, dan aplikasi.
- AWS Backupadalah layanan perlindungan data yang dikelola sepenuhnya yang membuatnya mudah untuk memusatkan dan mengotomatisasi di seluruh Layanan AWS, di cloud, dan di tempat.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon. EC2
- Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang dapat diubah ukurannya yang Anda gunakan untuk membangun dan meng-host sistem perangkat lunak Anda.

Opsi Backup 4: AWS Storage GatewayVTL

- Emulator Stromasys Charon-SSP menciptakan replika virtual dari perangkat keras SPARC asli di dalam sistem komputer standar 64-bit x86 yang kompatibel. Ini menjalankan kode biner SPARC asli, termasuk OSs seperti SunOS atau Solaris, produk berlapis mereka, dan aplikasi.
- Bacula adalah sistem cadangan komputer tingkat perusahaan open-source. Untuk informasi selengkapnya tentang apakah aplikasi cadangan yang ada mendukung Tape Gateway, lihat Aplikasi pencadangan pihak ketiga yang didukung untuk Tape Gateway dalam AWS Storage Gateway dokumentasi.
- Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang dapat diubah ukurannya yang Anda gunakan untuk membangun dan meng-host sistem perangkat lunak Anda.
- Amazon Relational Database Service (Amazon RDS) untuk MySQL mendukung instans DB yang menjalankan beberapa versi MySQL.
- Amazon Simple Storage Service (Amazon S3) adalah penyimpanan untuk internet.
- AWS Storage Gatewaymenghubungkan perangkat lunak lokal dengan penyimpanan berbasis cloud untuk menyediakan integrasi tanpa batas dengan fitur keamanan data antara lingkungan TI lokal dan infrastruktur penyimpanan. AWS

Epik

Opsi Backup 1 - Buat cadangan pita virtual Stromasys

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file bersama Amazon EFS untuk penyimpanan file tape virtual.	Masuk ke Konsol Manajemen AWS atau gunakan AWS Command Line Interface (AWS CLI) untuk membuat sistem file Amazon EFS. Untuk informasi selengkap nya tentang ini, lihat Membuat sistem file Amazon EFS di dokumentasi Amazon EFS.	Arsitek awan
Konfigurasikan host Linux untuk me-mount sistem file bersama.	Instal driver Amazon EFS di instans Amazon EC2 Linux dan konfigurasikan OS Linux untuk memasang sistem file bersama Amazon EFS selama startup. Untuk informasi selengkap nya tentang hal ini, lihat Memasang sistem file menggunakan Amazon EFS mount helper dalam dokumentasi Amazon EFS.	DevOps insinyur
Instal emulator Charon-SSP.	Instal emulator Charon-SSP di instans Amazon EC2 Linux. Untuk informasi selengkapnya tentang ini, lihat Menyiapka n AWS Cloud instance untuk	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Charon-SSP di dokumentasi Stromasys.	
Buat wadah file tape virtual dalam sistem file bersama untuk setiap server tamu Sun SPARC.	Jalankan `touch <vtape-container-name>` perintah untuk membuat wadah file tape virtual di sistem file bersama untuk setiap server tamu Sun SPARC yang digunakan di emulator Charon-SSP.	DevOps insinyur
Konfigurasikan Charon-SSP Manager untuk membuat perangkat rekaman virtual untuk server tamu Sun SPARC.	Masuk ke Charon-SSP Manager, buat perangkat kaset virtual, dan konfigurasikan untuk menggunakan file wadah pita virtual untuk setiap server tamu Sun SPARC. Untuk informasi lebih lanjut tentang ini, lihat panduan pengguna Charon-SSP 5.2 untuk Linux di dokumentasi Stromasys.	DevOps insinyur
Validasi bahwa perangkat pita virtual tersedia di server tamu Sun SPARC.	Masuk ke setiap server tamu Sun SPARC dan jalankan `mt -f /dev/rmt/1` perintah untuk memvalidasi bahwa perangkat pita virtual dikonfigurasi di OS.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembangkan runbook dan otomatisasi Systems Manager Automation.	Kembangkan runbook Systems Manager Automation dan atur jendela pemeliharaan dan asosiasi di Systems Manager untuk menjadwalkan proses pencadangan. Untuk informasi selengkapnya tentang ini, lihat Panduan otomatisasi dan Menyiapkan jendela pemeliharaan di dokumentasi. AWS Systems Manager	Arsitek awan
Konfigurasikan Otomasi Systems Manager untuk mengarsipkan file kontainer pita virtual yang diputar.	Gunakan contoh kode dari opsi Kembali 1 di bagian Informasi tambahan untuk mengembangkan runbook Otomasi Manajer Sistem untuk mengarsipkan file kontainer pita virtual yang diputar ke Amazon S3.	Arsitek awan
Menerapkan runbook Systems Manager Automation untuk pengarsipan dan penjadwalan.	Terapkan runbook Systems Manager Automation dan jadwalkan untuk berjalan secara otomatis di Systems Manager. Untuk informasi selengkapnya tentang hal ini, lihat Panduan otomatisasi dalam dokumentasi Systems Manager.	Arsitek awan

Opsi Backup 2 - Buat snapshot Stromasys

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file bersama Amazon EFS untuk penyimpanan file tape virtual.	Masuk ke Konsol Manajemen AWS atau gunakan file AWS CLI untuk membuat sistem file Amazon EFS. Untuk informasi selengkapnya tentang hal ini, lihat Membuat sistem file Amazon EFS Anda di dokumentasi Amazon EFS.	Arsitek awan
Konfigurasikan host Linux untuk me-mount sistem file bersama.	Instal driver Amazon EFS di instans Amazon EC2 Linux dan konfigurasikan OS Linux untuk memasang sistem file bersama Amazon EFS selama startup. Untuk informasi selengkap nya tentang hal ini, lihat Memasang sistem file menggunakan Amazon EFS mount helper dalam dokumentasi Amazon EFS.	DevOps insinyur
Instal emulator Charon-SSP.	Instal emulator Charon-SSP di instans Amazon EC2 Linux. Untuk informasi selengkapnya tentang ini, lihat Menyiapka n AWS Cloud instance untuk Charon-SSP di dokumentasi Stromasys.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan server tamu Sun SPARC untuk memulai dengan opsi snapshot.	Gunakan Charon-SSP Manager untuk mengatur opsi snapshot untuk setiap server tamu Sun SPARC. Untuk informasi lebih lanjut tentang ini, lihat panduan pengguna Charon-SSP 5.2 untuk Linux di dokumentasi Stromasys.	DevOps insinyur
Kembangkan runbook Otomasi Systems Manager.	Gunakan contoh kode dari opsi Backup 2 di bagian Informasi tambahan untuk mengembangkan runbook Automation Systems Manager untuk menjalankan perintah snapshot dari jarak jauh di server tamu Sun SPARC selama jendela pemeliharaan.	Arsitek awan
Menerapkan runbook Systems Manager Automation dan atur asosiasi ke host Amazon EC2 Linux.	Terapkan runbook Systems Manager Automation dan siapkan jendela pemelihar aan dan asosiasi di Systems Manager untuk menjadwalkan proses pencadangan. Untuk informasi selengkap nya tentang ini, lihat Panduan otomatisasi dan Menyiapka n Windows Pemeliharaan di dokumentasi. AWS Systems Manager	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Arsipkan snapshot ke penyimpanan jangka panjang.	Gunakan kode contoh runbook dari bagian Informasi tambahan untuk mengembangkan runbook Automation Systems Manager untuk mengarsipkan file snapshot ke Amazon S3.	Arsitek awan

Opsi Backup 3 - Buat snapshot volume Amazon EBS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal emulator Charon-SSP.	Instal emulator Charon-SSP di instans Amazon EC2 Linux. Untuk informasi selengkapnya tentang ini, lihat Menyiapkan AWS Cloud instance untuk Charon-SSP di dokumentasi Stromasys.	DevOps insinyur
Buat volume Amazon EBS untuk server tamu Sun SPRAC.	Masuk ke Konsol Manajemen AWS, buka konsol Amazon EBS, lalu buat volume Amazon EBS untuk server tamu Sun SPRAC. Untuk informasi selengkapnya tentang ini, lihat Menyiapkan AWS Cloud instance untuk Charon-SSP di dokumentasi Stromasys.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan volume Amazon EBS ke instans Amazon EC2 Linux.	Di EC2 konsol Amazon, lampirkan volume Amazon EBS ke instans Amazon EC2 Linux. Untuk informasi selengkapnya tentang hal ini, lihat Melampirkan volume Amazon EBS ke instans di EC2 dokumentasi Amazon.	AWS DevOps
Petakan volume Amazon EBS sebagai drive SCSI di emulator Charon-SSP.	Konfigurasikan Charon-SSP Manager untuk memetakan volume Amazon EBS sebagai drive SCSI di server tamu Sun SPARC. Untuk informasi lebih lanjut tentang ini, lihat bagian konfigurasi penyimpanan SCSI dari panduan Charon-SSP V5.2 untuk Linux dalam dokumentasi Stromasys.	AWS DevOps
Konfigurasikan AWS Backup jadwal untuk snapshotting volume Amazon EBS.	Siapkan AWS Backup kebijakan dan jadwal untuk memotret volume Amazon EBS. Untuk informasi selengkapnya tentang ini, lihat cadangan dan pemulihan Amazon EBS menggunakan AWS Backup tutorial di dokumentasi Pusat AWS Pengembang.	AWS DevOps

Opsi Backup 4 - Buat AWS Storage Gateway VTL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat perangkat Tape Gateway.	Masuk ke Konsol Manajemen AWS, buka AWS Storage Gateway konsol, lalu buat perangkat Tape Gateway di VPC. Untuk informasi selengkap nya tentang ini, lihat Membuat gateway dalam AWS Storage Gateway dokumentasi.	Arsitek awan
Buat instans Amazon RDS DB untuk Katalog Bacula.	Buka konsol Amazon RDS dan buat instans Amazon RDS for MySQL DB. Untuk informasi selengkapnya tentang hal ini, lihat Membuat instans MySQL DB dan menghubungkan ke database pada instans MySQL DB dalam dokumentasi Amazon RDS.	Arsitek awan
Menyebarkan pengontrol aplikasi cadangan di VPC.	Instal Bacula di EC2 instans Amazon, gunakan pengontro l aplikasi cadangan, lalu konfigurasikan penyimpanan cadangan untuk terhubung dengan perangkat Tape Gateway. Anda dapat menggunakan sampel konfigurasi daemon penyimpanan Bacula Director dalam `Bacula-storage-`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`daemon-config.txt` file (terlampir). Untuk informasi lebih lanjut tentang ini, lihat dokumentasi Bacula.	
Siapkan aplikasi cadangan di server tamu Sun SPARC.	Siapkan klien kedua untuk menginstal dan mengatur aplikasi cadangan pada server tamu Sun SPARC dengan menggunakan konfigurasi Bacula sampel dalam `SUN-SPARC-Guest-Bacula-Config.txt` file (terlampir).	DevOps insinyur
Siapkan konfigurasi dan jadwal cadangan.	Siapkan konfigurasi dan jadwal cadangan di pengontrol aplikasi cadangan dengan menggunakan sampel konfigurasi Bacula Director dalam `Bacula-Directory-Config.txt` file (terlampir). Untuk informasi lebih lanjut tentang ini, lihat dokumentasi Bacula.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi bahwa konfigurasi dan jadwal cadangan sudah benar.	Ikuti instruksi dari dokumentasi Bacula untuk melakukan validasi dan pengujian cadangan untuk pengaturan Anda di server tamu Sun SPARC. Misalnya, Anda dapat menggunakan perintah berikut untuk memvalidasi file konfigurasi: • `bacula-dir -t -c bacula-dir.conf` • `bacula-fd -t -c bacula-fd.conf` • `bacula-sd -t -c bacula-sd.conf`	DevOps insinyur

Sumber daya terkait

- Charon virtual SPARC dengan lisensi VE
- Charon SPARC virtual
- Menggunakan layanan cloud dan penyimpanan objek dengan Bacula Enterprise Edition
- Tujuan pemulihan bencana (DR)
- Solusi emulasi sistem warisan Charon

Informasi tambahan

Opsi Backup 1 - Buat pita virtual Stromasys

Anda dapat menggunakan contoh kode runbook Systems Manager Automation berikut untuk memulai pencadangan secara otomatis dan kemudian menukar kasetnya:

```
...
# example backup script saved in SUN SPARC Server
 #!/usr/bin/bash
 mt -f  rewind
 tar -cvf
 mt -f  offline
...
         mainSteps:
         - action: aws:runShellScript
           name:
           inputs:
             onFailure: Abort
             timeoutSeconds: "1200"
             runCommand:
             - |
               # Validate tape backup container file exists
               if [ ! -f {{TapeBackupContainerFile}} ]; then
                 logger -s -p local3.warning "Tape backup container file is not exists
 - {{TapeBackupContainerFile}}, create a new one"
                 touch {{TapeBackupContainerFile}}
               fi
         - action: aws:runShellScript
           name: startBackup
           inputs:
             onFailure: Abort
             timeoutSeconds: "1200"
             runCommand:
             - |
               user={{BACKUP_USER}}
               keypair={{KEYPAIR_PATH}}
               server={{SUN_SPARC_IP}}
               backup_script={{BACKUP_SCRIPT}}
               ssh -i $keypair $user@$server -c "/usr/bin/bash $backup_script"
         - action: aws:runShellScript
           name: swapVirtualDiskContainer
           inputs:
             onFailure: Abort
             timeoutSeconds: "1200"
             runCommand:
             - |
               mv {{TapeBackupContainerFile}} {{TapeBackupContainerFile}}.$(date +%s)
               touch {{TapeBackupContainerFile}}
         - action: aws:runShellScript
```

```
          name: uploadBackupArchiveToS3
          inputs:
            onFailure: Abort
            timeoutSeconds: "1200"
            runCommand:
            - |
              aws s3 cp {{TapeBackupContainerFile}} s3://{{BACKUP_BUCKET}}/
{{SUN_SPARC_IP}}/$(date '+%Y-%m-%d')/
 ...
```

Opsi Backup 2 - Stromasys snapshot

Anda dapat menggunakan contoh kode runbook Systems Manager Automation berikut untuk mengotomatiskan proses pencadangan:

```
     ...

        mainSteps:
        - action: aws:runShellScript
          name: startSnapshot
          inputs:
            onFailure: Abort
            timeoutSeconds: "1200"
            runCommand:
            - |
              # You may consider some graceful stop of the application before taking a
 snapshot
              # Query SSP PID by configuration file
              # Example: ps ax | grep ssp-4 | grep Solaris10.cfg | awk '{print $1"
"$5}' | grep ssp4 | cut -f1 -d" "
              pid=`ps ax | grep ssp-4 | grep {{SSP_GUEST_CONFIG_FILE}} | awk '{print
$1" "$5}' | grep ssp4 | cut -f1 -d" "`
              if [ -n "${pid}" ]; then
                kill -SIGTSTP ${pid}
              else
                echo "No PID found for SPARC guest with config
 {{SSP_GUEST_CONFIG_FILE}}"
                exit 1
              fi
        - action: aws:runShellScript
          name: startBackup
          inputs:
            onFailure: Abort
```

```
              timeoutSeconds: "1200"
              runCommand:
              - |
                # upload snapshot and virtual disk files into S3
                aws s3 sync {{SNAPSHOT_FOLDER}} s3://{{BACKUP_BUCKET}}/$(date '+%Y-%m-
%d')/
                aws s3 cp {{VIRTUAL_DISK_FILE}} s3://{{BACKUP_BUCKET}}/$(date '+%Y-%m-
%d')/
          - action: aws:runShellScript
            name: restratSPARCGuest
            inputs:
              onFailure: Abort
              timeoutSeconds: "1200"
              runCommand:
              - |
                /opt/charon-ssp/ssp-4u/ssp4u -f {{SSP_GUEST_CONFIG_FILE}} -d -a
 {{SPARC_GUEST_NAME}} --snapshot {{SNAPSHOT_FOLDER}}
 ...
```

Opsi Backup 4 -AWS Storage GatewayVTL

Jika Anda menggunakan zona non-global Solaris untuk menjalankan server Sun SPARC warisan tervirtualisasi, pendekatan aplikasi cadangan dapat diterapkan ke zona non-global yang berjalan di server Sun SPARC (misalnya, klien cadangan dapat berjalan di dalam zona non-global). Namun, klien cadangan juga dapat berjalan di host Solaris dan mengambil snapshot dari zona non-global. Snapshot kemudian dapat dicadangkan pada kaset.

Konfigurasi sampel berikut menambahkan sistem file yang menghosting zona non-global Solaris ke dalam konfigurasi cadangan untuk host Solaris:

```
FileSet {
   Name = "Branded Zones"
   Include {
     Options {
       signature = MD5
     }
     File = /zones
   }
 }
```

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Cadangkan dan arsipkan data ke Amazon S3 dengan Veeam Backup & Replication

Jeanna James, Anthony Fiore (AWS), dan William Quigley, Amazon Web Services

Ringkasan

Pola ini merinci proses pengiriman cadangan yang dibuat oleh Veeam Backup & Replication ke kelas penyimpanan objek Amazon Simple Storage Service (Amazon S3) yang didukung dengan menggunakan kemampuan repositori cadangan skala Veeam.

Veeam mendukung beberapa kelas penyimpanan Amazon S3 agar paling sesuai dengan kebutuhan spesifik Anda. Anda dapat memilih jenis penyimpanan berdasarkan akses data, ketahanan, dan persyaratan biaya cadangan atau arsip data Anda. Misalnya, Anda dapat menyimpan data yang tidak Anda rencanakan untuk digunakan selama 30 hari atau lebih lama di Amazon S3 akses jarang (IA) dengan biaya lebih rendah. Jika Anda berencana untuk mengarsipkan data selama 90 hari atau lebih, Anda dapat menggunakan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive dengan tingkat arsip Veeam. Anda juga dapat menggunakan S3 Object Lock untuk membuat cadangan yang tidak dapat diubah dalam Amazon S3.

Pola ini tidak mencakup cara mengatur Veeam Backup & Replication dengan tape gateway di. AWS Storage Gateway Untuk informasi tentang topik tersebut, lihat Veeam Backup & Replication menggunakan AWS VTL Gateway - Panduan Penerapan di situs web Veeam.

> Peringatan: Skenario ini mengharuskan pengguna AWS Identity and Access Management (IAM) dengan akses terprogram dan kredensil jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan. Kunci akses dapat diperbarui jika perlu. Untuk informasi selengkap nya, lihat Memperbarui kunci akses di Panduan Pengguna IAM.

Prasyarat dan batasan

Prasyarat

- Veeam Backup & Replication, termasuk Veeam Availability Suite atau Veeam Backup Essentials, diinstal (Anda dapat mendaftar untuk uji coba gratis)

- Lisensi Cadangan & Replikasi Veeam dengan fungsionalitas Enterprise atau Enterprise Plus, yang mencakup Lisensi Universal Veeam (VUL)
- Pengguna IAM aktif dengan akses ke bucket Amazon S3
- Pengguna IAM aktif dengan akses ke Amazon Elastic Compute Cloud (Amazon EC2) dan Amazon Virtual Private Cloud (Amazon VPC), jika menggunakan tingkat arsip
- Konektivitas jaringan dari tempat ke Layanan AWS dengan bandwidth yang tersedia untuk cadangan dan memulihkan lalu lintas melalui koneksi internet publik atau antarmuka virtual AWS Direct Connect publik (VIF)
- Port jaringan dan titik akhir berikut dibuka untuk memastikan komunikasi yang tepat dengan repositori penyimpanan objek:
  - Penyimpanan Amazon S3 - TCP - port 443: Digunakan untuk berkomunikasi dengan penyimpanan Amazon S3.
  - Penyimpanan Amazon S3 — titik akhir cloud — `*.amazonaws.com` untuk Wilayah AWS dan AWS GovCloud (US) Regions, atau untuk Wilayah `*.amazonaws.com.cn` Tiongkok: Digunakan untuk berkomunikasi dengan penyimpanan Amazon S3. Untuk daftar lengkap titik akhir koneksi, lihat titik akhir Amazon S3 dalam dokumentasi. AWS
  - Penyimpanan Amazon S3 - TCP HTTP - port 80: Digunakan untuk memverifikasi status sertifikat. Pertimbangkan bahwa titik akhir verifikasi sertifikat — daftar pencabutan sertifikat (CRL) URLs dan server Protokol Status Sertifikat Online (OCSP) — dapat berubah sewaktu-waktu. Daftar alamat yang sebenarnya dapat ditemukan di sertifikat itu sendiri.
  - Penyimpanan Amazon S3 — titik akhir verifikasi sertifikat —`*.amazontrust.com`: Digunakan untuk memverifikasi status sertifikat. Pertimbangkan bahwa titik akhir verifikasi sertifikat (server CRL URLs dan OCSP) dapat berubah sewaktu-waktu. Daftar alamat yang sebenarnya dapat ditemukan di sertifikat itu sendiri.

Batasan

- Veeam tidak mendukung kebijakan Siklus Hidup S3 pada bucket S3 apa pun yang digunakan sebagai repositori penyimpanan objek Veeam. Ini termasuk kebijakan dengan transisi kelas penyimpanan Amazon S3 dan aturan kedaluwarsa Siklus Hidup S3. Veeam harus menjadi satu-satunya entitas yang mengelola objek-objek ini. Mengaktifkan kebijakan Siklus Hidup S3 mungkin memiliki hasil yang tidak terduga, termasuk kehilangan data.

Versi produk

- Veeam Backup & Replication v9.5 Pembaruan 4 atau lebih baru (hanya cadangan atau tingkat kapasitas)
- Veeam Backup & Replication v10 atau yang lebih baru (tingkat cadangan atau kapasitas dan Kunci Objek S3)
- Veeam Backup & Replication v11 atau yang lebih baru (tingkat cadangan atau kapasitas, tingkat arsip atau arsip, dan Kunci Objek S3)
- Veeam Backup & Replication v12 atau yang lebih baru (tingkat kinerja, tingkat cadangan atau kapasitas, tingkat arsip atau arsip, dan Kunci Objek S3)
- S3 Standard
- S3 Standard-IA
- S3 One Zone-IA
- Pengambilan Fleksibel Gletser S3 (hanya v11 dan yang lebih baru)
- S3 Glacier Deep Archive (hanya v11 dan yang lebih baru)
- Pengambilan Instan Gletser S3 (hanya v12 dan yang lebih baru)

Arsitektur

Tumpukan teknologi sumber

- Instalasi Cadangan & Replikasi Veeam lokal dengan konektivitas dari server cadangan Veeam atau server gateway Veeam ke Amazon S3

Tumpukan teknologi target

- Amazon S3
- Amazon VPC dan Amazon EC2 (jika menggunakan tingkat arsip)

Arsitektur target: SOBR

Diagram berikut menunjukkan arsitektur scale-out backup repository (SOBR).

Perangkat lunak Cadangan dan Replikasi Veeam melindungi data dari kesalahan logis seperti kegagalan sistem, kesalahan aplikasi, atau penghapusan yang tidak disengaja. Dalam diagram

ini, backup dijalankan di tempat pertama, dan salinan sekunder dikirim langsung ke Amazon S3. Cadangan mewakili point-in-time salinan data.

Alur kerja terdiri dari tiga komponen utama yang diperlukan untuk tiering atau menyalin cadangan ke Amazon S3, dan satu komponen opsional:

- Veeam Backup & Replication (1) — Server cadangan yang bertanggung jawab untuk mengkoordinasikan, mengendalikan, dan mengelola infrastruktur cadangan, pengaturan, pekerjaan, tugas pemulihan, dan proses lainnya.
- Server gateway Veeam (tidak ditampilkan dalam diagram) — Server gateway lokal opsional yang diperlukan jika server cadangan Veeam tidak memiliki konektivitas keluar ke Amazon S3.
- Repositori cadangan scale-out (2) - Sistem repositori dengan dukungan penskalaan horizontal untuk penyimpanan data multi-tier. Repositori cadangan scale-out terdiri dari satu atau lebih repositori cadangan yang menyediakan akses cepat ke data dan dapat diperluas dengan repositori penyimpanan objek Amazon S3 untuk penyimpanan jangka panjang (tingkat kapasitas) dan pengarsipan (tingkat arsip). Veeam menggunakan repositori cadangan scale-out untuk mengatur data secara otomatis antara penyimpanan objek lokal (tingkat kinerja) dan Amazon S3 (kapasitas dan tingkatan arsip).

  > **Note**
  >
  > Dimulai dengan Veeam Backup & Replication v12.2, fitur Direct to S3 Glacier menjadikan tingkat kapasitas S3 opsional. SOBR dapat dikonfigurasi dengan tingkat kinerja dan tingkat arsip S3 Glacier. Konfigurasi ini berguna bagi pengguna yang memiliki investasi signifikan dalam penyimpanan lokal (lokal) untuk tingkat kapasitas dan yang hanya memerlukan retensi arsip jangka panjang di cloud. Untuk informasi selengkapnya, lihat dokumentasi Backup & Replikasi Veeam.

- Amazon S3 (3) — layanan penyimpanan AWS objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja.

Arsitektur target: DTO

Diagram berikut menunjukkan arsitektur direct-to-object (DTO).

Dalam diagram ini, data cadangan langsung masuk ke Amazon S3 tanpa disimpan di tempat terlebih dahulu. Salinan sekunder dapat disimpan di S3 Glacier.

Otomatisasi dan skala

Anda dapat mengotomatiskan pembuatan sumber daya IAM dan bucket S3 dengan menggunakan CloudFormation templat yang disediakan di repositori. VeeamHub GitHub Template mencakup opsi standar dan tidak dapat diubah.

Alat

Alat dan Layanan AWS

- Veeam Backup & Replication adalah solusi dari Veeam untuk melindungi, mencadangkan, mereplikasi, dan memulihkan beban kerja virtual dan fisik Anda.
- CloudFormationmembantu Anda memodelkan dan menyiapkan AWS sumber daya Anda, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Anda dapat mengelola dan menyediakan tumpukan di beberapa Akun AWS dan Wilayah AWS.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, dan Anda dapat meningkatkan skala atau menskalakan.
- AWS Identity and Access Management (IAM) adalah layanan web untuk mengontrol akses dengan aman. Layanan AWS Dengan IAM, Anda dapat mengelola pengguna secara terpusat, kredensi keamanan seperti kunci akses, dan izin yang mengontrol AWS sumber daya mana yang dapat diakses pengguna dan aplikasi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web.
- Amazon S3 Glacier (S3 Glacier) adalah layanan yang aman dan tahan lama untuk pengarsipan data berbiaya rendah dan pencadangan jangka panjang.
- Amazon Virtual Private Cloud (Amazon VPC) menyediakan bagian yang terisolasi secara logis di AWS Cloud mana Anda dapat meluncurkan AWS sumber daya di jaringan virtual yang telah Anda tentukan. Jaringan virtual ini sangat mirip dengan jaringan konvensional yang akan Anda operasikan di pusat data Anda sendiri dengan manfaatnya, yaitu menggunakan infrastruktur AWS yang dapat diskalakan.

Kode

Gunakan CloudFormation template yang disediakan di VeeamHub GitHub repositori untuk secara otomatis membuat sumber daya IAM dan bucket S3 untuk pola ini. Jika Anda lebih suka membuat sumber daya ini secara manual, ikuti langkah-langkah di bagian Epik.

Praktik terbaik

Sesuai dengan praktik terbaik IAM, kami sangat menyarankan agar Anda secara teratur memutar kredensil pengguna IAM jangka panjang, seperti pengguna IAM yang Anda gunakan untuk menulis cadangan Cadangan & Replikasi Veeam ke Amazon S3. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan dalam dokumentasi IAM.

Epik

Konfigurasikan penyimpanan Amazon S3 di akun Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna IAM.	Ikuti petunjuk dalam dokumentasi IAM untuk membuat pengguna IAM. Pengguna ini seharusnya tidak memiliki akses AWS konsol, dan Anda perlu membuat kunci akses untuk pengguna ini. Veeam menggunakan entitas ini untuk mengauten tikasi dengan membaca dan menulis AWS ke bucket S3 Anda. Anda harus memberika n hak istimewa paling sedikit (yaitu, hanya memberika n izin yang diperlukan untuk melakukan tugas) sehingga pengguna tidak memiliki otoritas lebih dari yang dibutuhkan. Misalnya	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kebijakan IAM untuk dilampirkan ke pengguna Veeam IAM Anda, lihat bagian Informasi tambahan. **Note** Atau, Anda dapat menggunakan CloudFormation template yang disediakan di VeeamHub GitHub repositori untuk membuat pengguna IAM dan bucket S3 untuk pola ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3. 2. Jika Anda belum memiliki bucket S3 yang sudah ada untuk digunakan sebagai penyimpanan target, pilih Buat bucket, dan tentukan nama bucket Wilayah AWS, serta pengaturan bucket. • Kami menyarankan Anda mengaktifkan opsi Blokir Akses Publik untuk bucket S3 dan menyiapkan kebijakan akses dan izin pengguna untuk memenuhi persyaratan organisasi Anda. Sebagai contoh, lihat Amazon S3Documentation. • Kami menyarankan Anda mengaktifkan S3 Object Lock, bahkan jika Anda tidak berniat untuk menggunakannya segera. Pengaturan ini hanya dapat diaktifkan pada saat pembuatan bucket S3. Untuk informasi selengkapnya, lihat Membuat bucket di dokumentasi Amazon S3.	Administrator AWS

Tambahkan Pengambilan Fleksibel Amazon S3 dan S3 Glacier (atau S3 Glacier Deep Archive) ke Cadangan & Replikasi Veeam

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan wizard Repositori Objek Baru.	Sebelum Anda mengatur penyimpanan objek dan repositori cadangan skala di Veeam, Anda harus menambahkan repositori penyimpanan Amazon S3 dan S3 Glacier yang ingin Anda gunakan untuk kapasitas dan tingkatan arsip. Dalam epik berikutnya, Anda akan menghubungkan repositori penyimpanan ini ke repositori cadangan scale-out Anda. 1. Pada konsol Veeam, buka tampilan Backup Infrastru cture. 2. Di panel inventaris, pilih node Backup Repositories, lalu pilih Add Repository. 3. Di kotak dialog Add Backup Repository, pilih Object Storage, Amazon S3.	Administrator AWS, Pemilik aplikasi
Tambahkan penyimpanan Amazon S3 untuk tingkat kapasitas.	1. Di kotak dialog Amazon Cloud Storage Services, pilih Amazon S3. 2. Pada langkah Nama wizard, tentukan nama penyimpan an objek dan deskripsi singkat, seperti pembuat dan tanggal pembuatan.	Administrator AWS, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pada langkah Akun wizard, tentukan akun penyimpanan objek. • Untuk Kredensial, pilih pengguna IAM yang Anda buat di epik pertama untuk mengakses penyimpanan objek Amazon S3 Anda. • Untuk wilayah AWS, pilih lokasi bucket S3. Wilayah AWS 4. Pada langkah Bucket wizard, tentukan pengaturan penyimpanan objek. • Untuk wilayah pusat data, pilih Wilayah AWS tempat bucket S3 berada. • Untuk Bucket, pilih bucket S3 yang Anda buat di epik pertama. • Untuk Folder, buat atau pilih folder cloud untuk memetakan repositori penyimpanan objek Anda. • Jika Anda ingin mengaktifkan kekekalan, pilih Buat cadangan terbaru tidak dapat diubah selama X hari dan atur periode waktu di mana cadangan Anda harus	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dikunci. Perhatikan bahwa mengaktifkan kekekalan menghasilkan peningkatan biaya karena meningkatnya jumlah panggilan API ke Amazon S3 dari Veeam. 5. Pada langkah Ringkasan wizard, tinjau informasi konfigurasi, lalu pilih Selesai.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan penyimpanan S3 Glacier untuk tingkat arsip.	Jika Anda ingin membuat tingkat arsip, gunakan izin IAM yang dirinci di bagian Informasi tambahan. 1. Luncurkan wizard Repositori Objek Baru seperti yang dijelaskan sebelumnya. 2. Di kotak dialog Amazon Cloud Storage Services, pilih Amazon S3 Glacier. 3. Pada langkah Nama wizard, tentukan nama penyimpanan objek dan deskripsi singkat, seperti pembuat dan tanggal pembuatan. 4. Pada langkah Akun wizard, tentukan akun penyimpanan objek. • Untuk Kredensial, pilih pengguna IAM yang Anda buat di epik pertama untuk mengakses penyimpanan objek S3 Glacier Anda. • Untuk wilayah AWS, pilih lokasi bucket S3. Wilayah AWS 5. Pada langkah Bucket wizard, tentukan pengaturan penyimpanan objek. • Untuk wilayah Pusat data, pilih Wilayah AWS.	Administrator AWS, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk Bucket, pilih bucket S3 untuk menyimpan data cadangan Anda. Ini bisa menjadi ember yang sama yang Anda gunakan untuk tingkat kapasitas. • Untuk Folder, buat atau pilih folder cloud untuk memetakan repositor i penyimpanan objek Anda. • Jika Anda ingin mengaktif kan kekekalan, pilih Jadikan cadangan terbaru tidak dapat diubah selama seluruh durasi kebijakan retensi mereka. Perhatikan bahwa mengaktifkan kekekalan menghasilkan peningkatan biaya karena meningkatnya jumlah panggilan API ke Amazon S3 dari Veeam. • Jika Anda ingin menggunakan S3 Glacier Deep Archive sebagai kelas penyimpanan arsip Anda, pilih Use the Deep Archive Storage Class. 6. Pada langkah Proxy Appliance dari wizard, konfigurasikan instance	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tambahan yang digunakan untuk mentransfer data dari Amazon S3 ke S3 Glacier. Anda dapat menggunak an pengaturan default atau mengkonfigurasi setiap pengaturan secara manual. Untuk mengkonfigurasi pengaturan secara manual: • Pilih Sesuaikan. • EC2 Misalnya jenis, pilih jenis instans untuk alat proxy, berdasark an kecepatan dan persyaratan biaya Anda untuk mentransfer file cadangan ke tingkat arsip repositori cadangan scale-out Anda. • Untuk Amazon VPC, pilih VPC untuk instance target. • Untuk Subnet, pilih subnet untuk alat proxy. • Untuk grup Keamanan, pilih grup keamanan untuk diasosiasikan dengan alat proxy. • Untuk port Redirector, tentukan port TCP untuk permintaan perutean antara alat proxy dan komponen infrastruktur cadangan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Pilih OK untuk mengonfir masi pengaturan Anda. 7. Pada langkah Ringkasan wizard, tinjau informasi konfigurasi, lalu pilih Selesai.	

Tambahkan repositori cadangan skala

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan wizard Repositori Cadangan Scale-Out Baru.	1. Pada konsol Veeam, buka tampilan Backup Infrastru cture. 2. Di panel inventaris, pilih Scale-out Repositories, lalu pilih Add Scale-out Repository.	Pemilik aplikasi, administrator sistem AWS
Tambahkan repositori cadangan scale-out dan konfigurasikan kapasitas dan tingkatan arsip.	1. Pada langkah Nama wizard, tentukan nama dan deskripsi singkat tentang repositori cadangan scale-out. 2. Jika perlu, tambahkan perluasan kinerja. Anda juga dapat menggunak an repositori cadangan lokal Veeam yang ada sebagai tingkat kinerja Anda. Dimulai dengan Veeam versi 12, Anda dapat menambahkan bucket S3 sebagai tingkat	Pemilik aplikasi, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kinerja untuk cadangan direct-to-object (DTO), melewati tingkat kinerja lokal. 3. Pilih Advanced, dan tentukan opsi tambahan untuk repositori cadangan scale-out. • Pilih Gunakan file cadangan per mesin untuk membuat file cadangan terpisah untuk setiap mesin dan tulis file-file ini ke repositori cadangan di beberapa aliran secara bersamaan . Opsi ini direkomen dasikan untuk penyimpan an yang lebih baik dan pemanfaatan sumber daya komputasi. • Pilih Lakukan pencadang an penuh saat batas yang diperlukan offline untuk membuat file cadangan lengkap jika tingkat yang berisi titik pemulihan untuk cadangan tambahan offline. Opsi ini membutuhkan ruang kosong di repositori cadangan scale-out untuk meng-host file cadangan lengkap.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pada langkah Kebijakan wizard, tentukan kebijakan penempatan cadangan untuk repositori. • Pilih Lokalitas data untuk menyimpan file cadangan penuh dan tambahan yang termasuk dalam rantai yang sama bersama-sama, dengan tingkat kinerja yang sama. Anda dapat menyimpan file yang termasuk dalam rantai cadangan baru dengan tingkat kinerja yang sama atau ke yang lain (kecuali jika Anda menggunakan alat penyimpanan deduplikasi sebagai tingkat kinerja). • Pilih Kinerja untuk menyimpan file cadangan penuh dan tambahan ke tingkat kinerja yang berbeda. Opsi ini membutuhkan koneksi jaringan yang cepat dan andal. Jika Anda memilih Kinerja, Anda dapat membatasi jenis file cadangan untuk disimpan pada setiap tingkat kinerja. Misalnya,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda dapat menyimpan file cadangan penuh pada satu tingkat dan file cadangan tambahan pada ekstensi lain. Untuk memilih jenis file: • Pilih Sesuaikan. • Di kotak dialog Pengaturan Penempatan Cadangan, pilih tingkat kinerja, lalu pilih Edit. • Pilih jenis file cadangan yang ingin Anda simpan sejauh itu. 5. Pada langkah Tingkat Kapasitas wizard, konfigura sikan tingkat penyimpanan jangka panjang yang ingin Anda lampirkan ke repositor i cadangan scale-out. • Pilih Perluas kapasitas repositori cadangan scale-out dengan penyimpanan objek. Untuk repositori penyimpanan objek, pilih penyimpanan Amazon S3 untuk tingkat kapasitas yang Anda tambahkan di epik sebelumnya. • Pilih Jendela untuk memilih jendela waktu	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk memindahkan atau menyalin data. • Pilih Salin cadangan ke penyimpanan objek segera setelah dibuat untuk menyalin semua atau hanya file cadangan yang baru dibuat sejauh kapasitas. • Pilih Pindahkan cadangan ke penyimpanan objek saat mereka menua dari jendela pemulihan operasional untuk mentransfer rantai cadangan yang tidak aktif ke tingkat kapasitas. Di bidang Pindahkan file cadangan yang lebih lama dari X hari, tentukan durasi setelah file cadangan harus diturunkan. (Untuk membongkar rantai cadangan yang tidak aktif pada hari pembuatannya, tentukan 0 hari.) Anda juga dapat memilih Override untuk memindahkan file cadangan lebih cepat jika repositori cadangan scale-out telah mencapai	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ambang batas yang Anda tentukan. • Pilih Enkripsi data yang diunggah ke penyimpanan objek dan tentukan kata sandi untuk mengenkripsi semua data dan metadatanya untuk dibongkar. Pilih Tambah atau Kelola kata sandi untuk menentukan kata sandi baru. 6. Pada langkah Tingkat Arsip wizard, konfigurasikan tingkat penyimpanan arsip yang ingin Anda lampirkan ke repositori cadangan scale-out. (Langkah ini tidak muncul jika Anda melewatkan menambahkan penyimpanan Amazon S3 Glacier.) • Pilih Arsipkan cadangan penuh GFS ke penyimpanan objek. Untuk repositori penyimpanan objek, pilih penyimpanan Amazon Glacier yang Anda tambahkan di epik sebelumnya. • Untuk cadangan Arsip GFS yang lebih lama dari N hari, pilih jendela waktu	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk memindahkan file ke tingkat arsip. (Untuk mengarsipkan rantai cadangan yang tidak aktif pada hari pembuatannya, tentukan 0 hari.) 7. Pada langkah Ringkasan wizard, tinjau konfigurasi repositori cadangan scale-out, lalu pilih Selesai.	

Sumber daya terkait

- Membuat pengguna IAM di Akun AWS(dokumentasi IAM) Anda
- Membuat ember (dokumentasi Amazon S3)
- Memblokir akses publik ke penyimpanan Amazon S3 Anda (dokumentasi Amazon S3)
- Menggunakan Kunci Objek S3 (dokumentasi Amazon S3)
- Dokumentasi teknis Veeam
- Cara Membuat Kebijakan IAM Aman untuk Koneksi ke Penyimpanan Objek S3 (dokumentasi Veeam)

Informasi tambahan

Bagian berikut menyediakan contoh kebijakan IAM yang dapat Anda gunakan saat membuat pengguna IAM di bagian Epics dari pola ini.

Kebijakan IAM untuk tingkat kapasitas

> **Note**
>
> Ubah nama bucket S3 dalam kebijakan contoh dari `<yourbucketname>` ke nama bucket S3 yang ingin Anda gunakan untuk cadangan tingkat kapasitas Veeam. Perhatikan juga bahwa kebijakan harus dibatasi pada sumber daya spesifik yang digunakan untuk Veeam (ditunjukkan oleh `Resource` spesifikasi dalam kebijakan berikut), dan bahwa bagian pertama

kebijakan menonaktifkan enkripsi sisi klien, seperti yang dibahas dalam posting AWS blog Mencegah enkripsi objek Amazon S3 yang tidak diinginkan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RestrictSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<your-bucket-name>/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "false"
                }
            }
        },
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion",
                "s3:ListBucketVersions",
                "s3:ListBucket",
                "s3:PutObjectLegalHold",
                "s3:GetBucketVersioning",
                "s3:GetObjectLegalHold",
                "s3:GetBucketObjectLockConfiguration",
                "s3:PutObject*",
                "s3:GetObject*",
                "s3:GetEncryptionConfiguration",
                "s3:PutObjectRetention",
                "s3:PutBucketObjectLockConfiguration",
                "s3:DeleteObject*",
                "s3:DeleteObjectVersion",
                "s3:GetBucketLocation"

            ],
            "Resource": [
                "arn:aws:s3:::<yourbucketname>",
                "arn:aws:s3:::<yourbucketname>/*"
            ]
```

```
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

Kebijakan IAM untuk tingkat arsip

> **Note**
>
> Ubah nama bucket S3 dalam kebijakan contoh dari <yourbucketname> ke nama bucket S3 yang ingin Anda gunakan untuk cadangan tingkat arsip Veeam.

Untuk menggunakan VPC, subnet, dan grup keamanan yang ada:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3Permissions",
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
                "s3:PutObject",
                "s3:GetObject",
                "s3:RestoreObject",
                "s3:ListBucket",
                "s3:AbortMultipartUpload",
                "s3:GetBucketVersioning",
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation",
                "s3:GetBucketObjectLockConfiguration",
                "s3:PutObjectRetention",
                "s3:GetObjectVersion",
```

```
                "s3:PutObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:DeleteObjectVersion",
                "s3:ListBucketVersions"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket-name>",
                "arn:aws:s3:::<bucket-name>/*"
            ]
        }
    ]
}

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EC2Permissions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:CreateKeyPair",
                "ec2:DescribeKeyPairs",
                "ec2:RunInstances",
                "ec2:DeleteKeyPair",
                "ec2:DescribeVpcAttribute",
                "ec2:CreateTags",
                "ec2:DescribeSubnets",
                "ec2:TerminateInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeImages",
                "ec2:DescribeVpcs"
            ],
            "Resource": "arn:aws:ec2:<region>:<account-id>:*"
        }
    ]
}
```

Untuk membuat VPC, subnet, dan grup keamanan baru:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "S3Permissions",
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
                "s3:PutObject",
                "s3:GetObject",
                "s3:RestoreObject",
                "s3:ListBucket",
                "s3:AbortMultipartUpload",
                "s3:GetBucketVersioning",
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation",
                "s3:GetBucketObjectLockConfiguration",
                "s3:PutObjectRetention",
                "s3:GetObjectVersion",
                "s3:PutObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:DeleteObjectVersion",
                "s3:ListBucketVersions"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket-name>",
                "arn:aws:s3:::<bucket-name>/*"
            ]
        }
    ]
}

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EC2Permissions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:CreateKeyPair",
                "ec2:DescribeKeyPairs",
                "ec2:RunInstances",
                "ec2:DeleteKeyPair",
                "ec2:DescribeVpcAttribute",
                "ec2:CreateTags",
                "ec2:DescribeSubnets",
```

```
                "ec2:TerminateInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeImages",
                "ec2:DescribeVpcs",
                "ec2:CreateVpc",
                "ec2:CreateSubnet",
                "ec2:DescribeAvailabilityZones",
                "ec2:CreateRoute",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:ModifyVpcAttribute",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteSecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:DescribeRouteTables",
                "ec2:DescribeInstanceTypes"
            ],
            "Resource": "*"
        }
    ]
}
```

Aktifkan pengarsipan DB2 log langsung ke Amazon S3 dalam database IBM Db2

Ambarish Satarkar, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan Amazon Simple Storage Service (Amazon S3) sebagai penyimpanan katalog untuk log arsip yang dihasilkan oleh IBM Db2, tanpa menggunakan area pementasan.

Anda dapat menentukan penyimpanan Amazon S3 DB2REMOTE untuk parameter konfigurasi metode arsip log logarchmeth1 dan logarchmeth2. Anda dapat menggunakan `logarchmeth1` parameter untuk menentukan tujuan utama untuk log yang diarsipkan dari jalur log saat ini. Dengan kemampuan ini, Anda dapat mengarsipkan dan mengambil log transaksi ke dan dari Amazon S3 secara langsung, tanpa menggunakan area pementasan.

Amazon S3 menyimpan data yang diunggah ke dalamnya setidaknya di tiga perangkat dalam satu. Wilayah AWS Jutaan pelanggan dari semua ukuran dan industri menggunakan Amazon S3 untuk menyimpan cadangan perusahaan mengingat ketersediaannya yang tinggi, opsi penyimpanan yang fleksibel, kebijakan siklus hidup, dan keamanan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Database IBM Db2 berjalan pada instans Amazon Elastic Compute Cloud (Amazon). EC2
- AWS Command Line Interface (AWS CLI) diinstal
- libcurl dan libxml2 diinstal pada instance Db2. EC2

Batasan

- Hanya Db2 11.5.7 atau yang lebih baru yang memungkinkan pengarsipan log langsung ke penyimpanan Amazon S3.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

- Di semua konfigurasi, batasan berikut ada untuk Amazon S3:
  - AWS Key Management Service (AWS KMS) tidak didukung.
  - AWS kredensyal berbasis peran AWS Identity and Access Management ((IAM)) atau berbasis token (AWS Security Token Service (AWS STS)) tidak didukung.

Versi produk

- AWS CLI versi 2 atau yang lebih baru
- IBM Db2 11.5.7 atau yang lebih baru
- Linux SUSE Linux Enterprise Server (SLES) 11 atau yang lebih baru
- Red Hat Enterprise Linux (RHEL) 6 atau lebih baru
- Windows Server 2008 R2, 2012 (R2), 2016, atau 2019

Arsitektur

Diagram berikut menunjukkan komponen dan alur kerja untuk pola ini.

Arsitektur pada AWS Cloud meliputi yang berikut:

- Virtual Private Cloud (VPC) — Bagian yang terisolasi secara logis dari AWS Cloud tempat Anda meluncurkan sumber daya.
- Availability Zone - Menyediakan ketersediaan tinggi dengan menjalankan beban kerja Db2 LUW (Linux, Unix, Windows) di pusat data yang terisolasi di dalam. Wilayah AWS
- Public subnet — Menyediakan akses RDP (Remote Desktop Protocol) untuk administrator dan konektivitas internet melalui gateway NAT.
- Subnet pribadi - Menghosting database Db2 LUW. Instance Db2 LUW dikonfigurasi dengan parameter. `LOGARCHMETH1` Parameter menulis file arsip log database langsung ke jalur Amazon S3 melalui titik akhir gateway.

Berikut ini Layanan AWS memberikan dukungan:

- Amazon S3 - Berfungsi sebagai lokasi penyimpanan yang tahan lama dan dapat diskalakan untuk file arsip log Db2.

- Amazon Elastic File System (Amazon EFS) - Menyediakan sistem file bersama yang dikelola sepenuhnya yang dapat digunakan Db2 untuk pencadangan dan pementasan basis data. Db2 juga dapat menggunakan Amazon EFS sebagai titik pemasangan untuk file log sebelum diarsipkan ke Amazon S3.
- Amazon CloudWatch — Mengumpulkan dan memantau metrik, log, dan peristiwa dari Db2 dan instans yang mendasarinya. EC2 Anda dapat menggunakannya CloudWatch untuk membuat alarm, dasbor, dan respons otomatis terhadap masalah kinerja atau ketersediaan.

Otomatisasi dan skala

- Pola ini memberikan solusi yang sepenuhnya otomatis untuk menyimpan cadangan arsip log Db2.
- Anda dapat menggunakan bucket Amazon S3 yang sama untuk mengaktifkan arsip log dari beberapa database Db2.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di file AWS Cloud.
- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua aplikasi Anda Akun AWS dan cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat-alat lainnya

- libcurl adalah pustaka transfer URL sisi klien gratis.
- libxml2 adalah parser dan toolkit XMLC gratis.

Praktik terbaik

- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Konfigurasikan Layanan AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mengatur AWS CLI.	Untuk mengunduh dan menginstal AWS CLI, gunakan perintah berikut:
	`i) curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"`
	`ii) unzip awscliv2.zip`
	`iii) sudo ./aws/install`
		Administrator sistem AWS, administrator AWS

Konfigurasikan AWS CLI.	Untuk mengkonfigurasi AWS CLI, gunakan perintah berikut:
	`$ aws configure`
	`AWS Access Key ID [None]:**********`
	`**********`
	`AWS Secret Access Key [None]: *********`
	`**********`
		Administrator sistem AWS, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Default region name [None]: < aws region > Default output format [None]: text`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat pengguna IAM.	Untuk membuat pengguna IAM untuk digunakan nanti untuk koneksi database Db2 dengan Amazon S3, gunakan perintah berikut:
	`aws iam create-user --user-name <unique username>`
	Berikut ini adalah contoh dari perintah:
	`aws iam create-user --user-name db_backup_user`
	Warning
	Skenario ini mengharuskan pengguna IAM dengan akses terprogram dan kredensyal jangka panjang, yang menghadirkan risiko keamanan. Untuk mengurangi risiko ini, kami menyarankan agar Anda memberi pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lagi diperlukan. Kunci akses dapat diperbaru i jika perlu. Untuk informasi selengkap nya, lihat kredensyal AWS keamanan dan Kelola kunci akses untuk pengguna IAM dalam dokumentasi IAM.	

Buat ember Amazon S3.	Untuk membuat bucket Amazon S3 untuk menyimpan cadangan database, gunakan perintah berikut:
	`aws s3api create-bucket --bucket <unique bucket name> --region <aws region>`
	Berikut ini adalah contoh perintah:
	`aws s3api create-bucket --bucket myfirstbucket --region af-south-1`
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Otorisasi pengguna IAM.	Untuk mengizinkan pengguna IAM yang baru dibuat agar memiliki izin Amazon S3, gunakan langkah-langkah berikut: 1. Untuk membuat file kebijakan IAM baru `db2bucket.json` untuk bucket Amazon S3, gunakan perintah berikut: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "ListObjectsInBucket", "Effect": "Allow", "Action": "s3:ListBucket", "Resource": [ "arn:aws:s3:::myfirstbucket" ] }, { "Sid": "AllObjectActions", "Effect": "Allow", "Action": [` 	Administrator sistem AWS, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"s3:PutObject",
	"s3:GetObject",
	"s3:DeleteObject"
	],
	"Resource": [
	"arn:aws:s3:::myfirstbucket/*"
	]
	}
	]
	}</pre>
	2. Buat kebijakan IAM berdasarkan `db2bucket.json` file. Output JSON menunjukkan Nama Sumber Daya Amazon (ARN) untuk kebijakan :. `"Arn": "arn:aws:iam::111122223333:policy/db2s3policy"`
	<pre>aws iam create-policy \
	--policy-name db2s3policy \
	--policy-document file://db2bucket.json</pre>
	3. Lampirkan kebijakan IAM ke pengguna IAM dengan kebijakan ARN.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws iam attach-user-policy --user-name db_backup_user --policy-arn "arn:aws:iam::111122223333:policy/db2s3policy "`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat kunci akses.	Untuk menghasilkan kunci akses untuk mengakses Amazon S3 secara terprogram dari DB2 instance, gunakan perintah berikut:
	
	`aws iam create-access-key --user-name <username>`
	
	Berikut ini adalah contoh dari perintah:
	
	`aws iam create-access-key --user-name db_backup_user`
	
	Warning
	Skenario ini mengharuskan pengguna IAM dengan akses terprogram dan kredensyal jangka panjang, yang menghadirkan risiko keamanan. Untuk mengurangi risiko ini, kami menyarankan agar Anda memberi pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lagi diperlukan. Kunci akses dapat diperbaru i jika perlu. Untuk informasi selengkap nya, lihat kredensyal AWS keamanan dan Kelola kunci akses untuk pengguna IAM dalam dokumentasi IAM.	
Buat keystore PKCS.	Untuk membuat keystore PKCS untuk menyimpan kunci dan membuat kunci akses rahasia untuk mentransfer data ke Amazon S3, gunakan perintah berikut: `gsk8capicmd_64 -keydb -create -db "/db2/db2<sid>/.keystore/db6-s3.p12" -pw "<password>" -type pkcs12 -stash`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
DB2 Konfigurasikan untuk menggunakan keystore.	Untuk mengkonfigurasi DB2 untuk menggunakan keystore dengan `keystore_type` parameter `keystore_location` dan, gunakan perintah berikut: `db2 "update dbm cfg using keystore_location /db2/db2<sid>/.keystore/db6-s3.p12 keystore_type pkcs12"`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat alias akses DB2 penyimpanan.	Alias akses penyimpanan menentukan bucket Amazon S3 yang akan digunakan . Ini juga menyediakan detail koneksi seperti nama pengguna dan kata sandi yang disimpan di keystore lokal dalam format terenkripsi. Untuk informasi selengkapnya, lihat perintah CATALOG STORAGE ACCESS dalam dokumentasi IBM Db2.
	
	Untuk membuat alias akses penyimpanan, gunakan sintaks berikut:
	
	`db2 "catalog storage access alias <alias_name> vendor S3 server <S3 endpoint> user '<access_key>' password '<secret_access_key>' container '<bucket_name>'"`
	
	Berikut ini adalah contoh:
	
	`db2 "catalog storage access alias DB2BKPS3 vendor S3 server s3.us-west-2.amazonaws.com user '********** ********** ' password '********** ********** ' container 'myfirstbucket'"`
		Administrator sistem AWS

Perbarui lokasi logarchmeth1 dan mulai ulang DB2 DB2

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui `LOGARCHMETH1` lokasi.	Untuk menggunakan alias akses penyimpanan yang Anda tentukan sebelumnya, perbarui parameter `LOGARCHMETH1` database, gunakan perintah berikut:
	`db2 update db cfg for <DBNAME> using LOGARCHMETH1 'DB2REMOTE://<storage_alias_name>//<sub folder>'`
	Untuk memisahkan log dari file lain, tentukan subdirektori (yaitu, awalan bucket Amazon S3`TESTDB_LOGS`) untuk menyimpan log dalam bucket S3.
	Berikut ini adalah contoh:
	`db2 update db cfg for ABC using LOGARCHMETH1 'DB2REMOTE://DB2BKPS3//TESTDB_LOGS/'`
	Anda akan melihat pesan berikut: `DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.`
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai ulang DB2.	Mulai ulang DB2 instance setelah mengkonfigurasi ulang untuk pengarsipan log. Namun, jika `LOGARCHMETH1` sebelumnya disetel ke lokasi sistem file apa pun, maka restart tidak diperlukan.	Administrator AWS, administrator sistem AWS

Periksa jalur log arsip di Amazon S3 dan db2diag.log

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa log arsip di Amazon S3.	Pada titik ini, database Anda sepenuhnya dikonfigurasi untuk mengarsipkan log transaksi langsung ke penyimpanan Amazon S3. Untuk mengonfirmasi konfigurasi, mulailah menjalankan aktivitas transaksional pada database untuk mulai mengkonsumsi (dan mengarsipkan) ruang log. Kemudian, periksa log arsip di Amazon S3.	Administrator AWS, administrator sistem AWS
Periksa konfigurasi log arsip di`db2diag.log` .	Setelah Anda memeriksa log arsip di Amazon S3, cari pesan berikut di log DB2 diagnostik: `db2diag.log` `MESSAGE : ADM1846I Completed archive for log file "S0000079`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`.LOG" to Comple ted archive for log file S0000080.LOG to DB2REMOTE://<AWS S3 Bucket Name>/<SI D>/log1/db2<sid>/< SID>/NODE0000/LOGS TREAM0000/C0000001 / from /db2/<SID>/ log_dir/NODE0000/ LOGSTREAM0000/. M ESSAGE : ADM1846I Completed archive for log file "S0000080 .LOG" to Comple ted archive for log file S0000081.LOG to DB2REMOTE://<AWS S3 Bucket Name> / <SID>/log1/db2<si d>/<SID>/NODE0000/ LOGSTREAM0000/C000 0001/ from /db2/<SID >/log_dir/NODE0000/ LOGSTREAM0000/.` Pesan ini mengonfirmasi bahwa file log DB2 transaksi tertutup sedang diarsipkan ke penyimpanan Amazon S3 (jarak jauh).	

Sumber daya terkait

AWS dokumentasi layanan

- AWS kredensyal keamanan (dokumentasi IAM)
- Berikan hak istimewa paling sedikit (dokumentasi IAM)
- Mengelola kunci akses untuk pengguna IAM (dokumentasi IAM)
- Praktik terbaik keamanan dalam IAM (dokumentasi IAM)

Sumber daya IBM

- Basis Data IBM Db2
- logarchmeth1 - Parameter konfigurasi metode arsip log primer
- logarchmeth2 - Parameter konfigurasi metode arsip log sekunder
- Penyimpanan jarak jauh

Memigrasikan data dari lingkungan Hadoop lokal ke Amazon S3 menggunakan AWS untuk Amazon S3 DistCp PrivateLink

Jason Owens, Andres Cantor, Jeff Klopfenstein, Bruno Rocha Oliveira, dan Samuel Schmidt, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memigrasikan hampir semua jumlah data dari lingkungan Apache Hadoop lokal ke Amazon Web Services (AWS) Cloud dengan menggunakan alat open-source Apache dengan DistCpAWS PrivateLink for Amazon Simple Storage Service (Amazon S3). Alih-alih menggunakan internet publik atau solusi proxy untuk memigrasikan data, Anda dapat menggunakan AWS PrivateLink untuk Amazon S3 untuk memigrasikan data ke Amazon S3 melalui koneksi jaringan pribadi antara pusat data lokal dan Amazon Virtual Private Cloud (Amazon VPC). Jika Anda menggunakan entri DNS di Amazon Route 53 atau menambahkan entri dalam file /etc/hosts di semua node cluster Hadoop lokal, maka Anda secara otomatis diarahkan ke titik akhir antarmuka yang benar.

Panduan ini memberikan petunjuk penggunaan DistCp untuk memigrasikan data ke AWS Cloud. DistCp adalah alat yang paling umum digunakan, tetapi alat migrasi lainnya tersedia. Misalnya, Anda dapat menggunakan alat AWS offline seperti AWS Snowball atau AWS Snowmobile, atau alat AWS online seperti AWS Storage Gateway atau AWS. DataSync Selain itu, Anda dapat menggunakan alat sumber terbuka lainnya seperti NiFiApache.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan koneksi jaringan pribadi antara pusat data lokal dan AWS Cloud
- Hadoop, dipasang di tempat dengan DistCp
- Pengguna Hadoop dengan akses ke data migrasi di Hadoop Distributed File System (HDFS)
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Izin untuk memasukkan objek ke dalam bucket S3

Batasan

Batasan cloud pribadi virtual (VPC) berlaku PrivateLink untuk AWS untuk Amazon S3. Untuk informasi selengkapnya, lihat properti dan batasan titik akhir antarmuka PrivateLink serta kuota AWS (PrivateLink dokumentasi AWS).

AWS PrivateLink untuk Amazon S3 tidak mendukung hal berikut:

- Titik Akhir Standar Proses Informasi Federal (FIPS)
- Titik akhir situs web
- Titik akhir warisan global

Arsitektur

Tumpukan teknologi sumber

- Cluster Hadoop dengan terpasang DistCp

Tumpukan teknologi target

- Amazon S3
- Amazon VPC

Arsitektur target

Diagram menunjukkan cara administrator Hadoop menggunakan DistCp untuk menyalin data dari lingkungan lokal melalui koneksi jaringan pribadi, seperti AWS Direct Connect, ke Amazon S3 melalui titik akhir antarmuka Amazon S3.

Alat

Layanan AWS

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Alat-alat lainnya

- Apache Hadoop DistCp (salinan terdistribusi) adalah alat yang digunakan untuk menyalin antar-cluster besar dan intra-cluster. DistCp menggunakan Apache MapReduce untuk distribusi, penanganan kesalahan dan pemulihan, dan pelaporan.

Epik

Migrasikan data ke AWS Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir untuk AWS PrivateLink untuk Amazon S3.	1. Masuk ke AWS Management Console dan buka konsol Amazon VPC. 2. Pada panel navigasi, pilih Endpoints, lalu pilih Create Endpoint. 3. Untuk kategori Layanan, pilih layanan AWS. 4. Di kotak pencarian, masukkan s3, lalu tekan Enter. 5. Dalam hasil pencarian , pilih com.amazonaws. < your-aws-region >.s3 nama layanan di mana nilai dalam kolom Type adalah Interface.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Untuk VPC, pilih VPC Anda. Untuk Subnet, pilih subnet Anda. 7. Untuk grup Keamanan, pilih atau buat grup keamanan yang memungkinkan TCP 443. 8. Tambahkan tag berdasarkan kebutuhan Anda dan kemudian pilih Buat titik akhir.	
Verifikasi titik akhir dan temukan entri DNS.	1. Buka konsol Amazon VPC, pilih Endpoints, lalu pilih endpoint yang Anda buat sebelumnya. 2. Pada tab Detail, temukan entri DNS pertama untuk nama DNS. Ini adalah entri DNS Regional. Saat Anda menggunakan nama DNS ini, permintaan bergantian antara entri DNS khusus untuk Availability Zones. 3. Pilih tab Subnet. Anda dapat menemukan alamat elastic network interface endpoint di setiap Availability Zone.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa aturan firewall dan konfigurasi perutean.	Untuk mengonfirmasi bahwa aturan firewall Anda terbuka dan konfigurasi jaringan Anda diatur dengan benar, gunakan Telnet untuk menguji titik akhir pada port 443. Contoh:
	`$ telnet vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com 443`
	`Trying 10.104.88.6...`
	`Connected to vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com.`
	`...`
	`$ telnet vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com 443`
	`Trying 10.104.71.141...`
	`Connected to vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com.`
	Note
	Jika Anda menggunakan entri Regional, pengujian yang
		Administrator jaringan, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	berhasil menunjukkan bahwa DNS bergantia n antara dua alamat IP yang dapat Anda lihat di tab Subnet untuk titik akhir yang Anda pilih di konsol VPC Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan resolusi nama.	Anda harus mengonfig urasi resolusi nama untuk memungkinkan Hadoop mengakses titik akhir antarmuka Amazon S3. Anda tidak dapat menggunakan nama endpoint itu sendiri. Sebaliknya, Anda harus menyelesaikan `<your-bucket-name>.s3.<your-aws-region>.amazonaws.com` atau `*.s3.<your-aws-region>.amazonaws.com`. Untuk informasi lebih lanjut tentang batasan penamaan ini, lihat Memperkenalkan klien Hadoop S3A (situs web Hadoop). Pilih salah satu opsi konfigurasi berikut: • Gunakan DNS lokal untuk menyelesaikan alamat IP pribadi titik akhir. Anda dapat mengganti perilaku untuk semua bucket atau bucket yang dipilih. Untuk informasi selengkapnya, lihat “Opsi 2: Akses Amazon S3 menggunakan Zona Kebijakan Respons Sistem Nama Domain (DNS RPZ)” di Akses hybrid aman ke	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon S3 menggunakan AWS (postingan blog PrivateLink AWS). • Konfigurasikan DNS lokal untuk meneruskan lalu lintas secara kondisional ke titik akhir masuk resolver di VPC. Lalu lintas diteruskan ke Route 53. Untuk informasi selengkapnya, lihat “Opsi 3: Meneruskan permintaan DNS dari lokal menggunakan Amazon Route 53 Resolver Endpoint” di Akses hybrid aman ke Amazon S3 menggunakan AWS (postingan blog AWS). PrivateLink • Edit file /etc/hosts pada semua node di cluster Hadoop Anda. Ini adalah solusi sementara untuk pengujian dan tidak direkomendasikan untuk produksi. Untuk mengedit file /etc/hosts, tambahkan entri untuk salah satu atau. `<your-bucket-name>.s3.<your-aws-region>.amazonaws.com` `s3.<your-aws-region>.amazonaws.com` File /etc/hosts tidak dapat	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memiliki beberapa alamat IP untuk entri. Anda harus memilih satu alamat IP dari salah satu Availabil ity Zones, yang kemudian menjadi satu titik kegagalan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan otentikasi untuk Amazon S3.	Untuk mengautentikasi ke Amazon S3 melalui Hadoop, kami menyarankan Anda mengekspor kredensi peran sementara ke lingkungan Hadoop. Untuk informasi selengkapnya, lihat Mengautentikasi dengan S3 (situs web Hadoop). Untuk pekerjaan yang berjalan lama, Anda dapat membuat pengguna dan menetapkan kebijakan yang memiliki izin untuk memasukkan data ke dalam bucket S3 saja. Kunci akses dan kunci rahasia dapat disimpan di Hadoop, hanya dapat diakses oleh DistCp pekerjaan itu sendiri dan ke administrator Hadoop. Untuk informasi selengkapnya tentang menyimpan rahasia, lihat Menyimpan rahasia dengan Penyedia Kredenal Hadoop (situs web Hadoop). Untuk informasi selengkapnya tentang metode autentikasi lainnya, lihat Cara mendapatkan kredensil peran IAM untuk digunakan dengan akses CLI ke akun AWS dalam dokumentasi Pusat Identitas AWS IAM (penerus AWS Single Sign-On).	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Untuk menggunakan kredensil sementara, tambahkan kredensil sementara ke file kredensil Anda, atau jalankan perintah berikut untuk mengekspor kredensil ke lingkungan Anda:
	`export AWS_SESSION_TOKEN=SECRET-SESSION-TOKEN`
	`export AWS_ACCESS_KEY_ID=SESSION-ACCESS-KEY`
	`export AWS_SECRET_ACCESS_KEY=SESSION-SECRET-KEY`
	Jika Anda memiliki kunci akses tradisional dan kombinasi tombol rahasia, jalankan perintah berikut:
	`export AWS_ACCESS_KEY_ID=my.aws.key`
	`export AWS_SECRET_ACCESS_KEY=my.secret.key`
	Note
	Jika Anda menggunakan kunci akses dan kombinasi tombol rahasia, maka ubah penyedia kredensi dalam DistCp perintah	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dari "org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider" ke. "org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider"	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Transfer data dengan menggunakan DistCp	Untuk digunakan DistCp untuk mentransfer data, jalankan perintah berikut: `hadoop distcp -Dfs.s3a.aws.credentials.provider=\"org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider" \ -Dfs.s3a.access.key="${AWS_ACCESS_KEY_ID}" \ -Dfs.s3a.secret.key="${AWS_SECRET_ACCESS_KEY}" \ -Dfs.s3a.session.token="${AWS_SESSION_TOKEN}" \ -Dfs.s3a.path.style.access=true \ -Dfs.s3a.connection.ssl.enabled=true \ -Dfs.s3a.endpoint=s3.<your-aws-region>.amazonaws.com \ hdfs:///user/root/ s3a://<your-bucket-name>` **Note** Wilayah AWS titik akhir tidak ditemukan secara otomatis saat Anda menggunakan DistCp perintah	Insinyur migrasi, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan AWS PrivateLink untuk Amazon S3. Hadoop 3.3.2 dan versi yang lebih baru menyelesaikan masalah ini dengan mengaktifkan opsi untuk secara eksplisit menyetel Wilayah AWS dari bucket S3. Untuk informasi selengkapnya, lihat S3A untuk menambahkan opsi fs.s3a.endpoint.region untuk menyetel wilayah AWS (situs web Hadoop). Untuk informasi lebih lanjut tentang penyedia S3A tambahan, lihat Konfigurasi Klien S3A Umum (situs web Hadoop). Misalnya, jika Anda menggunakan enkripsi, Anda dapat menambahkan opsi berikut ke rangkaian perintah di atas tergantung pada jenis enkripsi Anda: `-Dfs.s3a.server-side-encryption-algorithm=AES-256 [or SSE-C or SSE-KMS]`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Untuk menggunakan titik akhir antarmuka dengan S3A, Anda harus membuat entri alias DNS untuk nama Regional S3 (misalnya ,`s3.<your-aws-region>.amazonaws.com`) ke titik akhir antarmuka. Lihat bagian Konfigurasi autentikasi untuk Amazon S3 untuk petunjuk. Solusi ini diperlukan untuk Hadoop 3.3.2 dan versi sebelumnya. Versi S3A yang akan datang tidak memerlukan solusi ini. Jika Anda memiliki masalah tanda tangan dengan Amazon S3, tambahkan opsi untuk menggunakan penandatanganan Signature Version 4 (SigV4): `-Dmapreduce.map.java.opts="-Dcom.amazonaws.services.s3.enableV4=true"`	

Lebih banyak pola

- Akses Layanan AWS dari IBM z/OS dengan menginstal AWS CLI
- Secara otomatis mengarsipkan item ke Amazon S3 menggunakan DynamoDB TTL
- Secara otomatis mencadangkan database SAP HANA menggunakan Systems Manager dan EventBridge
- Cadangkan dan arsipkan data mainframe ke Amazon S3 menggunakan BMC AMI Cloud Data
- Buat pipeline layanan ETL untuk memuat data secara bertahap dari Amazon S3 ke Amazon Redshift menggunakan AWS Glue
- Konfigurasikan pencatatan pemanggilan model di Amazon Bedrock dengan menggunakan AWS CloudFormation
- Mengonversi dan membongkar data EBCDIC ke ASCII di AWS dengan menggunakan Python
- Mengkonversi VARCHAR2 (1) tipe data untuk Oracle ke tipe data Boolean untuk Amazon Aurora PostgreSQL
- Buat definisi tugas Amazon ECS dan pasang sistem file pada EC2 instans menggunakan Amazon EFS
- Mengirimkan data DynamoDB ke Amazon S3 menggunakan Kinesis Data Streams dan Firehose dengan AWS CDK
- Menerapkan sistem file Lustre untuk pemrosesan data berkinerja tinggi dengan menggunakan Terraform dan DRA
- Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB
- Identifikasi bucket Amazon S3 publik dengan menggunakan Security Hub AWS Organizations CSPM
- Memigrasi server SFTP lokal untuk menggunakan AWS AWS Transfer for SFTP
- Migrasikan tabel partisi Oracle ke PostgreSQL menggunakan AWS DMS
- Migrasi data dari Microsoft Azure Blob ke Amazon S3 dengan menggunakan Rclone
- Migrasikan nilai Oracle CLOB ke baris individual di PostgreSQL di AWS
- Migrasi sistem file bersama dalam migrasi AWS besar
- Migrasikan kumpulan data kecil dari lokasi ke Amazon S3 menggunakan AWS SFTP
- Pantau Amazon Aurora untuk instance tanpa enkripsi
- Pindahkan file mainframe langsung ke Amazon S3 menggunakan Transfer Family

- Hapus EC2 entri Amazon dalam hal yang sama Akun AWS dari AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi
- Jalankan beban kerja stateful dengan penyimpanan data persisten dengan menggunakan Amazon EFS di Amazon EKS dengan AWS Fargate
- Berhasil mengimpor bucket S3 sebagai tumpukan AWS CloudFormation
- Sinkronisasi data antara sistem file Amazon EFS di Wilayah AWS yang berbeda dengan menggunakan AWS DataSync
- Lihat detail snapshot EBS untuk akun atau organisasi AWS

Alat developer

Topik

- DevOps
- Infrastruktur
- Aplikasi web & seluler

DevOps

Topik

- Percepat MLOps dengan Backstage dan template Amazon SageMaker AI swalayan
- Mengotomatiskan operasi AWS infrastruktur dengan menggunakan Amazon Bedrock
- Otomatiskan CloudFront pembaruan saat titik akhir penyeimbang beban berubah dengan menggunakan Terraform
- Otomatiskan CodeGuru ulasan Amazon untuk aplikasi AWS CDK Python dengan menggunakan Tindakan GitHub
- Otomatiskan penerapan Rantai Pasokan AWS data lake dalam pengaturan multi-repositori dengan menggunakan GitHub Actions, Artifactory, dan Terraform
- Mengotomatiskan penilaian sumber daya AWS
- Instal sistem SAP secara otomatis dengan menggunakan alat sumber terbuka
- Mengotomatiskan portofolio AWS Service Catalog dan penerapan produk dengan menggunakan AWS CDK
- Otomatiskan manajemen pipeline dinamis untuk menerapkan solusi perbaikan terbaru di lingkungan Gitflow dengan menggunakan dan AWS Service Catalog AWS CodePipeline
- Mengotomatiskan penghapusan AWS CloudFormation tumpukan dan sumber daya terkait
- Otomatiskan konsumsi dan visualisasi metrik kustom Amazon MWAA di Amazon Managed Grafana dengan menggunakan Terraform
- Otomatiskan penerapan set tumpukan dengan menggunakan AWS dan AWS CodePipeline CodeBuild
- Secara otomatis melampirkan kebijakan terkelola AWS untuk Systems Manager ke profil EC2 instans menggunakan Cloud Custodian dan AWS CDK
- Membuat CI/CD pipeline dan kluster Amazon ECS secara otomatis untuk layanan mikro menggunakan AWS CDK
- Buat dan dorong gambar Docker ke Amazon ECR menggunakan GitHub Actions dan Terraform
- Buat dan uji aplikasi iOS dengan AWS CodeCommit, AWS CodePipeline, dan AWS Device Farm
- Konfigurasikan otentikasi TLS timbal balik untuk aplikasi yang berjalan di Amazon EKS
- Otomatiskan pembuatan sumber daya WorkSpaces Aplikasi Amazon menggunakan AWS CloudFormation
- Buat parser log khusus untuk Amazon ECS menggunakan router log Firelens

- Buat kerangka orkestrasi sumber daya berbasis API menggunakan Actions dan Terragrunt GitHub
- Buat permintaan tarik otomatis untuk infrastruktur yang dikelola Terraform AWS dengan menggunakan Tindakan GitHub
- Buat pipeline CI dinamis untuk proyek Java dan Python secara otomatis
- Menyebarkan kenari CloudWatch Synthetics dengan menggunakan Terraform
- Terapkan ChatOps solusi untuk mengelola hasil pemindaian SAST dengan menggunakan Amazon Q Developer dalam tindakan kustom aplikasi obrolan dan CloudFormation
- Terapkan sistem agen di Amazon Bedrock dengan framework CreWai dengan menggunakan Terraform
- Menerapkan pekerjaan AWS Glue dengan pipeline AWS CodePipeline CI/CD
- Menerapkan kode di beberapa Wilayah AWS menggunakan AWS CodePipeline, AWS CodeCommit, dan AWS CodeBuild
- Terapkan beban kerja dari DevOps pipeline Azure ke kluster Amazon EKS pribadi
- Jalankan kueri Amazon Redshift SQL dengan menggunakan Terraform
- Ekspor laporan AWS Backup dari seluruh organisasi di AWS Organizations sebagai file CSV
- Ekspor tag untuk daftar EC2 instans Amazon ke file CSV
- Buat CloudFormation template AWS yang berisi aturan terkelola AWS Config menggunakan Troposfer
- Berikan instans SageMaker notebook akses sementara ke CodeCommit repositori di akun AWS lain
- Menerapkan strategi percabangan GitHub Flow untuk lingkungan multi-akun DevOps
- Menerapkan strategi percabangan Gitflow untuk lingkungan multi-akun DevOps
- Menerapkan strategi percabangan Trunk untuk lingkungan multi-akun DevOps
- Menerapkan pemindaian Checkov kustom terpusat untuk menegakkan kebijakan sebelum menerapkan infrastruktur AWS
- Menerapkan diagnostik Kubernetes yang didukung AI dan pemecahan masalah dengan integrasi K8SGPT dan Amazon Bedrock
- Secara otomatis mendeteksi perubahan dan memulai CodePipeline jaringan pipa yang berbeda untuk monorepo di CodeCommit
- Integrasikan repositori Bitbucket dengan AWS Amplify menggunakan AWS CloudFormation
- Luncurkan CodeBuild proyek di seluruh akun AWS menggunakan Step Functions dan fungsi proxy Lambda

- Kelola failover Multi-AZ untuk cluster EMR dengan menggunakan Application Recovery Controller
- Kelola blue/green penerapan layanan mikro ke beberapa akun dan Wilayah dengan menggunakan layanan kode AWS dan kunci Multi-wilayah AWS KMS
- Pantau repositori Amazon ECR untuk izin wildcard menggunakan AWS dan AWS Config CloudFormation
- Optimalkan penerapan tanpa server multi-akun dengan menggunakan alur kerja dan Tindakan AWS CDK GitHub
- Menyediakan AWS Service Catalog produk berdasarkan AWS CloudFormation template dengan menggunakan GitHub Actions
- Menyediakan peran IAM dengan hak istimewa paling sedikit dengan menerapkan solusi mesin penjual otomatis peran
- Publikasikan CloudWatch metrik Amazon ke file CSV
- Hapus EC2 entri Amazon di Akun AWS seberang AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi
- Hapus EC2 entri Amazon dalam hal yang sama Akun AWS dari AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi
- Jalankan pengujian unit untuk pekerjaan ETL Python dalam AWS Glue menggunakan kerangka pytest
- Siapkan CI/CD pipeline dengan menggunakan AWS CodePipeline dan AWS CDK
- Siapkan logging terpusat pada skala perusahaan dengan menggunakan Terraform
- Siapkan end-to-end enkripsi untuk aplikasi di Amazon EKS menggunakan cert-manager dan Let's Encrypt
- Sederhanakan penerapan aplikasi multi-tenant Amazon EKS dengan menggunakan Flux
- Merampingkan pengembangan dan penerapan bot Amazon Lex dengan menggunakan alur kerja otomatis
- Berlangganan beberapa titik akhir email ke topik SNS dengan menggunakan sumber daya khusus
- Koordinasikan ketergantungan sumber daya dan eksekusi tugas dengan menggunakan konstruksi AWS Fargate WaitCondition kait
- Gunakan repositori sumber Git pihak ketiga di AWS CodePipeline
- Buat CI/CD pipeline untuk memvalidasi konfigurasi Terraform dengan menggunakan AWS CodePipeline
- Lebih banyak pola

Percepat MLOps dengan Backstage dan template Amazon SageMaker AI swalayan

Ashish Bhatt, Shashank Hiremath, dan Shivanshu Suryakar, Amazon Web Services

Ringkasan

Organizations yang menggunakan sistem machine learning operations (MLOps) menghadapi tantangan yang signifikan dalam penskalaan, standardisasi, dan mengamankan infrastruktur ML mereka. Pola ini memperkenalkan pendekatan transformatif yang menggabungkan Backstage, portal pengembang open source, dengan Amazon SageMaker AI dan modul hardened infrastructure as code (IAc) untuk meningkatkan cara tim ilmu data Anda dapat mengembangkan, menyebarkan, dan mengelola alur kerja ML.

Modul IAc untuk pola ini disediakan dalam repositori GitHub AWS AIOps modul. Modul-modul ini menawarkan template pra-bangun untuk menyiapkan infrastruktur ML dan menciptakan lingkungan ML yang konsisten. Namun, ilmuwan data sering berjuang untuk menggunakan template ini secara langsung karena mereka membutuhkan keahlian infrastruktur. Menambahkan portal pengembang seperti Backstage menciptakan cara yang mudah digunakan bagi ilmuwan data untuk menerapkan lingkungan ML standar tanpa perlu memahami detail infrastruktur yang mendasarinya.

Dengan menggunakan Backstage sebagai platform swalayan dan mengintegrasikan templat SageMaker AI yang telah dikonfigurasi sebelumnya, Anda dapat:

- Mempercepat waktu untuk menilai inisiatif ML Anda.
- Membantu menegakkan keamanan dan tata kelola yang konsisten.
- Menyediakan ilmuwan data dengan lingkungan standar dan sesuai.
- Mengurangi overhead operasional dan kompleksitas infrastruktur.

Pola ini memberikan solusi yang mengatasi tantangan kritis MLOps dan juga menyediakan kerangka kerja yang dapat diskalakan dan berulang yang memungkinkan inovasi sambil mempertahankan standar organisasi.

Target audiens

Pola ini ditujukan untuk khalayak luas yang terlibat dalam ML, arsitektur cloud, dan rekayasa platform dalam suatu organisasi. Hal ini mencakup:

- Insinyur ML yang ingin menstandarisasi dan mengotomatiskan penerapan alur kerja ML.
- Ilmuwan data yang menginginkan akses swalayan ke lingkungan ML yang telah dikonfigurasi dan sesuai.
- Insinyur platform yang bertanggung jawab untuk membangun dan memelihara platform pengembang internal dan infrastruktur bersama.
- Arsitek cloud yang merancang solusi cloud yang terukur, aman, dan hemat biaya untuk. MLOps
- DevOps insinyur yang tertarik untuk memperluas praktik integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) ke penyediaan infrastruktur dan alur kerja ML.
- Pemimpin teknis dan manajer yang mengawasi inisiatif ML dan ingin meningkatkan produktivitas tim, tata kelola, dan waktu ke pasar.

Untuk informasi selengkapnya tentang MLOps tantangan, MLOps modul SageMaker AI, dan bagaimana solusi yang disediakan oleh pola ini dapat memenuhi kebutuhan tim ML Anda, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- AWS Identity and Access Management (IAM) peran dan izin untuk menyediakan sumber daya ke dalam Akun AWS
- Pemahaman tentang konsep Amazon SageMaker Studio, SageMaker Projects, SageMaker Pipelines, dan SageMaker Model Registry
- Pemahaman tentang prinsip-prinsip IAc dan pengalaman dengan alat-alat seperti AWS Cloud Development Kit (AWS CDK)

Batasan

- Cakupan template terbatas. Saat ini, solusi hanya mendukung AIOps modul SageMaker terkait AI dari solusi yang lebih luasAIOps . Modul lain, seperti Ray on Amazon Elastic Kubernetes Service (Amazon EKS), Apache Airflow MLflow, dan fine-tuning untuk Amazon Bedrock, belum tersedia sebagai template Backstage.
- Pengaturan default yang tidak dapat dikonfigurasi. Template menggunakan konfigurasi default tetap dari AIOps SageMaker modul tanpa kustomisasi. Anda tidak dapat mengubah jenis instans, ukuran penyimpanan, konfigurasi jaringan, atau kebijakan keamanan melalui antarmuka Backstage, yang membatasi fleksibilitas untuk kasus penggunaan tertentu.

- AWS-hanya dukungan. Platform ini dirancang khusus untuk AWS penerapan dan tidak mendukung skenario multicloud. Organizations yang menggunakan layanan cloud di luar AWS Cloud tidak dapat menggunakan template ini untuk kebutuhan infrastruktur ML mereka.
- Manajemen kredensi manual. Anda harus secara manual memberikan AWS kredensil Anda untuk setiap penerapan. Solusi ini tidak menyediakan integrasi dengan penyedia identitas perusahaan AWS IAM Identity Center, atau rotasi kredenal otomatis.
- Manajemen siklus hidup terbatas. Template tidak memiliki fitur manajemen siklus hidup sumber daya yang komprehensif seperti kebijakan pembersihan otomatis, rekomendasi pengoptimalan biaya, dan deteksi penyimpangan infrastruktur. Anda harus mengelola dan memantau sumber daya yang digunakan secara manual setelah pembuatan.

Arsitektur

Diagram berikut menunjukkan arsitektur solusi untuk portal pengembang terpadu yang menstandarisasi dan mempercepat penyebaran infrastruktur ML dengan AI di seluruh lingkungan. SageMaker

Dalam arsitektur ini:

1. AWS cetak biru modernisasi aplikasi menyediakan penyiapan infrastruktur dengan kluster Amazon EKS sebagai dasar untuk kerangka kerja Cloud Native Operational Excellence (CNOE). Solusi komprehensif ini mengatasi tantangan manajemen infrastruktur cloud-native yang kompleks dengan menyediakan platform pengembang internal (IDP) yang dapat diskalakan. Cetak biru menawarkan pendekatan terstruktur untuk menyiapkan infrastruktur yang kuat dan fleksibel yang dapat beradaptasi dengan kebutuhan organisasi Anda yang terus berkembang.
2. Kerangka open source CNOE mengkonsolidasikan DevOps alat dan memecahkan fragmentasi ekosistem melalui pendekatan rekayasa platform terpadu. Dengan menyatukan alat dan teknologi yang berbeda, ini menyederhanakan lanskap kompleks pengembangan cloud-native, sehingga tim Anda dapat fokus pada inovasi alih-alih manajemen rantai alat. Kerangka kerja ini menyediakan metodologi standar untuk memilih, mengintegrasikan, dan mengelola alat pengembangan.
3. Dengan CNOE, Backstage digunakan sebagai solusi out-of-the-box dalam cluster Amazon EKS. Backstage dibundel dengan otentikasi yang kuat melalui Keycloak dan alur kerja penerapan komprehensif melalui Argo CD. Platform terintegrasi ini menciptakan lingkungan terpusat untuk mengelola proses pengembangan dan menyediakan satu tempat bagi tim untuk mengakses, menyebarkan, dan memantau infrastruktur dan aplikasi mereka di berbagai lingkungan.

4. GitHub Repositori berisi templat AIOps perangkat lunak yang telah dikonfigurasi sebelumnya yang mencakup seluruh SageMaker siklus hidup AI. Template ini memenuhi kebutuhan infrastruktur ML yang penting, termasuk penyediaan SageMaker Studio, pelatihan model, saluran inferensi, dan pemantauan model. Template ini membantu Anda mempercepat inisiatif ML Anda dan memastikan konsistensi di berbagai proyek dan tim.
5. GitHub Actions mengimplementasikan alur kerja otomatis yang secara dinamis memicu penyediaan sumber daya melalui utilitas Seed-Farmer. Pendekatan ini mengintegrasikan katalog Backstage dengan repositori AIOps modul dan menciptakan proses penyebaran infrastruktur yang efisien. Otomatisasi mengurangi intervensi manual, meminimalkan kesalahan manusia, dan memastikan penciptaan infrastruktur yang cepat dan konsisten di berbagai lingkungan.
6. AWS CDKIni membantu Anda mendefinisikan dan menyediakan infrastruktur sebagai kode, dan memastikan penyebaran sumber daya yang dapat diulang, aman, dan sesuai di seluruh yang ditentukan. Akun AWS Pendekatan ini memberikan tata kelola maksimum dengan intervensi manual minimal, sehingga Anda dapat membuat templat infrastruktur standar yang dapat dengan mudah direplikasi, dikendalikan versi, dan diaudit.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- Amazon SageMaker AI adalah layanan ML terkelola yang membantu Anda membangun dan melatih model ML dan kemudian menerapkannya ke lingkungan host yang siap produksi.

Alat-alat lainnya

- Backstage adalah kerangka kerja open source yang membantu Anda membangun portal pengembang internal.
- GitHub Actions adalah CI/CD platform yang mengotomatiskan alur kerja pengembangan perangkat lunak, termasuk tugas-tugas seperti membangun, menguji, dan menyebarkan kode.

Repositori kode

Pola ini menggunakan kode dan template dari GitHub repositori berikut:

- AIOps platform pengembang internal (IDP) dengan repositori Backstage
- SageMaker Modul terkait AI dari repositori AWS AIOps modul
- Rekayasa modern pada AWS repositori

Implementasi

Implementasi ini menggunakan pola penyebaran tingkat produksi untuk Backstage dari rekayasa Modern pada repositori. AWS Pendekatan ini secara signifikan menyederhanakan proses penyiapan sambil menggabungkan praktik AWS terbaik untuk keamanan dan skalabilitas.

Bagian Epik dari pola ini menguraikan pendekatan implementasi. Untuk petunjuk step-by-step penyebaran yang terperinci, lihat panduan penerapan komprehensif yang tersedia di platform pengembang AIOps internal (IDP) dengan repositori Backstage. Implementasinya meliputi:

- Penyebaran platform Backstage Awal
- Integrasi template SageMaker perangkat lunak dengan Backstage
- Mengkonsumsi dan memelihara template Backstage

Panduan penerapan juga mencakup panduan untuk pemeliharaan berkelanjutan, pemecahan masalah, dan penskalaan platform.

Praktik terbaik

Ikuti praktik terbaik ini untuk membantu memastikan keamanan, tata kelola, dan keunggulan operasional dalam implementasi MLOps infrastruktur Anda.

Manajemen Template

- Jangan pernah membuat perubahan besar pada template langsung.
- Selalu uji pembaruan secara menyeluruh sebelum penerapan produksi.
- Pertahankan versi template yang jelas dan terdokumentasi dengan baik.

Keamanan

- Pin GitHub Actions ke algoritma hash aman khusus (SHAs) untuk membantu mencegah serangan rantai pasokan.
- Gunakan peran IAM dengan hak istimewa paling sedikit dengan izin terperinci.
- Simpan kredensyal sensitif di GitHub Rahasia dan. AWS Secrets Manager
- Jangan pernah membuat hardcode kredenal di template.

Tata kelola dan pelacakan

- Menerapkan standar penandaan sumber daya yang komprehensif.
- Aktifkan pelacakan biaya yang tepat dan pemantauan kepatuhan.
- Pertahankan jejak audit yang jelas untuk perubahan infrastruktur.

Panduan ini memberikan dasar yang kuat untuk menerapkan praktik terbaik ini dengan menggunakan modul Backstage, SageMaker AI, dan IAc.

Epik

Siapkan lingkungan ML Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan Belakang Panggung.	Langkah ini menggunakan cetak biru dalam rekayasa Modern pada AWS repositor i untuk membangun infrastru ktur yang kuat dan dapat diskalakan yang menginteg rasikan beberapa untuk membuat IDP terpusat Layanan AWS untuk alur kerja ML. Ikuti instruksi di bagian penyebaran Backstage dari panduan penerapan untuk mengkloning repositori, menginstal dependensi, mem-bootstrap variabel lingkunga	Insinyur platform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n AWS CDK konfigurasi, dan menerapkan platform Backstage. Infrastruktur menggunak an Amazon EKS sebagai platform orkestrasi kontainer untuk menerapkan komponen IDP. Arsitektur Amazon EKS mencakup konfigurasi jaringan yang aman untuk menetapka n isolasi jaringan yang ketat dan mengontrol pola akses. Platform ini terintegrasi dengan mekanisme otentikasi untuk membantu mengamank an akses pengguna di seluruh layanan dan lingkungan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan templat SageMaker AI Anda.	Langkah ini menggunakan skrip di platform pengembang GitHub AIOps internal (IDP) dengan repositori Backstage. Ikuti petunjuk di bagian pengaturan SageMaker templat dari panduan penerapan untuk mengkloning repositori, mengatur prasyarat, dan menjalankan skrip penyiapan. Proses ini membuat repositori yang berisi template SageMaker AI yang diperlukan untuk integrasi dengan Backstage.	Insinyur platform
Integrasikan template SageMaker AI dengan Backstage.	Ikuti petunjuk di bagian integrasi SageMaker templat dari panduan penerapan untuk mendaftarkan templat SageMaker AI Anda. Langkah ini menginteg rasikan AIOps modul (templat SageMaker AI dari langkah terakhir) ke dalam penerapan Backstage Anda sehingga Anda dapat melayani sendiri kebutuhan infrastruktur ML Anda.	Insinyur platform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan template SageMaker AI dari Backstage.	Ikuti petunjuk di bagian Menggunakan SageMaker templat pada panduan penerapan untuk mengakses portal Backstage dan membuat lingkungan ML di SageMaker Studio. Di portal Backstage, Anda dapat memilih dari templat SageMaker AI yang tersedia, termasuk opsi untuk lingkungan SageMaker Studio, SageMaker notebook, templat SageMaker proyek khusus, dan pipeline penerapan model. Setelah Anda memberikan parameter konfigurasi, platform membuat repositori khusus secara otomatis dan menyediakan AWS sumber daya melalui GitHub Actions dan Seed-Farmer. Anda dapat memantau kemajuan melalui log GitHub Tindakan dan katalog komponen Backstage.	Ilmuwan data, Insinyur data, Pengembang

Mengelola template untuk tata kelola dan kepatuhan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui templat SageMaker AI.	Untuk memperbarui template SageMaker AI di Backstage, ikuti langkah-langkah ini. 1. Memodifikasi konten template: a. Buat perubahan yang diperlukan dalam `template.yaml` file atau dengan mengedit file di `skeleton/` direktori. b. Uji parameter, tindakan, atau struktur file baru secara lokal atau di lingkungan pengembangan. 2. Perubahan uji: a. Gunakan UI Backstage atau CLI `@backstage/create-app` () untuk membuat perancah komponen pengujian dengan menggunakan template yang diperbarui. b. Validasi bahwa semua langkah berjalan dengan sukses dan kode yang dihasilkan memenuhi harapan Anda. 3. Komit dan dorong perubahan:	Insinyur platform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	a. Dorong perubahan ke repositori Git tempat template disimpan. Jika template terdaftar di cabang tertentu (misalnya,`main`), pembaruan akan tercermin secara otomatis. b. Jika Anda menggunakan pembuatan versi (lihat langkah selanjutnya), pastikan versi atau tag yang benar diperbarui.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Membuat dan mengelola beberapa versi template.	Untuk melanggar perubahan atau peningkatan, Anda mungkin ingin membuat beberapa versi template SageMaker AI.
	
	1. Gunakan tag atau cabang Git untuk setiap versi; misalnya:
	
	`git checkout -b v2.0.0`
	`git push origin v2.0.0`
	
	2. (Opsional tetapi disarankan) Daftarkan setiap versi secara terpisah.
	
	Di Backstage, Anda dapat mendaftarkan versi template yang berbeda sebagai entitas terpisah dalam katalog, di mana setiap entitas menunjuk ke cabang atau tag tertentu. Misalnya (untuk `.yaml` file):
	
	`metadata:`
	`name: node-service-template-v2`
	`description: Node.js service template - Version 2`
	`spec:`
	`type: template`
	`lifecycle: experimental`
		Insinyur platform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`version: '2.0.0'` 3. Komunikasikan perubahan dengan jelas dengan memelihara `CHANGELOG.md` file di repositori template. Dalam file ini, dokumentasikan fitur atau perubahan mana yang diperkenalkan di setiap versi template. 4. Menghentikan versi template yang lebih lama, jika perlu: a. Tandai sebagai usang dalam deskripsi template. b. Hapus versi dari katalog jika tidak lagi diperlukan.	

Perluas lingkungan ML Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perluas cakupan template di luar SageMaker AI.	Solusi saat ini hanya mengimplementasikan template SageMaker terkait AI AIOps . Anda dapat memperluas lingkungan ML dengan menambahkan AIOps modul dan menginteg rasikan template perangkat lunak khusus untuk tambahan Layanan AWS dan aplikasi.	Insinyur platform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda dapat membuat ini dengan menggunakan antarmuka desainer template di Backstage, dengan menerapkan tindakan scaffolder kustom, atau dengan memelihara repositori template dengan metadata standar. Platform ini mendukung pembuatan versi template, berbagi lintas tim, dan alur kerja validasi untuk konsistensi. Untuk informasi selengkapnya, lihat dokumentasi Backstage. Anda juga dapat menerapkan pola pewarisan template untuk membuat versi khusus dari template dasar. Ekstensibilitas ini memungkinkan Anda mengelola beragam AWS sumber daya dan aplikasi di luar SageMaker AI sambil mempertahankan pengalaman pengembang yang disederhanakan dan mempertahankan standar organisasi Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan injeksi parameter dinamis.	Template saat ini menggunakan konfigurasi default tanpa penyesuaian, dan menjalankan CLI Seed-Farmer untuk menyebarkan sumber daya dengan variabel default. Anda dapat memperluas konfigurasi default dengan menggunakan injeksi parameter dinamis untuk konfigurasi khusus modul.	Insinyur platform
Meningkatkan keamanan dan kepatuhan.	Untuk meningkatkan keamanan dalam pembuatan sumber daya AWS, Anda dapat mengaktifkan integrasi kontrol akses berbasis peran (RBAC) dengan sistem masuk tunggal (SSO), SAMP, OpenID Connect (OIDC), dan kebijakan sebagai penegakan kode.	Insinyur platform
Tambahkan pembersihan sumber daya otomatis.	Anda dapat mengaktifkan fitur untuk kebijakan pembersihan otomatis, dan juga menambahkan deteksi dan remediasi drift infrastruktur.	Insinyur platform

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus infrastruktur Backstage dan sumber daya SageMaker AI.	Setelah selesai menggunakan lingkungan ML, ikuti petunjuk di bagian Pembersihan dan manajemen sumber daya panduan penerapan untuk menghapus infrastruktur Backstage dan menghapus sumber daya SageMaker AI di lingkungan ML Anda.	Insinyur platform

Pemecahan masalah

Isu	Solusi
AWS CDK kegagalan bootstrap	Verifikasi AWS kredensil dan konfigurasi Wilayah.
Masalah akses klaster Amazon EKS	Periksa konfigurasi kubectl dan izin IAM.
Masalah konektivitas Application Load Balancer	Pastikan bahwa grup keamanan mengizinkan lalu lintas masuk pada port 80/443.
GitHub masalah integrasi	Verifikasi izin GitHub token dan akses organisasi.
SageMaker Kegagalan penerapan AI	Periksa Layanan AWS kuota dan izin IAM.

Sumber daya terkait

- Rekayasa platform (dalam panduan AWS Cloud Adoption Framework: Perspektif platform)
- Dokumentasi Amazon SageMaker AI
- Template Perangkat Lunak Backstage (situs web Backstage)

- AIOps repositori modul (kumpulan modul IAc yang dapat digunakan kembali untuk ML)
- AIOps platform pengembang internal (IDP) dengan repositori Backstage
- Rekayasa modern pada AWS repositori
- Situs web Cloud Native Operational Excellence (CNOE)

Informasi tambahan

Tantangan bisnis

Organisasi-organisasi yang memulai atau menskalakan MLOps inisiatif mereka sering menghadapi tantangan bisnis dan teknis ini:

- Lingkungan yang tidak konsisten. Kurangnya pengembangan standar dan lingkungan penyebaran membuat kolaborasi menjadi sulit dan meningkatkan risiko penerapan.
- Overhead penyediaan manual. Menyiapkan infrastruktur ML secara manual dengan bucket SageMaker Studio, Amazon Simple Storage Service (Amazon S3), peran IAM, CI/CD dan pipeline memakan waktu dan rawan kesalahan, dan mengalihkan ilmuwan data dari tugas inti pengembangan model mereka.
- Kurangnya kemampuan untuk ditemukan dan digunakan kembali. Kurangnya katalog terpusat menyulitkan untuk menemukan model, kumpulan data, dan saluran pipa yang ada. Ini mengarah pada pekerjaan yang berlebihan dan kehilangan kesempatan untuk digunakan kembali.
- Tata kelola dan kepatuhan yang kompleks. Memastikan bahwa proyek ML mematuhi kebijakan keamanan organisasi, peraturan privasi data, dan standar kepatuhan seperti Health Insurance Portability and Accountability Act (HIPAA) dan General Data Protection Regulation (GDPR) dapat menjadi tantangan tanpa pagar pembatas otomatis.
- Waktu yang lambat untuk menilai. Efek kumulatif dari tantangan ini menghasilkan siklus hidup proyek ML yang berlarut-larut dan menunda realisasi nilai bisnis dari investasi ML.
- Risiko keamanan. Konfigurasi yang tidak konsisten dan proses manual dapat menimbulkan kerentanan keamanan yang membuatnya sulit untuk menerapkan hak istimewa dan isolasi jaringan yang paling sedikit.

Isu-isu ini memperpanjang siklus pengembangan, meningkatkan overhead operasional, dan menimbulkan risiko keamanan. Sifat berulang dari ML membutuhkan alur kerja berulang dan kolaborasi yang efisien.

Gartner memperkirakan bahwa pada tahun 2026, 80% organisasi rekayasa perangkat lunak akan memiliki tim platform. (Lihat Rekayasa Platform Memberdayakan Pengembang untuk Menjadi Lebih Baik, Lebih Cepat, Lebih Bahagia di situs web Gartner.) Prediksi ini menyoroti bagaimana IDP dapat mempercepat pengiriman perangkat lunak. Sebagai IDP, Backstage membantu memulihkan ketertiban ke infrastruktur yang kompleks sehingga tim dapat memberikan kode berkualitas tinggi dengan cepat dan aman. Mengintegrasikan Backstage dengan AIOps modul yang diperkeras membantu Anda beralih dari pemecahan masalah reaktif ke pencegahan proaktif.

MLOps SageMaker modul

AIOps Modul dalam GitHub repositori yang digunakan untuk pola ini memberikan dasar yang berharga untuk standarisasi MLOps AWS melalui IAc yang dapat digunakan kembali dan dikeraskan. Modul-modul ini merangkum praktik terbaik untuk SageMaker proyek penyediaan, jaringan pipa, dan sumber daya jaringan dan penyimpanan terkait, dengan tujuan untuk mengurangi kompleksitas dan mempercepat penyiapan lingkungan ML. Anda dapat menggunakan template ini untuk berbagai kasus MLOps penggunaan guna menetapkan pola penerapan yang konsisten dan aman yang mendorong pendekatan yang lebih teratur dan efisien terhadap alur kerja ML.

Menggunakan AIOps modul secara langsung seringkali membutuhkan tim platform untuk menyebarkan dan mengelola templat IAC ini, yang dapat menghadirkan tantangan bagi ilmuwan data yang menginginkan akses layanan mandiri. Menemukan dan memahami template yang tersedia, mengonfigurasi parameter yang diperlukan, dan memicu penerapannya mungkin memerlukan navigasi Layanan AWS konsol atau berinteraksi langsung dengan alat IAc. Ini dapat menciptakan gesekan, meningkatkan beban kognitif bagi ilmuwan data yang lebih suka fokus pada tugas-tugas ML, dan berpotensi menyebabkan parameterisasi atau penyimpangan yang tidak konsisten dari standar organisasi jika templat ini tidak dikelola melalui antarmuka yang terpusat dan ramah pengguna. Mengintegrasikan AIOps modul-modul canggih ini dengan IDP seperti Backstage membantu mengatasi tantangan ini dengan memberikan pengalaman swalayan yang efisien, kemampuan untuk ditemukan, dan kontrol tata kelola yang lebih kuat untuk menggunakan blok bangunan standar ini. MLOps

Backstage sebagai IDP

Platform pengembang internal (IDP) adalah lapisan layanan mandiri yang dibangun oleh tim platform untuk menyederhanakan dan menstandarisasi bagaimana pengembang membangun, menyebarkan, dan mengelola aplikasi. Ini mengabstraksi kompleksitas infrastruktur dan memberi pengembang akses mudah ke alat, lingkungan, dan layanan melalui antarmuka terpadu.

Tujuan utama IDP adalah untuk meningkatkan pengalaman dan produktivitas pengembang dengan:

- Mengaktifkan layanan mandiri untuk tugas-tugas seperti pembuatan dan penyebaran layanan.
- Mempromosikan konsistensi dan kepatuhan melalui templat standar.
- Mengintegrasikan alat di seluruh siklus hidup pengembangan (CI/CD, pemantauan, dan dokumentasi).

Backstage adalah portal pengembang open source yang dibuat oleh Spotify dan sekarang menjadi bagian dari Cloud Native Computing Foundation (CNCF). Ini membantu organisasi membangun IDP mereka sendiri dengan menyediakan platform terpusat dan dapat diperluas untuk mengelola komponen perangkat lunak, alat, dan dokumentasi. Dengan Backstage, pengembang dapat:

- Temukan dan kelola semua layanan internal melalui katalog perangkat lunak.
- Buat proyek baru dengan menggunakan templat yang telah ditentukan melalui plugin scaffolder.
- Akses perkakas terintegrasi seperti CI/CD pipeline, dasbor Kubernetes, dan sistem pemantauan dari satu lokasi.
- Pertahankan dokumentasi berbasis penurunan harga yang konsisten melalui. TechDocs

FAQ

Apa perbedaan antara menggunakan template Backstage ini versus menerapkan SageMaker Studio secara manual melalui konsol? SageMaker

Template Backstage memberikan beberapa keunggulan dibandingkan penerapan AWS konsol manual, termasuk konfigurasi standar yang mengikuti praktik terbaik organisasi, penyebaran IAC otomatis menggunakan Seed-Farmer dan, kebijakan keamanan bawaan dan tindakan kepatuhan AWS CDK, dan integrasi dengan alur kerja pengembang organisasi Anda. GitHub Template juga membuat penerapan yang dapat direproduksi dengan kontrol versi, yang membuatnya lebih mudah untuk mereplikasi lingkungan di berbagai tahap (pengembangan, pementasan, produksi) dan menjaga konsistensi di seluruh tim. Selain itu, template mencakup kemampuan pembersihan otomatis dan terintegrasi dengan sistem manajemen identitas organisasi Anda melalui Backstage. Penerapan manual melalui konsol memerlukan AWS keahlian yang mendalam dan tidak menyediakan kontrol versi atau tingkat standardisasi dan tata kelola yang sama dengan yang ditawarkan template. Untuk alasan ini, penerapan konsol lebih cocok untuk eksperimen satu kali daripada lingkungan produksi produksi.

Apa itu Petani Benih dan mengapa solusi ini menggunakannya?

Seed-Farmer adalah alat orkestrasi AWS penyebaran yang mengelola modul infrastruktur dengan menggunakan. AWS CDK Pola ini menggunakan Seed-Farmer karena menyediakan komponen infrastruktur standar yang dapat digunakan kembali yang dirancang khusus untuk AI/ML beban kerja, menangani dependensi kompleks antara secara Layanan AWS otomatis, dan memastikan penerapan yang konsisten di berbagai lingkungan.

Apakah saya perlu menginstal AWS CLI untuk menggunakan template ini?

Tidak, Anda tidak perlu menginstal AWS CLI di komputer Anda. Template berjalan sepenuhnya melalui GitHub Actions di cloud. Anda memberikan AWS kredensi Anda (kunci akses, kunci rahasia, dan token sesi) melalui antarmuka Backstage, dan penerapan terjadi secara otomatis di lingkungan Tindakan. GitHub

Berapa lama waktu yang dibutuhkan untuk menerapkan lingkungan SageMaker Studio?

Penyebaran SageMaker Studio yang khas membutuhkan waktu 15-25 menit untuk diselesaikan. Ini termasuk AWS CDK bootstrap (2-3 menit), pengaturan toolchain Seed-Farmer (3-5 menit), dan pembuatan sumber daya (10-15 menit). Waktu yang tepat tergantung pada Anda Wilayah AWS dan kompleksitas pengaturan jaringan Anda.

Bisakah saya menerapkan beberapa SageMaker lingkungan dalam hal yang sama? Akun AWS

Ya, Anda bisa. Setiap penerapan menciptakan sumber daya dengan nama unik berdasarkan nama komponen yang Anda berikan dalam template. Namun, perhatikan Layanan AWS kuota: Setiap akun dapat memiliki jumlah SageMaker domain terbatas per Wilayah, jadi periksa kuota Anda sebelum Anda membuat beberapa lingkungan.

Mengotomatiskan operasi AWS infrastruktur dengan menggunakan Amazon Bedrock

Ishwar Chauthaiwale dan Anand Bukkapatnam Tirumala, Amazon Web Services

Ringkasan

Dalam solusi cloud native, mengotomatisasi operasi infrastruktur umum memainkan peran penting dalam menjaga lingkungan yang efisien, aman, dan hemat biaya. Operasi penanganan secara manual memakan waktu dan rentan terhadap kesalahan manusia. Selain itu, anggota tim dengan berbagai tingkat AWS keahlian perlu melakukan tugas-tugas ini sambil memastikan kepatuhan terhadap protokol keamanan. Pola ini menunjukkan cara menggunakan Amazon Bedrock untuk mengotomatiskan operasi AWS infrastruktur umum melalui pemrosesan bahasa alami (NLP).

Pola ini dapat membantu organisasi untuk mengembangkan kode yang dapat digunakan kembali, modular, dan aman untuk menyebarkan infrastruktur berbasis AI generatif di berbagai lingkungan. Melalui fokusnya pada infrastruktur sebagai kode (IAc) dan otomatisasi, ia memberikan DevOps manfaat utama termasuk kontrol versi, penerapan yang konsisten, pengurangan kesalahan, penyediaan yang lebih cepat, dan kolaborasi yang lebih baik.

Pola ini mengimplementasikan arsitektur aman yang memungkinkan tim mengelola operasi yang terkait dengan kunci Layanan AWS termasuk:

- Manajemen versi bucket Amazon Simple Storage Service (Amazon S3)
- Pembuatan snapshot Amazon Relational Database Service (Amazon RDS)
- Manajemen instans Amazon Elastic Compute Cloud (Amazon EC2)

Arsitektur ini menggunakan titik akhir Amazon Virtual Private Cloud (Amazon VPC) dan jaringan pribadi untuk komunikasi yang aman, AWS Lambda dengan fungsi yang beroperasi sebagai pelaksana tugas dalam subnet pribadi. Amazon S3 menyediakan manajemen data dan mengimplementasikan peran dan izin komprehensif AWS Identity and Access Management (IAM) untuk memastikan kontrol akses yang tepat. Solusi ini tidak menyertakan fitur riwayat obrolan, dan obrolan tidak disimpan.

Prasyarat dan batasan

- Aktif Akun AWS.

- Langkah-langkah kontrol akses yang tepat harus dilakukan untuk membantu mengamankan dan mengontrol akses. Contoh kontrol akses termasuk penggunaan AWS Systems Manager, akses model dasar, peran IAM untuk penerapan, dan peran berbasis layanan, menonaktifkan akses publik ke bucket Amazon S3, dan menyiapkan antrian surat mati.
- Kunci yang dikelola pelanggan AWS Key Management Service (AWS KMS).
- AWS Command Line Interface (AWS CLI) versi 2 atau yang lebih baru, diinstal dan dikonfigurasi pada lingkungan penyebaran.
- AWS Penyedia Terraform versi 4 atau yang lebih baru diinstal dan dikonfigurasi.
- Terraform versi 1.5.7 atau yang lebih baru diinstal dan dikonfigurasi.
- Tinjau dan Tentukan skema OpenAPI untuk grup tindakan agen Anda di Amazon Bedrock untuk membantu melindungi dari akses yang tidak sah dan menjaga integritas data.
- Akses diaktifkan di Anda Akun AWS untuk Amazon Titan Text Embeddings v2 yang diperlukan dan model dasar Claude 3.5 Sonnet atau Claude 3 Haiku. Untuk menghindari kegagalan penerapan, konfirmasikan bahwa penerapan target Anda Wilayah AWS mendukung model yang diperlukan.
- Virtual Private Cloud (VPC) yang dikonfigurasi yang mengikuti praktik terbaik AWS Well Architected Framework.
- Tinjauan lengkap tentang kebijakan AI Bertanggung Jawab Amazon.

Versi produk

- Embeddings Teks Amazon Titan v2
- Antropik Claude 3.5 Soneta atau Claude 3 Haiku
- AWS Penyedia Terraform versi 4 atau yang lebih baru
- Terraform versi 1.5.7 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Arsitektur solusi terdiri dari beberapa lapisan yang bekerja sama untuk memproses permintaan bahasa alami dan menjalankan AWS operasi yang sesuai:

1. Pengguna membuat permintaan operasi melalui konsol obrolan Amazon Bedrock.

2. Chatbot menggunakan Pangkalan Pengetahuan Amazon Bedrock untuk pemrosesan permintaan. Ini mengimplementasikan model Amazon Titan Text Embeddings v2 untuk pemrosesan bahasa alami.
3. Jika prompt pengguna menyertakan permintaan tindakan, grup tindakan Amazon Bedrock menggunakan model Anthropic Claude 3 Haiku atau Claude 3.5 Sonnet (tergantung pilihan Anda) untuk logika eksekusi dan mendefinisikan operasi melalui skema OpenAPI.
4. Grup tindakan mencapai titik akhir VPC Amazon yang digunakan AWS PrivateLink untuk komunikasi layanan yang aman.
5. AWS Lambda Fungsi ini dicapai melalui titik akhir Amazon VPC untuk layanan Amazon Bedrock.
6. Fungsi Lambda adalah mesin eksekusi utama. Berdasarkan permintaan, fungsi Lambda memanggil API untuk melakukan tindakan pada file. Layanan AWS Fungsi Lambda juga menangani routing dan eksekusi operasi.
7. Permintaan Layanan AWS get API dari fungsi Lambda dan operasi terkait dilakukan.
8. Fungsi Lambda menghitung payload keluaran yang dipahami oleh Amazon Bedrock.
9. Payload ini dikirim ke Amazon Bedrock dengan menggunakan PrivateLink untuk komunikasi layanan yang aman. Model bahasa besar (LLM) yang digunakan oleh Amazon Bedrock memahami muatan ini dan mengubahnya menjadi format yang dapat dimengerti manusia.
10. Outputnya kemudian ditampilkan kepada pengguna di konsol obrolan Amazon Bedrock.

Solusinya memungkinkan operasi utama berikut:

- Amazon S3 — Aktifkan pembuatan versi bucket untuk kontrol versi.
- Amazon RDS - Buat snapshot database untuk cadangan.
- Amazon EC2 — Buat daftar instans dan kontrol awal dan penghentian instans.

Alat

Layanan AWS

- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon OpenSearch Serverless adalah konfigurasi tanpa server sesuai permintaan untuk Amazon Service. OpenSearch
- AWS PrivateLinkmembantu Anda membuat koneksi pribadi searah dari cloud pribadi virtual Anda (VPCs) ke layanan di luar VPC.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat lainnya

- Git adalah sistem kontrol versi terdistribusi open source.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di GitHub infra-ops-orchestratoraws-samples/ repositori.

Praktik terbaik

- Pantau log eksekusi Lambda secara teratur. Untuk informasi selengkapnya, lihat Memantau dan memecahkan masalah fungsi Lambda. Untuk informasi selengkapnya tentang praktik terbaik, lihat Praktik terbaik untuk bekerja dengan AWS Lambda fungsi.
- Tinjau konfigurasi keamanan secara berkala untuk memastikan kepatuhan terhadap persyaratan organisasi Anda. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositor i pada mesin lokal Anda, jalankan perintah berikut: `git clone "git@gith ub.com:aws-samples /infra-ops-orchest rator.git" cd infra-ops-orchestr ator`	AWS DevOps, DevOps insinyur
Edit variabel lingkungan.	Edit `terraform.tfvars` file. di direktori root dari repositori kloning. Tinjau placeholder yang ditunjukkan oleh`[XXXXX]`, dan perbarui sesuai dengan lingkungan Anda.	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat infrastruktur.	Untuk membuat infrastruktur, jalankan perintah berikut:
	`terraform init`
	`terraform plan`
	Tinjau rencana eksekusi dengan cermat. Jika perubahan yang direncanakan dapat diterima, maka jalankan perintah berikut:
	`terraform apply --auto-approve`
		AWS DevOps, DevOps insinyur

Akses solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses solusinya.	Setelah penerapan berhasil, ikuti langkah-langkah berikut untuk menggunakan antarmuka berbasis obrolan: 1. Untuk mengakses Infrastructure Orchestra tor Assistant, masuk ke Konsol Manajemen AWS menggunakan peran IAM dengan izin Amazon Bedrock, dan buka konsol Amazon Bedrock di. https://console.aws.amazon	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	.com/bedrock/ Pilih Agen dari panel navigasi kiri. Kemudian, pilih Asisten Orkestrator Infrastruktur di bagian Agen. 2. Pastikan sumber daya target ada di AWS lingkungan Anda untuk saran berikut, lalu coba contoh operasi ini: • Buat cadangan snapshot dari instans Amazon RDS dengan menanyakan: 'Buat snapshot instance RDS [nama instans] ' • Aktifkan pembuatan versi di bucket Amazon S3 dengan menanyakan: 'Aktifkan pembuatan versi untuk bucket [bucket-name] ' • Buat daftar instans Amazon EC2 dengan menanyakan: 'Daftar semua instans EC2' • Mulai atau hentikan instans Amazon EC2 dengan menanyakan: 'Mulai instans EC2 [instance-id] 'atau 'Hentikan instans EC2 [instance-id]'	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Catatan: Ganti nilai dalam tanda kurung dengan nama sumber daya aktual atau IDs dari AWS lingkungan Anda.	

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hapus sumber daya yang dibuat.	Untuk menghapus semua infrastruktur yang dibuat oleh pola ini, jalankan perintah berikut:
	`terraform plan -destroy`
	Tinjau rencana penghancuran dengan hati-hati. Jika penghapusan yang direncanakan dapat diterima, maka jalankan perintah berikut:
	`terraform destroy`
	Catatan: Perintah ini akan menghapus semua sumber daya yang dibuat oleh pola ini secara permanen. Perintah akan meminta konfirmasi sebelum menghapus sumber daya apa pun.
		AWS DevOps, DevOps insinyur

Pemecahan masalah

Isu	Solusi
Perilaku agen	Untuk informasi tentang masalah ini, lihat Menguji dan memecahkan masalah perilaku agen dalam dokumentasi Amazon Bedrock.
Masalah jaringan Lambda	Untuk informasi tentang masalah ini, lihat Memecahkan masalah jaringan di Lambda dalam dokumentasi Lambda.
Izin IAM	Untuk informasi tentang masalah ini, lihat Memecahkan Masalah IAM di dokumentasi IAM.

Sumber daya terkait

- Membuat snapshot DB untuk instans DB AZ tunggal untuk Amazon RDS
- Tentukan skema OpenAPI untuk grup tindakan agen Anda di Amazon Bedrock
- Mengaktifkan pembuatan versi pada bucket
- Bagaimana Agen Amazon Bedrock bekerja
- Ambil data dan hasilkan respons AI dengan Pangkalan Pengetahuan Amazon Bedrock
- Akses Layanan Dengan Aman AWS PrivateLink
- Hentikan dan mulai instans Amazon EC2
- Gunakan grup tindakan untuk menentukan tindakan yang harus dilakukan agen Anda

Otomatiskan CloudFront pembaruan saat titik akhir penyeimbang beban berubah dengan menggunakan Terraform

Tamilselvan P, Mohan Annam, dan Naveen Suthar, Amazon Web Services

Ringkasan

Saat pengguna Amazon Elastic Kubernetes Service (Amazon EKS) menghapus dan menginstal ulang konfigurasi ingress mereka melalui bagan Helm, Application Load Balancer (ALB) baru akan dibuat. Ini menciptakan masalah karena Amazon CloudFront terus mereferensikan catatan DNS ALB lama. Akibatnya, layanan yang ditujukan ke titik akhir ini tidak akan dapat dijangkau. (Untuk detail selengkapnya tentang alur kerja bermasalah ini, lihat Informasi tambahan.)

Untuk mengatasi masalah ini, pola ini menjelaskan penggunaan AWS Lambda fungsi kustom yang dikembangkan dengan Python. Fungsi Lambda ini secara otomatis mendeteksi ketika ALB baru dibuat melalui aturan Amazon. EventBridge Dengan menggunakan AWS SDK for Python (Boto3), fungsi tersebut kemudian memperbarui CloudFront konfigurasi dengan alamat DNS ALB baru, memastikan bahwa lalu lintas diarahkan ke titik akhir yang benar.

Solusi otomatis ini mempertahankan kontinuitas layanan tanpa perutean atau latensi tambahan. Proses ini membantu memastikan bahwa CloudFront selalu mereferensikan titik akhir DNS ALB yang benar, bahkan ketika infrastruktur yang mendasarinya berubah.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Contoh aplikasi web untuk pengujian dan validasi yang digunakan di Amazon EKS dengan menggunakan Helm. Untuk informasi selengkapnya, lihat Menerapkan aplikasi dengan Helm di Amazon EKS di dokumentasi Amazon EKS.
- Konfigurasikan CloudFront untuk merutekan panggilan ke ALB yang dibuat oleh pengontrol ingress Helm. Untuk informasi selengkapnya, lihat Menginstal AWS Load Balancer Controller dengan Helm di dokumentasi Amazon EKS dan Membatasi akses ke Application Load Balancer dalam dokumentasi. CloudFront
- Terraform diinstal dan dikonfigurasi di ruang kerja lokal.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Terraform versi 1.0.0 atau yang lebih baru
- AWS Penyedia Terraform versi 4.20 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Solusi ini melakukan langkah-langkah berikut:

1. Pengontrol ingress Amazon EKS membuat Application Load Balancer (ALB) baru setiap kali ada Helm restart atau deployment.
2. EventBridge mencari acara pembuatan ALB.
3. Acara pembuatan ALB memicu fungsi Lambda.
4. Fungsi Lambda telah digunakan berdasarkan python 3.9 dan menggunakan API boto3 untuk memanggil. Layanan AWS Fungsi Lambda memperbarui CloudFront entri dengan nama DNS penyeimbang beban terbaru, yang diterima dari peristiwa create load balancer.

Alat

Layanan AWS

- Amazon CloudFront mempercepat distribusi konten web Anda dengan mengirimkannya melalui jaringan pusat data di seluruh dunia, yang menurunkan latensi dan meningkatkan kinerja.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS

Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS

Alat lainnya

- Python adalah bahasa pemrograman komputer tujuan umum.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub aws-cloudfront-automation-terraform-samples.

Epik

Siapkan workstation lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan dan konfigurasikan Git CLI.	Untuk menginstal dan mengkonfigurasi antarmuka baris perintah Git (CLI) di workstation lokal Anda, ikuti instruksi Memulai — Menginstal Git dalam dokumentasi Git.	DevOps insinyur
Buat folder proyek dan tambahkan file.	1. Pergi ke GitHub repositori pola, dan pilih tombol Kode. 2. Pada dialog Clone, pilih tab HTTPS. Di Clone	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggunakan URL web, salin URL yang ditampilkan. 3. Buat folder di mesin lokal Anda. Beri nama dengan nama proyek Anda. 4. Buka terminal di mesin lokal Anda, dan arahkan ke folder ini. 5. Untuk mengkloning repositori git pola ini, jalankan perintah berikut: `git clone https://github.com/aws-samples/aws-cloudfront-automation-terraform-samples` 6. Setelah repositori telah dikloning, gunakan perintah berikut untuk pergi ke direktori kloning: `cd <directory name>/cloudfront-update` Buka proyek ini di Integrated Development Environment (IDE) pilihan Anda.	

Menyediakan arsitektur target menggunakan konfigurasi Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan solusinya.	Untuk menyebarkan sumber daya di target Akun AWS,	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	gunakan langkah-langkah berikut:
	1. Pergi ke folder `cloudfront-update` .
	2. Perbarui `terraform.tfvars` file dengan file`cloudfront_distribution_id` .
	3. Untuk mengatur AWS profil Anda, jalankan perintah berikut: Wilayah AWS
	`export AWS_REGION={{ REGION }}`
	4. Untuk menginisialisasi Terraform, jalankan perintah berikut:
	`terraform init`
	5. Untuk memvalidasi Terraform, jalankan perintah berikut:
	`terraform validate`
	6. Untuk membuat rencana eksekusi Terraform, jalankan perintah berikut:
	`terraform plan`
	7. Untuk menerapkan tindakan dari`terraform plan`, jalankan perintah berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`terraform apply`	

Verifikasi penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan.	1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol Amazon di https://console.aws.amazon.com/cloudfront/v4/home. 2. Di panel navigasi kiri, pilih Distribusi dan kemudian buka distribusi. CloudFront 3. Pada tab Origins, verifikasi bahwa nama asal dan pemetaan asal memiliki catatan DNS ALB yang diperbarui.	DevOps insinyur

Membersihkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan infrastruktur.	Untuk membersihkan infrastruktur yang Anda buat sebelumnya, gunakan langkah-langkah berikut: 1. Jalankan perintah berikut: `terraform destroy`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Untuk mengkonfirmasi perintah hancurkan, masukkan`yes`.	

Pemecahan masalah

Isu	Solusi
Kesalahan memvalidasi kredensi penyedia	Saat Anda menjalankan Terraform `apply` atau `destroy` perintah dari mesin lokal Anda, Anda mungkin mengalami kesalahan yang mirip dengan berikut ini: `Error: configuring Terraform AWS Provider: error validating provider credentials: error calling sts:GetCallerIdentity: operation error STS: GetCallerIdentity, https response error StatusCode: 403, RequestID: 123456a9-fbc1-40ed-b8d8-513d0133ba7f, api error InvalidClientTokenId: The security token included in the request is invalid.` Kesalahan ini disebabkan oleh kedaluwarsa token keamanan untuk kredensil yang digunakan dalam konfigurasi mesin lokal Anda. Untuk mengatasi kesalahan, lihat Mengatur dan melihat pengaturan konfigurasi dalam dokumentasi AWS Command Line Interface (AWS CLI).

Sumber daya terkait

AWS sumber daya

- Batasi akses ke Application Load Balancers
- Rute lalu lintas internet dengan AWS Load Balancer Controller

Dokumentasi Terraform

- AWS Penyedia
- Instal Terraform
- Negara Terpencil

Informasi tambahan

Alur kerja bermasalah

Diagram menunjukkan alur kerja berikut:

1. Saat pengguna mengakses aplikasi, panggilan masuk ke CloudFront.
2. CloudFront merutekan panggilan ke Application Load Balancer (ALB) masing-masing.
3. ALB mencakup alamat IP target yang merupakan alamat IP pod aplikasi. Dari sana, ALB memberikan hasil yang diharapkan kepada pengguna.

Namun, alur kerja ini menunjukkan masalah. Penerapan aplikasi terjadi melalui bagan Helm. Setiap kali ada penerapan atau jika seseorang memulai ulang Helm, ingress masing-masing juga dibuat ulang. Akibatnya, pengontrol penyeimbang beban eksternal menciptakan kembali ALB. Juga, selama setiap pembuatan ulang, ALB dibuat ulang dengan nama DNS yang berbeda. Karena itu, CloudFront akan memiliki entri basi di pengaturan asal. Karena entri basi ini, aplikasi tidak akan dapat dijangkau oleh pengguna. Masalah ini mengakibatkan downtime bagi pengguna.

Solusi alternatif

Solusi lain yang mungkin adalah membuat DNS eksternal untuk ALB dan kemudian mengarahkannya ke titik akhir zona host pribadi Amazon Route 53 di. CloudFront Namun, pendekatan ini menambahkan lompatan lain dalam alur aplikasi, yang dapat menyebabkan latensi aplikasi. Solusi fungsi Lambda pola ini tidak mengganggu aliran arus.

Otomatiskan CodeGuru ulasan Amazon untuk aplikasi AWS CDK Python dengan menggunakan Tindakan GitHub

Vanitha Dontireddy dan Sarat Chandra Pothula, Amazon Web Services

Ringkasan

Catatan: Mulai 7 November 2025, Anda tidak dapat membuat asosiasi repositori baru di Amazon Reviewer. CodeGuru Untuk mempelajari tentang layanan dengan kemampuan yang mirip dengan CodeGuru Reviewer, lihat Perubahan ketersediaan Amazon CodeGuru Reviewer di dokumentasi CodeGuru Reviewer.

Pola ini menampilkan integrasi tinjauan kode CodeGuru otomatis Amazon untuk aplikasi AWS Cloud Development Kit (AWS CDK) Python, yang diatur GitHub melalui Tindakan. Solusinya menerapkan arsitektur tanpa server yang didefinisikan dengan Python. AWS CDK Dengan mengotomatiskan analisis kode ahli dalam pipeline pengembangan, pendekatan ini dapat melakukan hal berikut untuk proyek AWS CDK Python:

- Tingkatkan kualitas kode.
- Merampingkan alur kerja.
- Maksimalkan manfaat komputasi tanpa server.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) versi 2.9.11 atau yang lebih baru, diinstal dan dikonfigurasi.
- GitHub Akun aktif dan GitHub repositori dengan izin alur kerja baca dan tulis serta pembuatan permintaan tarik (PR) oleh GitHub Tindakan untuk memastikan alur kerja PR beroperasi dengan benar.
- Peran OpenID Connect (OIDC) dalam GitHub Actions untuk menerapkan solusi di file. Akun AWS Untuk membuat peran, gunakan AWS CDK konstruksi.

Batasan

- Amazon CodeGuru Profiler mendukung aplikasi yang ditulis dalam semua bahasa Java virtual machine (JVM) (seperti Scala dan Kotlin) dan runtime dan Python 3.6 atau yang lebih baru.
- Amazon CodeGuru Reviewer mendukung asosiasi dengan repositori kode Java dan Python hanya dari penyedia sumber berikut AWS CodeCommit:, Bitbucket GitHub,, Enterprise Cloud, dan Enterprise GitHub Server. GitHub Selain itu, repositori Amazon Simple Storage Service (Amazon S3) hanya didukung melalui Actions. GitHub
- Tidak ada cara otomatis untuk mencetak temuan selama pipeline integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD). Sebaliknya, pola ini menggunakan GitHub Actions sebagai metode alternatif untuk menangani dan menampilkan temuan.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat layanan AWS menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan arsitektur untuk solusi ini.

Seperti yang ditunjukkan dalam diagram, ketika pengembang membuat permintaan tarik (PR) untuk ditinjau, GitHub Tindakan memicu langkah-langkah berikut:

1. Asumsi peran IAM — Pipeline menggunakan peran IAM yang ditentukan dalam GitHub Rahasia untuk melakukan tugas penerapan.
2. Analisis kode
   - CodeGuru Reviewer menganalisis kode yang disimpan di bucket Amazon S3. Ini mengidentifikasi cacat dan memberikan rekomendasi untuk perbaikan dan pengoptimalan.
   - CodeGuru Pemindaian keamanan untuk pelanggaran kebijakan dan kerentanan.
3. Tinjauan temuan
   - Pipeline mencetak tautan ke dasbor temuan di output konsol.
   - Jika temuan kritis terdeteksi, pipa segera gagal.
   - Untuk temuan tingkat keparahan tinggi, normal, atau rendah, pipa berlanjut ke langkah berikutnya.
4. Persetujuan PR
   - Seorang pengulas harus menyetujui PR secara manual.

- Jika PR ditolak, pipeline gagal dan menghentikan langkah penerapan lebih lanjut.

5. Penyebaran CDK — Setelah persetujuan PR, proses penyebaran CDK dimulai. Ini mengatur yang berikut Layanan AWS dan sumber daya:
   - CodeGuru Profiler
   - AWS Lambda fungsi
   - Antrean Amazon Simple Queue Service (Amazon SQS)
6. Pembuatan data pembuatan profil — Untuk menghasilkan data profil yang cukup untuk CodeGuru Profiler:
   - Pipeline memanggil fungsi Lambda beberapa kali dengan mengirim pesan ke antrean Amazon SQS secara berkala.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- CDK Toolkit adalah kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan aplikasi Anda AWS CDK .
- Amazon CodeGuru Profiler mengumpulkan data performa runtime dari aplikasi live Anda, dan memberikan rekomendasi yang dapat membantu Anda menyempurnakan performa aplikasi.
- Amazon CodeGuru Reviewer menggunakan analisis program dan pembelajaran mesin untuk mendeteksi potensi cacat yang sulit ditemukan oleh pengembang. Kemudian, CodeGuru Profiler menawarkan saran untuk meningkatkan kode Java dan Python Anda.
- Amazon CodeGuru Security adalah alat keamanan aplikasi statis yang menggunakan pembelajaran mesin untuk mendeteksi pelanggaran dan kerentanan kebijakan keamanan. Ini memberikan saran untuk mengatasi risiko keamanan dan menghasilkan metrik sehingga Anda dapat melacak postur keamanan aplikasi Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- GitHub Actions adalah platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang terintegrasi erat dengan GitHub repositori. Anda dapat menggunakan GitHub Actions untuk mengotomatiskan pipeline build, test, dan deployment.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub amazon-codeguru-suite-cdk-python.

Praktik terbaik

- Patuhi Praktik Terbaik untuk mengembangkan dan menerapkan infrastruktur cloud dengan. AWS CDK
- Ikuti praktik terbaik Keamanan di IAM saat menggunakan Layanan AWS dalam alur kerja GitHub Tindakan, termasuk:
  - Jangan menyimpan kredensi dalam kode repositori Anda.
  - Asumsikan peran IAM untuk menerima kredensil sementara, dan gunakan kredenal sementara jika memungkinkan.
  - Berikan hak istimewa paling sedikit untuk peran IAM yang digunakan dalam alur kerja GitHub Tindakan. Berikan hanya izin yang diperlukan untuk melakukan tindakan dalam alur kerja GitHub Tindakan Anda.
  - Pantau aktivitas peran IAM yang digunakan dalam alur kerja GitHub Tindakan.
  - Putar kredenal jangka panjang apa pun yang Anda gunakan secara berkala.

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan AWS kredensil.	Untuk mengekspor variabel yang menentukan Akun AWS dan Wilayah AWS di mana Anda menerapkan tumpukan, jalankan perintah berikut:
	`export CDK_DEFAU LT_ACCOUNT=<12-digit AWS account number>`
	`export CDK_DEFAU LT_REGION=<AWS Region>`
	AWS Kredensi untuk AWS CDK disediakan melalui variabel lingkungan.
		AWS DevOps, DevOps insinyur

Kloning repositori.	Untuk mengkloning repositor i pada mesin lokal Anda, jalankan perintah berikut: `git clone https://g ithub.com/aws-samp les/amazon-codeguru- suite-cdk-python.git`	AWS DevOps, DevOps insinyur
Instal Toolkit CDK.	Untuk mengonfirmasi bahwa CDK Toolkit diinstal dan untuk memeriksa versi, jalankan perintah berikut: `cdk --version`	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Jika versi CDK Toolkit lebih awal dari 2.27.0, masukkan perintah berikut untuk memperbaruinya ke versi 2.27.0:
	`npm install -g aws-cdk@2.27.0`
	Jika CDK Toolkit tidak diinstal, jalankan perintah berikut untuk menginstalnya:
	`npm install -g aws-cdk@2.27.0 --force`	

Instal dependensi yang diperlukan.	Untuk menginstal dependensi proyek yang diperlukan, jalankan perintah berikut:
	`python -m pip install --upgrade pip`
	`pip install -r requirements.txt`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bootstrap lingkungan CDK.	Untuk mem-bootstrap lingkungan AWS CDK, jalankan perintah berikut:
	`npm install`
	`npm run cdk bootstrap "aws://${ACCOUNT_NUMBER}/${AWS_REGION}"`
	Setelah Anda berhasil mem-bootstrap lingkungan, output berikut akan ditampilkan:
	`# Bootstrapping environment aws://{account}/{region}...`
	`# Environment aws://{account}/{region} bootstrapped`
		AWS DevOps, DevOps insinyur

Menerapkan aplikasi CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sintesis AWS CDK aplikasi.	Untuk mensintesis AWS CDK aplikasi, jalankan perintah berikut: `cdk synth` Untuk informasi selengkapnya tentang perintah ini, lihat cdk synthesize dalam dokumentasi. AWS CDK	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan sumber daya.	Untuk menyebarkan sumber daya, jalankan perintah berikut: `cdk deploy --require-approval never` **Note** `--require-approval never`Bendera berarti CDK akan menyetujui dan mengeksekusi semua perubahan secara otomatis. Ini termasuk perubahan yang biasanya ditandai CDK sebagai membutuhk an tinjauan manual (seperti perubahan kebijakan IAM atau penghapusan sumber daya). Pastikan kode CDK dan CI/CD pipeline Anda telah teruji dengan baik dan aman sebelum Anda menggunakan `--require-approval never` bendera di lingkungan produksi.	AWS DevOps, DevOps insinyur

Buat GitHub rahasia dan token akses pribadi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat rahasia yang diperlukan di GitHub.	Untuk memungkinkan alur kerja GitHub Actions mengakses AWS sumber daya dengan aman tanpa mengekspos informasi sensitif dalam kode repositori Anda, buat rahasia. Untuk membuat rahasia GitHub untuk`ROLE_TO_ASSUME` ,`CodeGuruReviewArtifactBucketName` , dan`AWS_ACCOUNT_ID` , ikuti instruksi dalam Membuat rahasia untuk repositori dalam dokumentasi GitHub Tindakan. Berikut ini adalah informasi lebih lanjut tentang variabel: • `AWS_ACCOUNT_ID` — Akun AWS ID tempat pipa dijalankan. • `CodeGuruReviewArtifactBucketName` — Nama bucket S3 tempat artefak CodeGuru Reviewer disimpan. Pola ini menggunakan nama bucket`codeguru-reviewer-build-artifacts-<ACCOUNT_ID>-<REGION>` .	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `AWS_REGION` — Di Wilayah AWS mana sumber daya berada. • `ROLE_TO_ASSUME` — Nama peran IAM yang diasumsikan oleh pipeline. Pola ini menggunakan nama peran`githubActionsDeployRole` .	
Buat token akses GitHub pribadi.	Untuk menyiapkan cara aman agar alur kerja GitHub Tindakan Anda mengautentikasi dan berinteraksi GitHub, lakukan hal berikut: 1. Untuk membuat token akses GitHub pribadi yang memiliki akses baca dan tulis ke repositori Anda, ikuti petunjuk dalam Mengelola token akses pribadi Anda dalam dokumentasi. GitHub 2. Untuk menyimpan token ini sebagai rahasia repositori untuk GitHub Tindakan, ikuti instruksi di Membuat rahasia untuk repositori di dokumentasi Tindakan. GitHub	AWS DevOps, DevOps insinyur

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Untuk membersihkan aplikasi AWS CDK Python Anda, jalankan perintah berikut: `cdk destroy --all`	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Tampilkan tautan ke temuan dasbor.	Tidak ada cara untuk mencetak temuan selama CI/CD pipa. Sebaliknya, pola ini menggunakan GitHub Actions sebagai metode alternatif untuk menangani dan menampilkan temuan.

Sumber daya terkait

AWS sumber daya

- AWS Cloud Development Kit
- CodeGuru Dokumentasi Amazon
- Amazon S3
- AWS Identity and Access Management
- Amazon Simple Queue Service
- Apa itu AWS Lambda?

GitHub dokumentasi

- Mengkonfigurasi OpenID Connect di Amazon Web Services
- GitHub Tindakan

- Menggunakan kembali alur kerja
- Memicu alur kerja

Otomatiskan penerapan Rantai Pasokan AWS data lake dalam pengaturan multi-repositori dengan menggunakan GitHub Actions, Artifactory, dan Terraform

Keshav Ganesh, Amazon Web Services

Ringkasan

Pola ini menyediakan pendekatan otomatis untuk menyebarkan dan mengelola Rantai Pasokan AWS data lake menggunakan integrasi berkelanjutan multi-repositori dan penerapan berkelanjutan (kemampuan. CI/CD) pipeline. It demonstrates two deployment methods: automated deployment using GitHub Actions workflows, or manual deployment using Terraform directly. Both approaches use Terraform for infrastructure as code (IaC), with the automated method adding GitHub Actions and JFrog Artifactory for enhanced CI/CD

Solusi ini memanfaatkan Rantai Pasokan AWS AWS Lambda, dan Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) untuk membuat infrastruktur data lake, sambil menggunakan salah satu metode penyebaran untuk mengotomatiskan konfigurasi dan pembuatan sumber daya. Otomatisasi ini menghilangkan langkah-langkah konfigurasi manual dan memastikan penerapan yang konsisten di seluruh lingkungan. Selain itu, Rantai Pasokan AWS menghilangkan kebutuhan akan keahlian mendalam dalam mengekstrak, mengubah, dan memuat (ETL) dan dapat memberikan wawasan dan analitik yang didukung oleh Amazon Quick Sight.

Dengan menerapkan pola ini, organisasi dapat mengurangi waktu penyebaran, memelihara infrastruktur sebagai kode, dan mengelola data lake rantai pasokan melalui proses otomatis yang dikendalikan versi. Pendekatan multi-repositori menyediakan kontrol akses berbutir halus dan mendukung penyebaran independen komponen yang berbeda. Tim dapat memilih metode penerapan yang paling sesuai dengan alat dan proses yang ada.

Prasyarat dan batasan

Prasyarat

Pastikan yang berikut ini diinstal pada mesin lokal Anda:

- AWS Command Line Interface (AWS CLI) versi 2
- GitHub CLI
- Python v3.13

- Terraform v1.12 atau yang lebih baru

Pastikan hal-hal berikut sudah ada sebelum penerapan:

- Aktif Akun AWS.
- Virtual Private Cloud (VPC) dengan dua subnet pribadi di Akun AWS pilihan Anda Wilayah AWS .
- Izin yang memadai untuk peran AWS Identity and Access Management (IAM) yang digunakan untuk penyebaran ke layanan berikut:
  - Rantai Pasokan AWS — Akses Penuh lebih disukai untuk menyebarkan komponennya seperti kumpulan data dan aliran integrasi, bersama dengan mengaksesnya dari file. Konsol Manajemen AWS
  - Amazon CloudWatch Logs — Untuk membuat dan mengelola grup CloudWatch log.
  - Amazon Elastic Compute Cloud (Amazon EC2) — Untuk grup EC2 keamanan Amazon dan titik akhir Amazon Virtual Private Cloud (Amazon VPC).
  - Amazon EventBridge — Untuk digunakan oleh Rantai Pasokan AWS.
  - IAM — Untuk membuat peran AWS Lambda layanan.
  - AWS Key Management Service (AWS KMS) — Untuk akses ke bucket artefak Amazon S3 yang AWS KMS keys digunakan dan bucket pementasan Amazon S3. Rantai Pasokan AWS
  - AWS Lambda — Untuk membuat fungsi Lambda yang menyebarkan komponen. Rantai Pasokan AWS
  - Amazon S3 — Untuk akses ke bucket artefak Amazon S3, bucket logging akses server, dan bucket staging. Rantai Pasokan AWS Jika Anda menggunakan penerapan manual, izin untuk bucket artefak Amazon S3 Terraform juga diperlukan.
  - Amazon VPC — Untuk membuat dan mengelola VPC.

Jika Anda lebih suka menggunakan alur kerja GitHub Tindakan untuk penerapan, lakukan hal berikut:

- Siapkan OpenID Connect (OIDC) untuk peran IAM dengan izin yang disebutkan sebelumnya.
- Buat peran IAM dengan izin serupa untuk mengakses. Konsol Manajemen AWS Untuk informasi selengkapnya, lihat Membuat peran untuk memberikan izin kepada pengguna IAM dalam dokumentasi IAM.

Jika Anda lebih suka melakukan penerapan manual, lakukan hal berikut:

- Buat pengguna IAM untuk mengambil peran IAM dengan izin yang disebutkan sebelumnya. Untuk informasi selengkapnya, lihat Membuat peran untuk memberikan izin kepada pengguna IAM dalam dokumentasi IAM.
- Asumsikan peran di terminal lokal Anda.

Jika Anda lebih suka menggunakan alur kerja GitHub Tindakan untuk penerapan, siapkan yang berikut ini:

- Akun JFrog Artifactory untuk mendapatkan nama host, nama pengguna login, dan token akses login.
- Kunci JFrog proyek dan repositori untuk menyimpan artefak.

Batasan

- Rantai Pasokan AWS Instance tidak mendukung teknik transformasi data yang kompleks.
- Rantai Pasokan AWS paling cocok untuk domain rantai pasokan karena menyediakan analitik dan wawasan bawaan. Untuk domain lain, Rantai Pasokan AWS dapat digunakan sebagai penyimpanan data sebagai bagian dari arsitektur danau data.
- Fungsi Lambda yang digunakan dalam solusi ini mungkin perlu ditingkatkan untuk menangani percobaan ulang API dan manajemen memori dalam penerapan skala produksi.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Anda dapat menerapkan solusi ini baik dengan menggunakan alur kerja GitHub Tindakan otomatis atau secara manual menggunakan Terraform.

Penerapan otomatis dengan Tindakan GitHub

Diagram berikut menunjukkan opsi penerapan otomatis yang menggunakan alur kerja GitHub Tindakan. JFrog Artifactory digunakan untuk manajemen artefak. Ini menyimpan informasi sumber daya dan output untuk digunakan dalam penyebaran multi-repositori.

Penerapan manual dengan Terraform

Diagram berikut menunjukkan opsi penerapan manual melalui Terraform. Alih-alih JFrog Artifactory, Amazon S3 digunakan untuk manajemen artefak.

Alur kerja penerapan

Diagram menunjukkan alur kerja berikut:

1. Menyebarkan infrastruktur dan database kumpulan data Rantai Pasokan AWS layanan menggunakan salah satu metode penerapan berikut:
   - Penerapan otomatis — Menggunakan alur kerja GitHub Tindakan untuk mengatur semua langkah penerapan dan menggunakan Artifactory untuk manajemen artefak. JFrog
   - Penerapan manual — Menjalankan perintah Terraform secara langsung untuk setiap langkah penerapan dan menggunakan Amazon S3 untuk manajemen artefak.
2. Buat AWS sumber daya pendukung yang diperlukan untuk operasi Rantai Pasokan AWS layanan:
   - Titik akhir Amazon VPC dan grup keamanan
   - AWS KMS keys
   - CloudWatch Grup log log
3. Membuat dan menyebarkan sumber daya infrastruktur berikut:
   - Fungsi Lambda yang mengelola (membuat, memperbarui, dan menghapus) instance Rantai Pasokan AWS layanan, ruang nama, dan kumpulan data.
   - Rantai Pasokan AWS pementasan bucket Amazon S3 untuk konsumsi data
4. Menerapkan fungsi Lambda yang mengelola alur integrasi antara bucket staging dan dataset. Rantai Pasokan AWS Setelah penerapan selesai, langkah-langkah alur kerja yang tersisa mengelola konsumsi dan analisis data.
5. Konfigurasikan konsumsi data sumber ke bucket Amazon Rantai Pasokan AWS S3 pementasan.
6. Setelah data ditambahkan ke bucket Amazon Rantai Pasokan AWS S3 pementasan, layanan secara otomatis memicu aliran integrasi ke kumpulan data. Rantai Pasokan AWS
7. Rantai Pasokan AWS terintegrasi dengan Quick Sight Analytics untuk menghasilkan dasbor berdasarkan data yang dicerna.

Alat

Layanan AWS

- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua aplikasi Anda Akun AWS dan cloud.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Q in Rantai Pasokan AWS adalah asisten AI generatif interaktif yang membantu Anda mengoperasikan rantai pasokan Anda secara lebih efisien dengan menganalisis data di Rantai Pasokan AWS data lake Anda.
- Amazon Quick Sight adalah layanan intelijen bisnis skala cloud (BI) yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda dalam satu dasbor.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Rantai Pasokan AWSadalah aplikasi terkelola berbasis cloud yang dapat digunakan sebagai penyimpanan data dalam organisasi untuk domain rantai pasokan, yang dapat digunakan untuk menghasilkan wawasan dan melakukan analisis pada data yang dicerna.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur

yang dapat diskalakan. AWSEndpoint Amazon VPC adalah perangkat virtual yang membantu Anda menghubungkan VPC Anda secara pribadi ke yang didukung Layanan AWS tanpa memerlukan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect

Alat-alat lainnya

- GitHub Actions adalah platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang terintegrasi erat dengan GitHub repositori. Anda dapat menggunakan GitHub Actions untuk mengotomatiskan pipeline build, test, dan deployment.
- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.
- JFrog Artifactory menyediakan end-to-end otomatisasi dan manajemen binari dan artefak melalui proses pengiriman aplikasi.
- Python adalah bahasa pemrograman komputer tujuan umum. Pola ini menggunakan Python untuk kode AWS fungsi untuk berinteraksi dengan Rantai Pasokan AWS

.

Praktik terbaik

- Pertahankan keamanan setinggi mungkin saat menerapkan pola ini. Seperti yang dinyatakan dalam Prasyarat, pastikan virtual private cloud (VPC) dengan dua subnet pribadi ada di pilihan Anda. Akun AWS Wilayah AWS
- Gunakan kunci terkelola AWS KMS pelanggan sedapat mungkin, dan berikan izin akses terbatas kepada mereka.
- Untuk mengatur peran IAM dengan akses paling sedikit yang diperlukan untuk menelan data untuk pola ini, lihat Pengambilan Data Aman dari Sistem Sumber ke Amazon S3 di repositori pola ini.

Epik

(Kedua opsi) Siapkan workstation lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositor i pola ini, jalankan perintah	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	berikut di workstation lokal Anda:
	
	`git clone https://github.com/aws-samples/sample-automate-aws-supply-chain-deployment.git`
	`cd ASC-Deployment`	

(Opsi otomatis) Verifikasi prasyarat untuk penerapan.	Pastikan Prasyarat lengkap untuk penerapan otomatis.	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

(Opsi manual) Bersiaplah untuk penyebaran kumpulan Rantai Pasokan AWS data.	Untuk pergi ke `terraform-deployment` direktori`ASC-Datasets`, jalankan perintah berikut:
	`cd ASC-Datasets/terraform-deployment`
	Untuk mengasumsikan peran ARN yang dibuat dalam Prasyarat, jalankan perintah berikut:
	`aws sts assume-role --role-arn <enter AWS user role ARN> --role-session-name <your-session-name>`
	Untuk mengkonfigurasi dan mengekspor variabel lingkungan, jalankan perintah berikut:
	`# Export Environment variables`
	`export REGION=<Enter deployment region>`
	`export REPO_NAME=<Enter Current ASC Datasets dir name>`
	`export PROJECT_NAME="asc-deployment-poc"`
	`export ACCOUNT_ID=<Enter deployment Account ID>`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`export ENVIRONME NT="dev" export LAMBDA_LA YER_TEMP_DIR_TERRA FORM="layerOutput" export LAMBDA_FU NCTION_TEMP_DIR_TE RRAFORM="lambdaOut put" export AWS_USER_ ROLE=<Enter user role ARN for AWS Console access and deployment> export S3_TERRAF ORM_ARTIFACTS_BUCK ET_NAME="$PROJECT_ NAME-$ACCOUNT_ID-$ REGION-terraform-a rtifacts-$ENVIRONM ENT"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

(Opsi manual) Bersiaplah untuk mengelola aliran Rantai Pasokan AWS integrasi dalam penerapan.	Untuk pergi ke `terraform-deployment` direktori`ASC-Integration-Flows` , jalankan perintah berikut:
	`cd ASC-Integration-Flows/terraform-deployment`
	Untuk mengasumsikan peran ARN yang dibuat sebelumnya, jalankan perintah berikut:
	`aws sts assume-role --role-arn <enter AWS user role ARN> --role-session-name <your-session-name>`
	Untuk mengkonfigurasi dan mengekspor variabel lingkungan, jalankan perintah berikut:
	`# Export Environment variables`
	`export REGION=<Enter deployment region>`
	`export REPO_NAME =<Enter Current ASC Integration Flows dir name>`
	`export ASC_DATASET_VARS_REPO=<Enter Current ASC Datasets dir name>  #Must be the same directory name`
		Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	used for ASC Datasets deployment export PROJECT_NAME="asc-deployment-poc" export ACCOUNT_ID=<Enter deployment Account ID> export ENVIRONMENT="dev" export LAMBDA_LAYER_TEMP_DIR_TERRAFORM="layerOutput" export LAMBDA_FUNCTION_TEMP_DIR_TERRAFORM="lambdaOutput" export S3_TERRAFORM_ARTIFACTS_BUCKET_NAME="$PROJECT_NAME-$ACCOUNT_ID-$REGION-terraform-artifacts-$ENVIRONMENT"	

(Opsi otomatis) Menyebarkan Rantai Pasokan AWS kumpulan data menggunakan alur kerja Tindakan GitHub

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin `ASC-Datasets` direktori.	Untuk menyalin `ASC-Datasets` direktori ke lokasi baru, gunakan langkah-langkah berikut: 1. Untuk pergi ke `ASC-Datasets` direktori, jalankan perintah berikut:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cd ASC-Datasets`
	2. Untuk menyalin `ASC-Datasets` direktori ke lokasi baru, jalankan perintah berikut:
	`cp -r ASC-Datas ets ../ASC-Datasets- standalone cd ../ASC-Datasets- standalone`	

Siapkan `ASC-Datasets` direktori.	Untuk mengatur `ASC-Datas ets` sebagai repositori mandiri di organisasi Anda, jalankan perintah berikut: `git init git add . git commit -m "Initial commit: ASC-Datasets standalone repositor y" git remote add origin <INSERT_ASC_DATASE TS_GITHUB_URL> git branch -M dev`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan nama cabang di file alur kerja.github.	Siapkan nama cabang dalam berkas alur kerja deployment seperti yang ditunjukkan pada contoh berikut: `on: workflow_dispatch: push: branches: - dev #Change to any other branch preferred for deployment`	Pemilik aplikasi
Siapkan GitHub lingkungan dan konfigurasikan nilai lingkungan.	Untuk menyiapkan GitHub lingkungan di GitHub organisasi Anda, gunakan petunjuk di GitHub lingkungan Pengaturan di repositori pola ini. Untuk mengonfigurasi nilai lingkungan dalam file alur kerja, gunakan instruksi di Nilai lingkungan pengaturan dalam file alur kerja di repositori pola ini.	Pemilik aplikasi
Memicu alur kerja.	Untuk mendorong perubahan ke GitHub organisasi dan memicu alur kerja penerapan, jalankan perintah berikut: `git push -u origin dev`	AWS DevOps

(Opsi otomatis) Menerapkan alur Rantai Pasokan AWS integrasi menggunakan alur kerja GitHub Tindakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Salin `ASC-Integration-Flows` direktori.	Untuk menyalin `ASC-Integration-Flows` direktori ke lokasi baru, gunakan langkah-langkah berikut:
	1. Untuk pergi ke `ASC-Integration-Flows` direktori, jalankan perintah berikut:
	`cd ASC-Integration-Flows`
	2. Untuk menyalin `ASC-Integration-Flows` direktori ke lokasi baru, jalankan perintah berikut:
	`cp -r ASC-Integration-Flows ../ASC-Integration-Flows-standalone`
	`cd ../ASC-Integration-Flows-standalone`
		AWS DevOps

Siapkan `ASC-Integration-Flows` direktori.	Untuk menyiapkan `ASC-Integration-Flows` direktori sebagai repositori mandiri di organisasi Anda, jalankan perintah berikut:
	`git init`
	`git add .`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`git commit -m "Initial commit: ASC-Integration-Flows standalone repository"`
	`git remote add origin <INSERT_ASC_Integration_Flows_GITHUB_URL>`
	`git branch -M dev`	

Konfigurasikan nama cabang di file alur kerja.github.	Siapkan nama cabang dalam berkas alur kerja deployment seperti yang ditunjukkan pada contoh berikut:
	`on:`
	`workflow_dispatch:`
	`push:`
	`branches:`
	`- dev`
	`#Change to any other branch preferred for deployment`
		Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan GitHub lingkungan dan konfigurasikan nilai lingkungan.	Untuk menyiapkan GitHub lingkungan di GitHub organisasi Anda, gunakan petunjuk di GitHub lingkungan Pengaturan di repositori pola ini. Untuk mengonfigurasi nilai lingkungan dalam file alur kerja, gunakan instruksi di Nilai lingkungan pengaturan dalam file alur kerja di repositori pola ini.	Pemilik aplikasi
Memicu alur kerja.	Untuk mendorong perubahan ke GitHub organisasi dan memicu alur kerja penerapan, jalankan perintah berikut: `git push -u origin dev`	AWS DevOps

(Opsi manual) Terapkan Rantai Pasokan AWS kumpulan data menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka direktori `terraform-deployment` tersebut.	Untuk pergi ke `terraform-deployment` direktori`ASC-Datasets` , jalankan perintah berikut: `cd ASC-Datasets/terraform-deployment`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan bucket Amazon S3 status Terraform.	Untuk menyiapkan bucket Amazon S3 status Terraform, gunakan skrip berikut:
	`# Setup terraform bucket`
	`chmod +x ../scripts/setup-terraform.sh`
	`../scripts/setup-terraform.sh`
		AWS DevOps

Siapkan ember artefak Terraform Amazon S3.	Untuk menyiapkan bucket Amazon S3 artefak Terraform, gunakan skrip berikut:
	`# Setup terraform artifacts bucket`
	`chmod +x ../scripts/setup-terraform-artifacts-bucket.sh`
	`../scripts/setup-terraform-artifacts-bucket.sh`
		AWS DevOps

Siapkan konfigurasi backend dan penyedia Terraform.	Untuk menyiapkan konfigurasi backend dan penyedia Terraform, gunakan skrip berikut:
	`# Setup terraform backend and providers config if they don't exist`
	`chmod +x ../scripts/generate-terraform-config.sh`
	`../scripts/generate-terraform-config.sh`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hasilkan rencana penerapan.	Untuk membuat rencana penyebaran, jalankan perintah berikut:
	`# Run terraform init and validate`
	`terraform init`
	`terraform validate`
	
	`# Run terraform plan`
	`terraform plan \`
	`-var-file="tfInputs/$ENVIRONMENT.tfvars" \`
	`-var="project_name=$PROJECT_NAME" \`
	`-var="environment=$ENVIRONMENT" \`
	`-var="user_role=$AWS_USER_ROLE" \`
	`-var="lambda_temp_dir=$LAMBDA_FUNCTION_TEMP_DIR_TERRAFORM" \`
	`-var="layer_temp_dir=$LAMBDA_LAYER_TEMP_DIR_TERRAFORM" \`
	`-parallelism=40 \`
	`-out='tfplan.out'`
		AWS DevOps

Menyebarkan konfigurasi.	Untuk menerapkan konfigurasi, jalankan perintah berikut:
	`# Run terraform apply`
	`terraform apply tfplan.out`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui konfigurasi lain dan simpan output.	Untuk memperbarui kebijakan AWS KMS utama dan menyimpan output konfigura si yang diterapkan di bucket Amazon S3 artefak Terraform, jalankan perintah berikut:
	`# Update AWS Supply Chain KMS Key policy with the service's requirements`
	`chmod +x ../script s/update-asc-kms-p olicy.sh`
	`../scripts/update- asc-kms-policy.sh`
	`# Update AWS KMS Keys' policy with IAM roles`
	`chmod +x ../scripts/ update-kms-policy.sh`
	`../scripts/update-kms- policy.sh`
	`# Create terraform outputs file to be used as input variables`
	`terraform output -json > raw_output.json`
	`jq -r 'to_entries | map( if .value.type == "string" then "\(.key) = \"\(.value.value)\"" else`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"\(.key) = \(.value.value | tojson)" end) | .[]' raw_output.json > $REPO_NAME-outputs.tfvars</pre>
	<pre># Upload reformed outputs file to Amazon S3 terraform artifacts bucket (For retrieval from other repositories)
	aws s3 cp $REPO_NAME-outputs.tfvars s3://$S3_TERRAFORM_ARTIFACTS_BUCKET_NAME/$REPO_NAME-outputs.tfvars
	rm -f raw_output.json
	rm -f $REPO_NAME-outputs.tfvars</pre>	

(Opsi manual) Terapkan aliran integrasi Rantai Pasokan AWS layanan menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka direktori `terraform-deployment` tersebut.	Untuk pergi ke `terraform-deployment` direktoriASC-Integration-Flows , jalankan perintah berikut: <pre>cd ASC-Integration-Flows/terraform-deployment</pre>	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan konfigurasi backend dan penyedia Terraform.	Untuk menyiapkan konfigurasi backend dan penyedia Terraform, gunakan skrip berikut:
	`# Setup terraform backend and providers config if they don't exist`
	`chmod +x ../scripts/generate-terraform-config.sh`
	`../scripts/generate-terraform-config.sh`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hasilkan rencana penerapan.	Untuk menghasilkan rencana penyebaran, jalankan perintah berikut. Perintah ini menginisi alisasi lingkungan Terraform Anda, menggabungkan variabel konfigurasi dari `ASC-Datasets` konfigura si Terraform Anda yang ada, dan menghasilkan rencana penerapan.
	
	`# Run terraform init and validate`
	`terraform init`
	`terraform validate`
	
	`# Download and merge ASC DATASET tfvars`
	`chmod +x ../scripts/download-vars-through-s3.sh`
	`../scripts/download-vars-through-s3.sh $ASC_DATASET_VARS_REPO`
	
	`# Run terraform plan`
	`terraform plan \`
	`-var-file="tfInputs/$ENVIRONMENT.tfvars" \`
	`-var="project_name=$PROJECT_NAME" \`
	`-var="environment=$ENVIRONMENT" \`
	`-var="lambda_temp_dir=$LAMBDA_FUNCTION_TEMP_DIR_TERRAFORM" \`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`-var="layer_temp_dir=$LAMBDA_LAYER_TEMP_DIR_TERRAFORM" \ -parallelism=40 \ -out='tfplan.out'`	
Menyebarkan konfigurasi.	Untuk menerapkan konfigurasi, jalankan perintah berikut: `# Run terraform apply terraform apply tfplan.out`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui konfigurasi lainnya.	Untuk memperbarui kebijakan AWS KMS utama dan menyimpan output konfigura si yang diterapkan di bucket Amazon S3 artefak Terraform, jalankan perintah berikut:
	`# Update AWS KMS Keys' policy with IAM roles`
	`chmod +x ../scripts/update-kms-policy-through-s3.sh`
	`../scripts/update-kms-policy-through-s3.sh $ASC_DATA SET_VARS_REPO`
	`# Create terraform outputs file to be used as input variables`
	`terraform output -json > raw_output.json`
	`jq -r 'to_entries | map( if .value.type == "string" then "\(.key) = \"\(.value.value)\"" else "\(.key) = \(.value.value | tojson)" end ) | .[]' raw_output.json > $REPO_NAME-outputs.tfvars`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`# Upload reformed outputs file to Amazon S3 terraform artifacts bucket (For retrieval from other repositories)`
	`aws s3 cp $REPO_NAME-outputs.tfvars s3://$S3_TERRAFORM_ARTIFACTS_BUCKET_NAME/$REPO_NAME-outputs.tfvars`
	`rm -f raw_output.json`
	`rm -f $REPO_NAME-outputs.tfvars`	

(Kedua opsi) Mencerna data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah contoh file CSV.	Untuk mengunggah file CSV sampel untuk kumpulan data, gunakan langkah-langkah berikut: 1. Buat contoh file CSV dengan beragam data untuk kumpulan data Kalender dan Garis Pesanan Keluar yang dibuat dalam penerapan. 2. Ambil ID Rantai Pasokan AWS instance `asc_instance_id` dari direktori keluaran terraform. 3. Perhatikan nama bucket Amazon S3 untuk Rantai	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Pasokan AWS yang dibuat dalam penerapan: `aws-supply-chain-data-<Instance_ID>`
	4. Untuk mengunggah file dengan menggunakan AWS CLI, jalankan perintah berikut:
	`# Upload Calendar CSV file`
	`aws s3 cp calendar_sample.csv s3://aws-supply-chain-data-<Instance_ID>/calendar-data/`
	`# Upload Outbound Order Line CSV file`
	`aws s3 cp outbound_order_line_sample.csv s3://aws-supply-chain-data-<Instance_ID>/outbound-order-line-data/`	

(Kedua opsi) Siapkan Rantai Pasokan AWS akses

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur Rantai Pasokan AWS akses.	Untuk mengatur Rantai Pasokan AWS akses dari Konsol Manajemen AWS, gunakan langkah-langkah berikut:	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Masuk ke Konsol Manajemen AWS dan cari Rantai Pasokan AWS layanan. 2. Pergi ke instance `asc-deployment-poc-dev-asc-instance` . 3. Pola ini menggunakan IAM Identity Center untuk mengelola akses pengguna ke Rantai Pasokan AWS instance. Untuk memastikan akses lengkap ke solusi ini, masuk sebagai Administrator aplikasi.	

(Opsi otomatis) Bersihkan semua sumber daya menggunakan alur kerja GitHub Tindakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memicu alur kerja penghancuran untuk sumber daya alur integrasi.	Memicu alur kerja penghancuran `ASC-Integration-Flows` dari cabang penyebaran Anda di organisasi Anda GitHub .	AWS DevOps
Memicu alur kerja penghancuran untuk sumber daya kumpulan data.	Memicu alur kerja penghancuran `ASC-Datasets` dari cabang penyebaran Anda di organisasi Anda GitHub .	AWS DevOps

(Opsi manual) Bersihkan sumber daya aliran Rantai Pasokan AWS integrasi menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka direktori `terraform-deployment` tersebut.	Untuk pergi ke `terraform-deployment` direktori`ASC-Integration-Flows` , jalankan perintah berikut: `cd ASC-Integration-Flows/terraform-deployment`	AWS DevOps

Siapkan konfigurasi backend dan penyedia Terraform.	Untuk menyiapkan konfigurasi backend dan penyedia Terraform, gunakan skrip berikut:
	`# Setup terraform backend and providers config if they don't exist`
	`chmod +x ../scripts/generate-terraform-config.sh`
	`../scripts/generate-terraform-config.sh`
		AWS DevOps

Menghasilkan rencana penghancuran infrastruktur.	Untuk mempersiapkan penghancuran AWS infrastruktur Anda yang terkontrol dengan membuat rencana pembongkaran terperinci, jalankan perintah berikut. Proses ini menginisialisasi Terraform, menggabungkan konfigurasi Rantai Pasokan AWS kumpulan data, dan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	membuat rencana penghancuran yang dapat Anda tinjau sebelum dijalankan.
	
	`# Run terraform init and validate`
	`terraform init`
	`terraform validate`
	
	`# Download and merge ASC DATASET tfvars`
	`chmod +x ../scripts/download-vars-through-s3.sh`
	`../scripts/download-vars-through-s3.sh $ASC_DATASET_VARS_REPO`
	
	`# Run terraform plan`
	`terraform plan -destroy \`
	`-var-file="tfInputs/$ENVIRONMENT.tfvars" \`
	`-var="project_name=$PROJECT_NAME" \`
	`-var="environment=$ENVIRONMENT" \`
	`-var="lambda_temp_dir=$LAMBDA_FUNCTION_TEMP_DIR_TERRAFORM" \`
	`-var="layer_temp_dir=$LAMBDA_LAYER_TEMP_DIR_TERRAFORM" \`
	`-parallelism=40 \`
	`-out='tfplan.out'`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan rencana penghancuran infrastruktur.	Untuk menjalankan penghancuran infrastruktur yang direncanakan, jalankan perintah berikut:
	`# Run terraform apply`
	`terraform apply`
	`tfplan.out`
		AWS DevOps

Hapus output Terraform dari bucket Amazon S3.	Untuk menghapus file output yang diunggah selama penyebaran`ASC-Integration-Flows`, jalankan perintah berikut:
	`# Delete the outputs file`
	`aws s3 rm s3://$S3_TERRAFORM_ARTIFACTS_BUCKET_NAME/$REPO_NAME-outputs.tfvars`
		AWS DevOps

(Opsi manual) Bersihkan sumber daya kumpulan data Rantai Pasokan AWS layanan menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka direktori `terraform-deployment` tersebut.	Untuk pergi ke `terraform-deployment` direktori`ASC-Datasets`, jalankan perintah berikut: `cd ASC-Datasets/terraform-deployment`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan konfigurasi backend dan penyedia Terraform.	Untuk menyiapkan konfigurasi backend dan penyedia Terraform, gunakan skrip berikut:
	`# Setup terraform backend and providers config if they don't exist`
	`chmod +x ../scripts/generate-terraform-config.sh`
	`../scripts/generate-terraform-config.sh`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menghasilkan rencana penghancuran infrastruktur.	Untuk membuat rencana untuk menghancurkan sumber daya Rantai Pasokan AWS dataset, jalankan perintah berikut:
	`# Run terraform init and validate`
	`terraform init`
	`terraform validate`
	
	`# Run terraform plan`
	`terraform plan -destroy \`
	`-var-file="tfInputs/$ENVIRONMENT.tfvars" \`
	`-var="project_name=$PROJECT_NAME" \`
	`-var="environment=$ENVIRONMENT" \`
	`-var="user_role=$AWS_USER_ROLE" \`
	`-var="lambda_temp_dir=$LAMBDA_FUNCTION_TEMP_DIR_TERRAFORM" \`
	`-var="layer_temp_dir=$LAMBDA_LAYER_TEMP_DIR_TERRAFORM" \`
	`-parallelism=40 \`
	`-out='tfplan.out'`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ember Amazon S3 kosong.	Untuk mengosongkan semua bucket Amazon S3 (kecuali bucket logging akses server, yang dikonfigurasi untuk`force-destroy`), gunakan skrip berikut:
	`# Delete S3 buckets excluding server access logging bucket`
	`chmod +x ../scripts/empty-s3-buckets.sh`
	`../scripts/empty-s3-buckets.sh tfplan.out`
		AWS DevOps

Jalankan rencana penghancuran infrastruktur.	Untuk menjalankan penghancuran infrastruktur Rantai Pasokan AWS dataset Anda yang direncanakan menggunakan rencana yang dihasilkan, jalankan perintah berikut:
	`# Run terraform apply`
	`terraform apply tfplan.out`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hapus keluaran Terraform dari ember artefak Amazon S3 Terraform.	Untuk menyelesaikan proses pembersihan, hapus file output yang diunggah selama penerapan `ASC-Datasets` dengan menjalankan perintah berikut:
	`# Delete the outputs file`
	`aws s3 rm s3://$S3_TERRAFORM_ARTIFACTS_BUCKET_NAME/$REPO_NAME-outputs.tfvars`
		AWS DevOps

Pemecahan masalah

Isu	Solusi
Rantai Pasokan AWS Kumpulan data atau alur integrasi tidak diterapkan dengan benar karena kesalahan Rantai Pasokan AWS internal atau izin IAM yang tidak memadai untuk peran layanan.	Pertama, bersihkan semua sumber daya. Kemudian, terapkan ulang sumber daya Rantai Pasokan AWS kumpulan data dan kemudian gunakan kembali sumber daya aliran integrasi. Rantai Pasokan AWS
Alur Rantai Pasokan AWS integrasi tidak mengambil file data baru yang diunggah untuk kumpulan data. Rantai Pasokan AWS	1. Periksa apakah awalan konfigurasi alur Rantai Pasokan AWS integrasi cocok dengan awalan yang digunakan saat mengunggah file data sampel. 2. Jika sumber daya untuk Rantai Pasokan AWS kumpulan data dibuat ulang, Amazon Resource Names (ARNs) yang terkait akan berubah secara internal. Oleh karena itu, menyebarkan kembali sumber daya untuk

Isu	Solusi
	sumber daya aliran integrasi Rantai Pasokan AWS layanan.

Sumber daya terkait

AWS dokumentasi

- Rantai Pasokan AWS

Sumber daya lainnya

- Memahami alur kerja GitHub Tindakan (GitHub dokumentasi)

Informasi tambahan

Solusi ini dapat direplikasi untuk lebih banyak kumpulan data dan dapat ditanyakan untuk analisis lebih lanjut, melalui dasbor bawaan yang disediakan atau integrasi Rantai Pasokan AWS khusus dengan Amazon Quick Sight. Selain itu, Anda dapat menggunakan Amazon Q untuk mengajukan pertanyaan terkait Rantai Pasokan AWS instans Anda.

Menganalisis data dengan Rantai Pasokan AWS Analytics

Untuk petunjuk menyiapkan Rantai Pasokan AWS Analytics, lihat Menyetel Rantai Pasokan AWS Analytics di Rantai Pasokan AWS dokumentasi.

Pola ini menunjukkan pembuatan kumpulan data Kalender dan Outbound_Order_Line. Untuk membuat analisis yang menggunakan kumpulan data ini, gunakan langkah-langkah berikut:

1. Untuk menganalisis kumpulan data, gunakan dasbor Analisis Musim. Untuk menambahkan dasbor, ikuti langkah-langkah di dasbor Prebuilt dalam dokumentasi. Rantai Pasokan AWS
2. Pilih dasbor untuk melihat analisisnya yang didasarkan pada contoh file CSV untuk data Kalender dan data Outbound Order Line.

Dasbor memberikan wawasan tentang permintaan selama bertahun-tahun berdasarkan data yang dicerna untuk kumpulan data. Anda dapat lebih lanjut menentukan ProductID, CustomerID, tahun, dan parameter lain untuk analisis.

Gunakan Amazon Q untuk mengajukan pertanyaan terkait Rantai Pasokan AWS instans Anda

Amazon Q in Rantai Pasokan AWS adalah asisten AI generatif interaktif yang membantu Anda mengoperasikan rantai pasokan dengan lebih efisien. Amazon Q dapat melakukan hal berikut:

- Analisis data di danau Rantai Pasokan AWS data Anda.
- Memberikan wawasan operasional dan keuangan.
- Jawab pertanyaan rantai pasokan langsung Anda.

Untuk informasi selengkapnya tentang menggunakan Amazon Q, lihat Mengaktifkan Amazon Q di Rantai Pasokan AWS dan Menggunakan Amazon Q Rantai Pasokan AWS dalam Rantai Pasokan AWS dokumentasi.

Mengotomatiskan penilaian sumber daya AWS

Naveen Suthar, Arun Bagal, Manish Garg, dan Sandeep Gawande, Amazon Web Services

Ringkasan

Pola ini menjelaskan pendekatan otomatis untuk menyiapkan kemampuan penilaian sumber daya dengan menggunakan AWS Cloud Development Kit (AWS CDK). Dengan menggunakan pola ini, tim operasi mengumpulkan detail audit sumber daya secara otomatis dan melihat detail semua sumber daya yang digunakan di akun AWS pada satu dasbor. Ini sangat membantu dalam kasus penggunaan berikut:

- Mengidentifikasi infrastruktur sebagai alat kode (IAc) dan mengisolasi sumber daya yang dibuat oleh berbagai solusi IAc seperti HashiCorp Terraform, AWS, AWS CDK CloudFormation, dan AWS Command Line Interface (AWS CLI)
- Mengambil informasi audit sumber daya

Solusi ini juga akan membantu tim kepemimpinan memperoleh wawasan tentang sumber daya dan aktivitas dalam akun AWS dari satu dasbor.

Catatan: Amazon Quick Sight adalah layanan berbayar. Sebelum menjalankannya untuk menganalisis data dan membuat dasbor, tinjau harga Amazon Quick Sight.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Peran dan izin AWS Identity and Access Management (IAM) dengan akses ke sumber daya penyediaan
- Akun Amazon Quick Suite yang dibuat dengan akses ke Amazon Simple Storage Service (Amazon S3) dan Amazon Athena
- AWS CDK versi 2.55.1 atau yang lebih baru diinstal
- Python versi 3.9 atau yang lebih baru diinstal

Batasan

- Solusi ini diterapkan ke satu akun AWS.
- Solusinya tidak akan melacak peristiwa yang terjadi sebelum penerapannya kecuali AWS CloudTrail sudah menyiapkan dan menyimpan data dalam bucket S3.

Versi produk

- AWS CDK versi 2.55.1 atau yang lebih baru
- Python versi 3.9 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- Amazon Athena
- AWS CloudTrail
- AWS Glue
- AWS Lambda
- Penglihatan Cepat Amazon
- Amazon S3

Arsitektur target

Kode AWS CDK akan menerapkan semua sumber daya yang diperlukan untuk menyiapkan kemampuan penilaian sumber daya di akun AWS. Diagram berikut menunjukkan proses pengiriman CloudTrail log ke AWS Glue, Amazon Athena, dan Quick Sight.

1. CloudTrail mengirim log ke ember S3 untuk penyimpanan.
2. Pemberitahuan peristiwa memanggil fungsi Lambda yang memproses log dan menghasilkan data yang difilter.
3. Data yang difilter disimpan di bucket S3 lain.
4. Crawler AWS Glue disiapkan pada data yang difilter yang ada di bucket S3 untuk membuat skema di tabel AWS Glue Data Catalog.
5. Data yang difilter siap untuk ditanyakan oleh Amazon Athena.
6. Data yang ditanyakan diakses oleh Quick Sight untuk visualisasi.

Otomatisasi dan skala

- Solusi ini dapat diskalakan dari satu akun AWS ke beberapa akun AWS jika ada CloudTrail jejak di seluruh organisasi di AWS Organizations. Dengan menerapkan CloudTrail di tingkat organisasi, Anda juga dapat menggunakan solusi ini untuk mengambil detail audit sumber daya untuk semua sumber daya yang diperlukan.
- Pola ini menggunakan sumber daya tanpa server AWS untuk menerapkan solusi.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun AWS dan Wilayah AWS.
- AWS CloudTrail membantu Anda mengaudit tata kelola, kepatuhan, dan risiko operasional akun AWS Anda.
- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data. Pola ini menggunakan crawler AWS Glue dan tabel AWS Glue Data Catalog.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Quick Suite adalah layanan intelijen bisnis skala cloud (BI) yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda dalam satu dasbor.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Kode untuk pola ini tersedia di GitHub infrastructure-assessment-iac-automationrepositori.

Repositori kode berisi file dan folder berikut:

- `lib`folder — AWS CDK membangun file Python yang digunakan untuk membuat sumber daya AWS
- `src/lambda_code`— Kode Python yang dijalankan dalam fungsi Lambda
- `requirements.txt`— Daftar semua dependensi Python yang harus diinstal
- `cdk.json`— File input untuk memberikan nilai yang diperlukan untuk memutar sumber daya

Praktik terbaik

Siapkan pemantauan dan peringatan untuk fungsi Lambda. Untuk informasi selengkapnya, lihat Memantau dan memecahkan masalah fungsi Lambda. Untuk praktik terbaik umum saat bekerja dengan fungsi Lambda, lihat dokumentasi AWS.

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repo di mesin lokal Anda.	Untuk mengkloning repositori, jalankan perintah. `git clone https://github.com/aws-samples/infrastructure-assessment-iac-automation.git`	AWS DevOps, DevOps insinyur

Siapkan lingkungan virtual Python dan instal dependensi yang diperlukan.	Untuk mengatur lingkungan virtual Python, jalankan perintah berikut.
	`cd infrastructure-assessment-iac-automation`
	`python3 -m venv .venv`
	`source .venv/bin/activate`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk mengatur dependensi yang diperlukan, jalankan perintah. `pip install -r requirements.txt`	
Siapkan lingkungan AWS CDK dan sintesis kode AWS CDK.	1. Untuk mengatur lingkungan AWS CDK di akun AWS Anda, jalankan perintah `cdk bootstrap aws://ACCOUNT-NUMBER/REGION` . 2. Untuk mengonversi kode ke konfigurasi AWS CloudFormation stack, jalankan perintah `cdk synth`.	AWS DevOps, DevOps insinyur

Siapkan kredensi AWS di mesin lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ekspor variabel untuk akun dan Wilayah tempat tumpukan akan digunakan.	Untuk memberikan kredensi AWS untuk AWS CDK dengan menggunakan variabel lingkungan, jalankan perintah berikut.
	`export CDK_DEFAULT_ACCOUNT=<12 Digit AWS Account Number>`
	`export CDK_DEFAULT_REGION=<region>`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan profil AWS CLI.	Untuk menyiapkan profil AWS CLI untuk akun, ikuti petunjuk dalam dokumentasi AWS.	AWS DevOps, DevOps insinyur

Konfigurasikan dan terapkan alat penilaian sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan sumber daya di akun.	Untuk menerapkan sumber daya di akun AWS dengan menggunakan AWS CDK, lakukan hal berikut: 1. Di root repositori kloning, dalam `cdk.json` file, berikan input untuk parameter berikut: • `s3_context` • `ct_context` • `kms_context` • `lambda_context` • `glue_context` • `qs_context` Nilai-nilai ini menentukan konfigurasi sumber daya dan nomenklatur. Nilai default diatur dan dapat diubah jika diperlukan. **Note** Untuk menghinda ri kesalahan yang	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengatakan bahwa bucket S3 sudah ada, pastikan untuk memberikan nama unik untuk `s3_context` di `output` bagian `ct` dan. 2. Untuk menyebarkan sumber daya, jalankan perintah`cdk deploy`. `cdk deploy`Perintah membuat CloudTrail sumber daya untuk mencatat peristiwa dan menyimpan file log di bucket input S3. File log jejak akan diproses oleh fungsi Lambda. Hasil yang difilter disimpan dalam bucket output S3 dan siap dikonsumsi oleh Amazon Athena dan Amazon QuickSight.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan crawler AWS Glue dan buat tabel Katalog Data.	Crawler AWS Glue digunakan untuk menjaga skema data tetap dinamis. Solusi ini membuat dan memperbarui partisi dalam tabel AWS Glue Data Catalog dengan menjalankan crawler secara berkala seperti yang ditentukan oleh penjadwal crawler AWS Glue. Setelah data tersedia di bucket keluaran S3, gunakan langkah-langkah berikut untuk menjalankan crawler AWS Glue dan buat skema tabel Katalog Data untuk pengujian: 1. Masuk ke AWS Management Console dan navigasikan ke konsol AWS Glue. 2. Di panel navigasi, di bawah Katalog Data, pilih Crawler. 3. Pilih `iac-tool-qa-resource-iac-json-crawler` crawler. 4. Jalankan crawler. 5. Setelah crawler berjalan dengan sukses, crawler akan membuat tabel AWS Glue Data Catalog. Quick Sight akan menggunakan tabel untuk memvisualisasikan data.	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Kode AWS CDK mengonfigurasi crawler AWS Glue untuk berjalan pada waktu tertentu, tetapi Anda juga dapat menjalankannya sesuai permintaan.	
Menyebarkan konstruksi Quick Sight.	1. Untuk menerapkan konstruksi Quick Sight, hapus komentar kode antara `#QuickSight setup - start` dan di. `#QuickSight setup – ends resource_ iac_tool_stack.py` 2. Setelah Anda menghapus komentar, jalankan `cdk deploy` perintah untuk membuat `QuickSigh t DataSource` dan `QuickSight DataSet` di akun Quick Sight.	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dasbor Quick Sight.	Untuk membuat contoh dasbor dan analisis Quick Sight, lakukan hal berikut: 1. Arahkan ke konsol Quick Sight dan pilih Wilayah AWS tempat sumber daya digunakan. 2. Di panel navigasi, pilih Datasets, dan validasi bahwa kumpulan data bernama `ct-operations-iac-ds` telah dibuat dalam kumpulan data Quick Sight. Jika Anda tidak melihat kumpulan data, gunakan kembali konstruksi Quick Sight. 3. Pilih `ct-operations-iac-ds` dataset, dan pilih GUNAKAN DALAM ANALISIS. 4. Pilih lembar default. 5. Pilih kolom masing-masing dari daftar bidang di sisi kiri. 6. Setelah memilih kolom yang diperlukan, pilih jenis visual yang sesuai untuk melihat data. Untuk informasi selengkap nya, lihat Memulai analisis di	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon Quick Sight dan tipe Visual di Amazon Quick Sight.	

Bersihkan semua sumber daya AWS dalam solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya AWS.	1. Untuk menghapus sumber daya AWS yang digunakan oleh solusi, jalankan perintah`cdk destroy`. 2. Hapus semua objek dari dua ember S3, lalu lepaskan ember. Untuk informasi selengkap nya, lihat Menghapus bucket.	AWS DevOps, DevOps insinyur

Siapkan fitur tambahan di atas otomatisasi alat penilaian sumber daya AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pantau dan bersihkan sumber daya yang dibuat secara manual.	(Opsional) Jika organisasi Anda memiliki persyaratan kepatuhan untuk membuat sumber daya menggunak an alat IAC, Anda dapat mencapai kepatuhan dengan menggunakan otomatisa si alat penilaian sumber daya AWS untuk mengambil sumber daya yang disediaka n secara manual. Anda juga	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dapat menggunakan alat ini untuk mengimpor sumber daya ke alat IAc atau untuk membuatnya kembali. Untuk memantau sumber daya yang disediakan secara manual, lakukan tugas tingkat tinggi berikut: 1. Terapkan otomatisasi alat penilaian sumber daya AWS. 2. Siapkan fungsi Lambda untuk menanyakan tabel Athena setiap hari, menemukan data yang relevan tentang sumber daya yang disediaka n secara manual, dan mengekspornya ke file nilai yang dipisahkan koma (CSV). 3. Setelah fungsi Lambda berjalan, pemberita huan dengan data yang diperlukan dapat dikirim ke masing-masing pemangku kepentingan. 4. Untuk retensi yang lebih lama, file.csv dapat disimpan di bucket S3. 5. Berdasarkan informasi dalam file.csv, hapus sumber daya yang dibuat	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	secara manual atau impor ke solusi IAc yang ada.	

Pemecahan masalah

Isu	Solusi
AWS CDK mengembalikan kesalahan.	Untuk bantuan terkait masalah AWS CDK, lihat Memecahkan masalah umum AWS CDK.

Sumber daya terkait

- Membangun fungsi Lambda dengan Python
- Memulai AWS CDK
- Bekerja dengan AWS CDK dengan Python
- Membuat jejak CloudTrail log
- Memulai dengan Amazon Quick Sight

Informasi tambahan

Beberapa akun

Untuk menyiapkan kredensi AWS CLI untuk beberapa akun, gunakan profil AWS. Untuk informasi selengkapnya, lihat bagian Mengonfigurasi beberapa profil di Mengatur AWS CLI.

Perintah AWS CDK

Saat bekerja dengan AWS CDK, ingatlah perintah berguna berikut:

- Daftar semua tumpukan di aplikasi

```
cdk ls
```

- Memancarkan template AWS yang disintesis CloudFormation

```
cdk synth
```

- Menerapkan tumpukan ke akun AWS default dan Wilayah

```
cdk deploy
```

- Membandingkan tumpukan yang diterapkan dengan status saat ini

```
cdk diff
```

- Membuka dokumentasi AWS CDK

```
cdk docs
```

Instal sistem SAP secara otomatis dengan menggunakan alat sumber terbuka

Guilherme Sesterheim, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara mengotomatiskan instalasi sistem SAP dengan menggunakan alat sumber terbuka untuk membuat sumber daya berikut:

- Database SAP S/4HANA 1909
- Contoh SAP ABAP Central Services (ASCS)
- Contoh SAP Primary Application Server (PAS)

HashiCorp Terraform menciptakan infrastruktur sistem SAP dan Ansible mengkonfigurasi sistem operasi (OS) dan menginstal aplikasi SAP. Jenkins menjalankan instalasi.

Pengaturan ini mengubah instalasi sistem SAP menjadi proses berulang, yang dapat membantu meningkatkan efisiensi dan kualitas penerapan.

> **Note**
>
> Kode contoh yang disediakan dalam pola ini berfungsi untuk sistem ketersediaan tinggi (HA) dan sistem non-HA.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Bucket Amazon Simple Storage Service (Amazon S3) yang berisi semua file media SAP Anda
- Prinsipal AWS Identity and Access Management (IAM) dengan kunci akses dan kunci rahasia, dan yang memiliki izin berikut:
  - Izin baca saja: Amazon Route 53, AWS Key Management Service (AWS KMS)
  - Izin baca dan tulis: Amazon S3, Amazon Elastic Compute Cloud (Amazon), EC2 Amazon Elastic File System (Amazon EFS), IAM, Amazon, Amazon DynamoDB CloudWatch

- Zona host pribadi Route 53
- Berlangganan Red Hat Enterprise Linux untuk SAP dengan HA dan Update Services 8.2 Amazon Machine Image (AMI) di Amazon Marketplace
- Kunci yang dikelola pelanggan AWS KMS
- Sebuah key pair Secure Shell (SSH)
- Grup EC2 keamanan Amazon yang memungkinkan koneksi SSH pada port 22 dari nama host tempat Anda menginstal Jenkins (nama host kemungkinan besar localhost)
- Vagrant dengan HashiCorp diinstal dan dikonfigurasi
- VirtualBoxoleh Oracle diinstal dan dikonfigurasi
- Keakraban dengan Git, Terraform, Ansible, dan Jenkins

Batasan

- Hanya SAP S/4HANA 1909 yang sepenuhnya diuji untuk skenario khusus ini. Contoh kode Ansible dalam pola ini memerlukan modifikasi jika Anda menggunakan versi lain dari SAP HANA.
- Contoh prosedur dalam pola ini berfungsi untuk sistem operasi Mac OS dan Linux. Beberapa perintah hanya dapat dijalankan di terminal berbasis Unix. Namun, Anda dapat mencapai hasil yang serupa dengan menggunakan perintah yang berbeda dan OS Windows.

Versi produk

- GETAH S/4HANA 1909
- Red Hat Enterprise Linux (RHEL) 8.2 atau versi yang lebih tinggi

Arsitektur

Diagram berikut menunjukkan contoh alur kerja yang menggunakan alat sumber terbuka untuk mengotomatiskan instalasi sistem SAP di akun AWS:

Diagram menunjukkan alur kerja berikut:

1. Jenkins mengatur menjalankan instalasi sistem SAP dengan menjalankan kode Terraform dan Ansible.
2. Kode Terraform membangun infrastruktur sistem SAP.

3. Kode Ansible mengkonfigurasi OS dan menginstal aplikasi SAP.
4. Database SAP S/4HANA 1909, instans ASCS, dan instans PAS yang mencakup semua prasyarat yang ditentukan diinstal pada instans Amazon. EC2

> **Note**
>
> Contoh penyiapan dalam pola ini secara otomatis membuat bucket Amazon S3 di akun AWS Anda untuk menyimpan file status Terraform.

Tumpukan teknologi

- Terraform
- Ansible
- Jenkins
- Database SAP S/4HANA 1909
- Sebuah contoh SAP ASCS
- Sebuah contoh SAP PAS
- Amazon EC2

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan, dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk melindungi data Anda.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang

akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Alat-alat lainnya

- HashiCorp Terraform adalah aplikasi antarmuka baris perintah yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- Ansible adalah alat konfigurasi sumber terbuka sebagai kode (CaC) yang membantu mengotomatiskan aplikasi, konfigurasi, dan infrastruktur TI.
- Jenkins adalah server otomatisasi sumber terbuka yang memungkinkan pengembang untuk membangun, menguji, dan menyebarkan perangkat lunak mereka.

Kode

Kode untuk pola ini tersedia di repositori GitHub aws-install-sap-with-jenkins-ansible.

Epik

Konfigurasikan prasyarat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan file media SAP Anda ke bucket Amazon S3.	Buat bucket Amazon S3 yang berisi semua file media SAP Anda. **Important** Pastikan Anda mengikuti hierarki folder AWS Launch Wizard untuk S/4HANA dalam dokumentasi Launch Wizard.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal VirtualBox.	Instal dan konfigurasikan VirtualBoxoleh Oracle.	DevOps insinyur
Instal Vagrant.	Instal dan konfigurasikan Vagrant oleh HashiCorp.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan akun AWS Anda.	1. Verifikasi bahwa Anda memiliki prinsipal IAM dengan kunci akses dan kunci rahasia, dan yang memiliki izin berikut: • Izin baca saja: Amazon Route 53, AWS Key Management Service (AWS KMS) • Izin baca dan tulis: Amazon S3, Amazon Elastic Compute Cloud (Amazon), EC2 Amazon Elastic File System (Amazon EFS), IAM, Amazon, Amazon DynamoDB CloudWatch 2. Simpan kunci akses dan kunci rahasia kepala IAM untuk referensi nanti. 3. Buat zona host pribadi Route 53, jika Anda belum memilikinya. Simpan nama zona (misalnya, sapteam.net) untuk referensi nanti. 4. Berlangganan Red Hat Enterprise Linux untuk SAP dengan HA dan Update Services 8.2 AMI di Amazon Marketplace. Simpan ID AMI (misalnya, ami-0000000) untuk referensi nanti.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Buat kunci terkelola pelanggan AWS KMS. Simpan Amazon Resource Name (ARN) kunci KMS untuk referensi nanti. **Note** Berikut ini adalah contoh ARN kunci terkelola pelanggan AWS KMS: arn:aws:kms:us-east- 1:123412341234:key/uuid 6. Buat key pair SSH. Simpan nama pasangan kunci dan file.pem untuk referensi nanti. 7. Buat grup EC2 keamanan Amazon yang memungkinkan koneksi SSH pada port 22 dari nama host tempat Anda menginstal Jenkins. Simpan ID grup keamanan untuk referensi nanti. **Note** Nama host kemungkinan besar localhost.	

Membangun dan menjalankan instalasi SAP Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode dari. GitHub	Kloning repositori aws-install-sap-with-jenkins-ansible pada. GitHub	DevOps insinyur
Mulai layanan Jenkins.	Buka terminal Linux. Kemudian, arahkan ke folder lokal yang berisi folder repositori kode kloning dan jalankan perintah berikut: `sudo vagrant up` **Note** Startup Jenkins memakan waktu sekitar 20 menit. Perintah mengembal ikan Layanan aktif dan menjalankan pesan saat berhasil.	DevOps insinyur
Buka Jenkins di browser web dan masuk.	1. Di browser web, masukkan http://localhost:5555. Jenkins terbuka. 2. Masuk ke Jenkins dengan menggunakan admin untuk nama pengguna dan my_secret_pass_from_vault untuk kata sandi.	DevOps insinyur
Konfigurasikan parameter instalasi sistem SAP Anda.	1. Di Jenkins, pilih Kelola Jenkins. Kemudian, pilih	Administrator sistem AWS, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Manage Credentials. Daftar variabel kredensyal yang dapat Anda konfigurasikan muncul. 2. Konfigurasikan semua variabel kredensi berikut: • Untuk AWS_ACCOUNT_CREDENTIALS, masukkan ID kunci akses dan ID kunci akses rahasia kepala IAM Anda. • Untuk AMI_ID, masukkan Red Hat Enterprise Linux untuk SAP dengan HA dan Update Services 8.2 AMI ID AMI. • Untuk KMS_KEY_ARN, masukkan ARN kunci terkelola pelanggan AWS KMS Anda. • Untuk SSH_KEYPAIR_NAME, masukkan nama key pair SSH Anda, tanpa memasukkan tipe file.pem. • Untuk SSH_KEYPAIR_FILE, masukkan nama lengkap file.pem pasangan kunci Anda (misalnya, mykeypair.pem). Pastikan Anda juga mengunggah file.pem pasangan kunci ke Jenkins.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk S3_ROOT_F OLDER_INSTALL_FILE S, masukkan nama ember Amazon S3—dan folder, jika berlaku— (misalnya, s3:///S4H1909) yang berisi file media SAP Anda. my-media-bucket • Untuk PRIVATE_D NS_ZONE_NAME, masukkan nama zona host pribadi Route 53 Anda (misalnya, myprivate companyurl.net). • Untuk VPC_ID, masukkan ID VPC (misalnya, vpc-12345) dari VPC Amazon tempat Anda membuat sumber daya SAP. • Untuk SUBNET_IDS, masukkan dua subnet publik IDs jika Anda bekerja di lingkungan pengujian (untuk kemampuan HA masa depan). Jika Anda bekerja di lingkungan produksi, itu adalah praktik terbaik untuk menggunakan dua subnet pribadi dengan host benteng. • Untuk SECURITY_ GROUP_ID, masukkan ID grup EC2 keamanan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon yang memungkinkan koneksi SSH pada port 22 dari nama host tempat Anda menginstal Jenkins. **Note** Anda dapat mengonfigurasi parameter lain yang tidak diperlukan sesuai kebutuhan, berdasarkan kasus penggunaan Anda. Misalnya, Anda dapat mengubah ID sistem SAP (SID) dari instance, kata sandi default, nama, dan tag untuk sistem SAP Anda. Semua variabel yang diperlukan memiliki (Diperlukan) di awal nama mereka.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan instalasi sistem SAP Anda.	1. Di Jenkins, pilih Jenkins Home. Kemudian, pilih SAP Hana+ASCS+PAS 3 Instances. 2. Pilih Spin up dan install. Kemudian, pilih Main. 3. Pilih Build sekarang. Untuk informasi tentang langkah-langkah pipeline, lihat bagian Memahami langkah pipeline dari Mengotomatiskan instalasi SAP dengan alat sumber terbuka di Blog AWS. **Note** Jika terjadi kesalahan, gerakkan kursor Anda ke kotak kesalahan merah yang muncul dan pilih Log. Log untuk langkah pipa yang salah muncul. Sebagian besar kesalahan terjadi karena pengaturan parameter yang salah.	DevOps insinyur, administrator sistem AWS

Sumber daya terkait

- DevOps untuk Instalasi SAP - SAP: Dari 2 Bulan hingga 2 Jam (Perpustakaan Video DevOps Enterprise Summit)

Mengotomatiskan portofolio AWS Service Catalog dan penerapan produk dengan menggunakan AWS CDK

Sandeep Gawande, Viyoma Sachdeva, dan RAJNEESH TYAGI, Amazon Web Services

Ringkasan

AWS Service Catalog membantu Anda mengelola katalog layanan TI, atau produk secara terpusat, yang disetujui untuk digunakan di lingkungan AWS organisasi Anda. Kumpulan produk disebut portofolio, dan portofolio juga berisi informasi konfigurasi. Dengan AWS Service Catalog, Anda dapat membuat portofolio yang disesuaikan untuk setiap jenis pengguna di organisasi Anda dan kemudian memberikan akses ke portofolio yang sesuai. Pengguna tersebut kemudian dapat dengan cepat menyebarkan produk apa pun yang mereka butuhkan dari dalam portofolio.

Jika Anda memiliki infrastruktur jaringan yang kompleks, seperti arsitektur Multi-region dan multi-akun, Anda disarankan untuk membuat dan mengelola portofolio Service Catalog dalam satu akun sentral. Pola ini menjelaskan cara menggunakan AWS Cloud Development Kit (AWS CDK) untuk mengotomatiskan pembuatan portofolio Service Catalog di akun pusat, memberikan akses kepada pengguna akhir, dan kemudian, secara opsional, menyediakan produk dalam satu atau beberapa akun AWS target. ready-to-useSolusi ini membuat portofolio Service Catalog di akun sumber. Ini juga, secara opsional, menyediakan produk dalam akun target dengan menggunakan CloudFormation tumpukan AWS dan membantu Anda mengonfigurasi TagOptions produk:

- AWS CloudFormation StackSets — Anda dapat menggunakannya StackSets untuk meluncurkan produk Service Catalog di beberapa Wilayah dan akun AWS. Dalam solusi ini, Anda memiliki opsi untuk menyediakan produk secara otomatis saat Anda menerapkan solusi ini. Untuk informasi selengkapnya, lihat Menggunakan AWS CloudFormation StackSets (Dokumentasi Service Catalog) dan StackSets konsep (CloudFormation dokumentasi).
- TagOption library — Anda dapat mengelola tag pada produk yang disediakan dengan menggunakan TagOption library. A TagOptionadalah pasangan nilai kunci yang dikelola di AWS Service Catalog. Ini bukan tag AWS, tetapi berfungsi sebagai templat untuk membuat tag AWS berdasarkan tag TagOption. Untuk informasi selengkapnya, lihat TagOption library (Dokumentasi Service Catalog).

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif yang ingin Anda gunakan sebagai akun sumber untuk mengelola portofolio Service Catalog.
- Jika Anda menggunakan solusi ini untuk menyediakan produk dalam satu atau beberapa akun target, akun target harus sudah ada dan aktif.
- Izin AWS Identity and Access Management (IAM) untuk mengakses AWS Service Catalog, AWS CloudFormation, dan AWS IAM.

Versi produk

- AWS CDK versi 2.27.0

Arsitektur

Tumpukan teknologi target

- Portofolio Service Catalog di akun AWS terpusat
- Produk Service Catalog dikerahkan di akun target

Arsitektur target

1. Di akun portofolio (atau sumber), Anda memperbarui file config.json dengan akun AWS, Wilayah AWS, peran IAM, portofolio, dan informasi produk untuk kasus penggunaan Anda.
2. Anda menerapkan aplikasi AWS CDK.
3. Aplikasi AWS CDK mengasumsikan peran IAM penerapan dan membuat portofolio dan produk Service Catalog yang ditentukan dalam file config.json.

   Jika Anda mengonfigurasi StackSets untuk menyebarkan produk di akun target, proses akan berlanjut. Jika Anda tidak mengonfigurasi StackSets untuk menyediakan produk apa pun, maka prosesnya selesai.
4. Aplikasi AWS CDK mengasumsikan peran StackSet administrator dan menerapkan set CloudFormation tumpukan AWS yang Anda tentukan dalam file config.json.
5. Dalam akun target, StackSets mengasumsikan peran StackSet eksekusi dan ketentuan produk.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS CDK Toolkit adalah kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan aplikasi AWS CDK Anda.
- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Service Catalog membantu Anda mengelola katalog layanan TI secara terpusat yang disetujui untuk AWS. Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.

Repositori kode

Kode untuk pola ini tersedia di GitHub, di aws-cdk-servicecatalog-automationrepositori. Repositori kode berisi file dan folder berikut:

- cdk-sevicecatalog-app— Folder ini berisi aplikasi AWS CDK untuk solusi ini.
- config - Folder ini berisi file config.json dan CloudFormation template untuk menyebarkan produk dalam portofolio Service Catalog.
- config/config.json - File ini berisi semua informasi konfigurasi. Anda memperbarui file ini untuk menyesuaikan solusi ini untuk kasus penggunaan Anda.
- config/templates - Folder ini berisi CloudFormation template untuk produk Service Center.
- setup.sh - Skrip ini menyebarkan solusi.
- uninstall.sh - Skrip ini menghapus tumpukan dan semua sumber daya AWS yang dibuat saat menerapkan solusi ini.

Untuk menggunakan kode sampel, ikuti instruksi di bagian Epik.

Praktik terbaik

- Peran IAM yang digunakan untuk menerapkan solusi ini harus mematuhi prinsip hak istimewa paling sedikit (dokumentasi IAM).
- Patuhi praktik terbaik untuk mengembangkan aplikasi cloud dengan AWS CDK (postingan blog AWS).
- Patuhi praktik CloudFormation terbaik AWS (CloudFormation dokumentasi).

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal AWS CDK Toolkit.	Pastikan Anda telah menginstal AWS CDK Toolkit. Masukkan perintah berikut untuk mengonfirmasi apakah sudah diinstal dan periksa versinya.
	`cdk --version`
	Jika AWS CDK Toolkit tidak diinstal, maka masukkan perintah berikut untuk menginstalnya.
	`npm install -g aws-cdk@2.27.0`
	Jika versi AWS CDK Toolkit lebih awal dari 2.27.0, maka masukkan perintah berikut untuk memperbaruinya ke versi 2.27.0.
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`npm install -g aws-cdk@2.27.0 --force`	

Kloning repositori.	Masukkan perintah berikut. Di Kloning repositori di bagian Informasi tambahan, Anda dapat menyalin perintah lengkap yang berisi URL untuk repositori. Ini mengkloning aws-cdk-servicecatalog-automationrepositori dari. GitHub
	`git clone <repository-URL>.git`
	Ini membuat `cd aws-cdk-servicecatalog-automation` folder di direktori target. Masukkan perintah berikut untuk menavigasi ke folder ini.
	`cd aws-cdk-servicecatalog-automation`
		AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan kredensi AWS.	Masukkan perintah berikut. Ini mengekspor variabel berikut, yang menentukan akun AWS dan Wilayah tempat Anda menerapkan tumpukan.
	`export CDK_DEFAU LT_ACCOUNT=<12-digit AWS account number>`
	`export CDK_DEFAU LT_REGION=<AWS Region>`
	Kredensi AWS untuk AWS CDK disediakan melalui variabel lingkungan.
		AWS DevOps, DevOps insinyur

Konfigurasikan izin untuk peran IAM pengguna akhir.	Jika Anda akan menggunakan peran IAM untuk memberikan akses ke portofolio dan produk di dalamnya, peran harus memiliki izin yang akan diambil oleh prinsipal layanan servicecatalog.amazonaws.com. Untuk petunjuk tentang cara memberikan izin ini, lihat Mengaktifkan akses tepercaya dengan Service Catalog (dokumentasi AWS Organizations).	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan peran IAM yang dibutuhkan oleh StackSets.	Jika Anda menggunakan StackSets untuk secara otomatis menyediakan produk di akun target, Anda perlu mengonfigurasi peran IAM yang mengelola dan menjalankan kumpulan tumpukan. 1. Di akun sumber, konfirmasikan apakah `AWSCloudFormationStackSetAdministrationRole` sudah ada. Di akun target, konfirmasikan apakah `AWSCloudFormationStackSetExecutionRole` sudah ada. Jika peran ini sudah ada, Anda dapat melompat ke epik berikutnya. 2. Ikuti petunjuk dalam Berikan izin yang dikelola sendiri (dokumentasi IAM) untuk membuat peran administrasi set tumpukan di akun portofolio dan membuat peran eksekusi di setiap akun target.	AWS DevOps, DevOps insinyur

Sesuaikan dan terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation template.	Di `config/templates` folder, buat CloudFormation template untuk produk apa pun yang ingin Anda sertakan dalam portofolio Anda. Untuk informasi selengkapnya, lihat Bekerja dengan CloudFormation templat AWS (CloudFormation dokumentasi).	Pengembang aplikasi, AWS DevOps, DevOps insinyur
Sesuaikan file konfigurasi.	Di `config` folder, buka file config.json dan tentukan parameter yang sesuai untuk kasus penggunaan Anda. Parameter berikut diperlukan kecuali dinyatakan lain: Di `portfolios` bagian ini, tentukan parameter berikut untuk membuat satu atau beberapa portofolio Service Catalog: • `portfolioName` — Nama portofolio. • `providerName` Nama orang, tim, atau organisasi yang mengelola portofolio. • `description` — Penjelasan singkat tentang portofolio. • `roles`— (Opsional) Nama dari setiap peran IAM yang	Pengembang aplikasi, DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	harus memiliki akses ke portofolio ini. Pengguna yang memiliki peran ini dapat mengakses produk dalam portofolio ini. • `users`— (Opsional) Nama setiap pengguna IAM yang harus memiliki akses ke portofolio ini dan produknya. • `groups`— (Opsional) Nama setiap grup pengguna IAM yang harus memiliki akses ke portofolio ini dan produknya. ⚠ Warning Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Important
	`roles`,`users`, dan `groups` semua parameter opsional, tetapi jika Anda tidak menentukan salah satu parameter ini, maka tidak ada yang dapat melihat produk portofolio di konsol Service Catalog. Tentukan setidakny a satu dari parameter ini. Untuk informasi selengkapnya, lihat Memberikan izin kepada pengguna akhir Service Catalog (Dokumentasi Service Catalog).
	
	• (Opsional) Di `tagOption` bagian ini, tentukan TagOptions untuk produk:
	  • `key`— Nama TagOption kuncinya
	  • `value`— Nilai string yang diizinkan untuk TagOption
	
	Untuk informasi selengkap nya, lihat TagOption library (Dokumentasi Service Catalog).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Di **`products`**bagian ini, tentukan parameter berikut untuk produk:
	  • `portfolioName` — Nama portofolio tempat Anda ingin menambahkan produk. Anda hanya dapat menentukan satu portofolio.
	  • `productName` — Nama produk.
	  • `owner`— Pemilik produk.
	  • `productVersionName` — Nama versi produk dalam nilai string, sepertiv1.
	  • `templatePath` — Jalur file untuk CloudFormation template untuk produk.
	  • `deployWithStackSets` — (Opsional) Tentukan satu atau beberapa akun dan Wilayah tempat Anda ingin menggunakan StackSets untuk secara otomatis menyediakan produk dalam portofolio. Jika Anda menggunakan opsi penerapan ini, semua parameter berikut di bagian ini diperlukan:
	  • `accounts`— Akun target.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `regions`— Daerah sasaran. • `stackSetAdministrationRoleName` — Nama peran IAM yang digunakan untuk mengelola konfigurasi. StackSets Jangan ubah nilai ini. Peran ini harus memiliki nama yang tepat ini. • `stackSetExecutionRoleName` — Nama peran IAM di akun target yang menyebarkan instance tumpukan. Jangan ubah nilai ini. Peran ini harus memiliki nama yang tepat ini. Untuk contoh file konfigurasi yang telah selesai, lihat Contoh file konfigurasi di bagian Informasi tambahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan solusinya.	Masukkan perintah berikut. Ini menerapkan aplikasi AWS CDK dan menyediakan portofolio dan produk Service Catalog sebagaimana ditentukan dalam file config.json. `sh +x setup.sh`	Pengembang aplikasi, DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi penyebaran.	Verifikasi penerapan yang berhasil dengan melakukan hal berikut: 1. Masuk ke AWS Management Console dengan kredensyal yang dapat mengakses satu atau beberapa portofolio yang Anda tetapkan dalam file konfigurasi. 2. Buka konsol Service Catalog di https://console.aws.amazon.com/servicecatalog/. 3. Di panel navigasi, di bawah Penyediaan, pilih Produk. Verifikasi bahwa Anda melihat daftar produk yang Anda tentukan untuk portofolio. 4. Ikuti petunjuk dalam Meluncurkan produk (dokumentasi Service Catalog) untuk meluncurkan salah satu produk yang tersedia. Konfirmasikan bahwa versi dan tag produk yang tersedia cocok dengan nilai yang Anda berikan di file konfigurasi. 5. Jika Anda memilih untuk secara otomatis menyediakan produk dalam satu	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	atau beberapa akun target dengan menggunakan StackSets, lakukan hal berikut: a. Masuk dengan kredensyal yang memberi Anda izin untuk melihat produk yang disediakan di salah satu akun target. b. Di konsol Service Catalog, di panel navigasi, di bawah Penyediaan, pilih Produk yang disediakan. c. Konfirmasikan bahwa produk yang diharapkan muncul dalam daftar.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Perbarui portofolio dan produk.	Jika Anda ingin menggunakan solusi ini untuk memperbarui portofolio atau produk atau untuk menyediakan produk baru: 1. Buat perubahan yang diperlukan dalam file config.json. 2. Tambahkan atau ubah CloudFormation templat apa pun sesuai kebutuhan di `config/template` folder. 3. Menerapkan kembali solusinya. Misalnya, Anda dapat menambahkan portofolio tambahan atau menyediakan lebih banyak sumber daya. Aplikasi AWS CDK hanya mengimplementasikan perubahan. Jika tidak ada perubahan pada portofolio atau produk yang digunakan sebelumnya, pemindahan tidak akan memengaruhi portofolio atau produk tersebut.	Pengembang aplikasi, DevOps insinyur, AWS Umum

Bersihkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Hapus sumber daya AWS yang digunakan oleh solusi ini.	Jika Anda ingin menghapus produk yang disediakan, ikuti petunjuk di Menghapus produk yang disediakan (dokumentasi Service Catalog). Jika Anda ingin menghapus semua sumber daya yang dibuat oleh solusi ini, masukkan perintah berikut. `sh uninstall.sh`	AWS DevOps, DevOps insinyur, Pengembang aplikasi

Sumber daya terkait

- Perpustakaan Konstruksi AWS Service Catalog (AWS API Referensi)
- StackSets konsep (CloudFormation dokumentasi)
- AWS Service Catalog (AWS marketing)
- Menggunakan Service Catalog dengan AWS CDK (AWS workshop)

Informasi tambahan

Kloning repositori

Masukkan perintah berikut untuk mengkloning repositori dari. GitHub

```
git clone https://github.com/aws-samples/aws-cdk-servicecatalog-automation.git
```

Contoh file konfigurasi

Berikut ini adalah contoh file config.json dengan nilai contoh.

```
{
```

```
    "portfolios": [
        {
            "displayName": "EC2 Product Portfolio",
            "providerName": "User1",
            "description": "Test1",
            "roles": [
                "<Names of IAM roles that can access the products>"
            ],
            "users": [
                "<Names of IAM users who can access the products>"
            ],
            "groups": [
                "<Names of IAM user groups that can access the products>"
            ]
        },
        {
            "displayName": "Autoscaling Product Portfolio",
            "providerName": "User2",
            "description": "Test2",
            "roles": [
                "<Name of IAM role>"
            ]
        }
    ],
    "tagOption": [
        {
            "key": "Group",
            "value": [
                "finance",
                "engineering",
                "marketing",
                "research"
            ]
        },
        {
            "key": "CostCenter",
            "value": [
                "01",
                "02",
                "03",
                "04"
            ]
        },
        {
```

```
            "key": "Environment",
            "value": [
                "dev",
                "prod",
                "stage"
            ]
        }
    ],
    "products": [
        {
            "portfolioName": "EC2 Product Profile",
            "productName": "Ec2",
            "owner": "owner1",
            "productVersionName": "v1",
            "templatePath": "../../config/templates/template1.json"
        },
        {
            "portfolioName": "Autoscaling Product Profile",
            "productName": "autoscaling",
            "owner": "owner1",
            "productVersionName": "v1",
            "templatePath": "../../config/templates/template2.json",
            "deployWithStackSets": {
                "accounts": [
                    "012345678901",
                ],
                "regions": [
                    "us-west-2"
                ],
                "stackSetAdministrationRoleName":
 "AWSCloudFormationStackSetAdministrationRole",
                "stackSetExecutionRoleName": "AWSCloudFormationStackSetExecutionRole"
            }
        }
    ]
}
```

Otomatiskan manajemen pipeline dinamis untuk menerapkan solusi perbaikan terbaru di lingkungan Gitflow dengan menggunakan dan AWS Service Catalog AWS CodePipeline

Balaji Vedagiri, Faisal Shahdad, Shanmugam Shanker, dan Vivek Thangamuthu, Amazon Web Services

Ringkasan

> **Note**
>
> AWS CodeCommit tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS CodeCommit dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

Pola ini membahas skenario pengelolaan pipeline hotfix dinamis yang didedikasikan semata-mata untuk menerapkan solusi hotfix ke lingkungan produksi dengan aman. Solusinya diimplementasikan dan dikelola dengan menggunakan AWS Service Catalog portofolio dan produk. EventBridge Aturan Amazon digunakan untuk otomatisasi acara. Pembatasan diberlakukan dengan menggunakan batasan portofolio Service Catalog dan peran AWS Identity and Access Management (IAM) untuk pengembang. Hanya AWS Lambda fungsi yang diizinkan untuk meluncurkan produk Service Catalog, yang dipicu oleh EventBridge aturan. Pola ini dirancang untuk lingkungan dengan pengaturan Gitflow tertentu, yang dijelaskan dalam Informasi tambahan.

Biasanya, perbaikan terbaru diterapkan untuk mengatasi masalah kritis atau keamanan yang dilaporkan di lingkungan hidup, seperti Produksi. Perbaikan terbaru harus diterapkan langsung ke lingkungan Pementasan dan Produksi saja. Pipa Pementasan dan Produksi digunakan secara luas untuk permintaan pengembangan reguler. Pipeline ini tidak dapat digunakan untuk menerapkan perbaikan terbaru karena ada fitur berkelanjutan dalam jaminan kualitas yang tidak dapat dipromosikan ke Produksi. Untuk merilis perbaikan terbaru, pola ini menjelaskan pipeline dinamis dan berumur pendek dengan fitur keamanan berikut:

- Pembuatan otomatis - Pipeline hotfix dibuat secara otomatis setiap kali cabang hotfix dibuat di repositori. AWS CodeCommit
- Pembatasan akses — Pengembang tidak memiliki akses untuk membuat pipeline ini di luar proses perbaikan terbaru.

- Tahap terkontrol — Pipeline memiliki tahap terkontrol dengan token akses khusus, memastikan bahwa permintaan tarik (PR) hanya dapat dibuat sekali.
- Tahap persetujuan — Tahapan persetujuan disertakan dalam jalur pipa untuk mendapatkan persetujuan yang diperlukan dari pemangku kepentingan terkait.
- Penghapusan otomatis - Pipa hotfix secara otomatis dihapus setiap kali `hotfix` cabang dihapus di CodeCommit repositori setelah digabungkan dengan PR.

Prasyarat dan batasan

Prasyarat

- Tiga aktif Akun AWS diperlukan sebagai berikut:
  - Akun alat - Untuk pengaturan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD).
  - Akun panggung - Untuk pengujian penerimaan pengguna.
  - Akun produksi - Untuk pengguna akhir bisnis.
  - (Opsional) Tambahkan Akun AWS untuk bertindak sebagai akun QA. Akun ini diperlukan jika Anda menginginkan pengaturan pipeline utama, termasuk QA, dan solusi pipeline hotfix untuk pengujian.
- AWS CloudFormation Tumpukan dengan kondisi opsional untuk diterapkan di akun QA menggunakan pipeline utama, jika diperlukan. Pola masih dapat diuji tanpa pengaturan pipeline utama dengan membuat dan menghapus `hotfix` cabang.
- Bucket Amazon Simple Storage Service (Amazon S3) untuk menyimpan CloudFormation template yang digunakan untuk membuat produk Service Catalog.
- Buat aturan persetujuan PR untuk CodeCommit repositori sesuai dengan persyaratan kepatuhan (setelah membuat repositori).
- Batasi izin IAM dari pengembang dan arahan tim untuk menolak eksekusi fungsi [prcreation-lambda Lambda karena harus dipanggil] hanya dari pipeline.

Batasan

- CloudFormation Penyedia digunakan dalam tahap penerapan, dan aplikasi digunakan menggunakan set CloudFormation perubahan. Jika Anda ingin menggunakan opsi penyebaran yang berbeda, ubah CodePipeline tumpukan sesuai kebutuhan.

- Pola ini menggunakan AWS CodeBuild dan file konfigurasi lainnya untuk menyebarkan layanan mikro sampel. Jika Anda memiliki jenis beban kerja yang berbeda (misalnya, beban kerja tanpa server), Anda harus memperbarui semua konfigurasi yang relevan.
- Pola ini menyebarkan aplikasi dalam satu Wilayah AWS (misalnya, US East (Virginia N.) us-east-1) di seberang. Akun AWS Untuk menyebarkan di beberapa Wilayah, ubah referensi Wilayah dalam perintah dan tumpukan.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat layanan AWS menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram di bagian ini menyediakan alur kerja untuk peristiwa siklus hidup pembuatan dan untuk peristiwa siklus hidup penghapusan.

Diagram sebelumnya untuk membuat peristiwa siklus hidup menunjukkan hal berikut:

1. Pengembang membuat `hotfix-*` cabang di CodeCommit repositori untuk mengembangkan solusi terkait hotfix.
2. Acara pembuatan `hotfix-*` cabang ditangkap melalui EventBridge aturan. Rincian acara termasuk nama repositori dan nama cabang.
3. EventBridge Aturan memanggil AWS Lambda fungsi`hotfix-lambda-function`. EventBridge Aturan meneruskan informasi acara ke fungsi Lambda sebagai input.
4. Fungsi Lambda memproses input untuk mengambil nama repositori dan nama cabang. Ini meluncurkan produk Service Catalog dengan nilai yang diambil dari input yang diproses.
5. Produk Service Catalog mencakup pengaturan pipeline yang akan menyebarkan solusi ke lingkungan Stage dan Production. Blok pipa mencakup tahapan sumber, pembuatan, dan penerapan. Juga, ada tahap persetujuan manual untuk mempromosikan penyebaran untuk lingkungan Produksi.
6. Tahap sumber mengambil kode dari repositori dan `hotfix-*` cabang yang dibuat pada langkah pertama. Kode diteruskan ke tahap pembuatan melalui bucket Amazon S3 untuk artefak. Pada tahap build, image container dibuat yang menyertakan hotfix yang dikembangkan di `hotfix-*` cabang dan didorong ke Amazon Elastic Container Registry (Amazon ECR).

7. Tahap penerapan ke lingkungan panggung memperbarui Amazon Elastic Container Service (Amazon ECS) dengan image container terbaru yang menyertakan hotfix. Hotfix digunakan dengan membuat dan mengeksekusi set perubahan. CloudFormation
8. Fungsi `prcreation-lambda` Lambda dipanggil setelah penerapan berhasil di lingkungan Stage. Fungsi Lambda ini membuat PR dari `hotfix-*` cabang ke `develop` dan `main` cabang repositori. Fungsi Lambda memastikan bahwa perbaikan yang dikembangkan di `hotfix-*` cabang terbelakang dan disertakan dalam penerapan berikutnya.
9. Tahap persetujuan manual membantu memastikan bahwa pemangku kepentingan yang diperlukan meninjau perbaikan dan memberikan persetujuan untuk diterapkan dalam Produksi.
10. Tahap penerapan ke lingkungan produksi memperbarui Amazon ECS dengan gambar kontainer terbaru yang menyertakan perbaikan terbaru. Hotfix digunakan dengan membuat dan mengeksekusi set perubahan. CloudFormation

Diagram sebelumnya untuk menghapus peristiwa siklus hidup menunjukkan hal berikut:

1. Pengembang menghapus `hotfix-*` cabang setelah berhasil menyebarkan hotfix ke lingkungan produksi.
2. Peristiwa penghapusan `hotfix-*` cabang ditangkap melalui aturan. EventBridge Rincian acara termasuk nama repositori dan nama cabang.
3. EventBridge Aturan tersebut memanggil fungsi Lambda. EventBridge Aturan meneruskan informasi acara ke fungsi Lambda sebagai input.
4. Fungsi Lambda memproses input untuk mengambil nama repositori dan nama cabang. Fungsi Lambda menentukan produk Service Catalog masing-masing dari input yang diteruskan dan kemudian mengakhiri produk.
5. Penghentian produk yang disediakan oleh Service Catalog menghapus pipeline dan sumber daya yang relevan yang dibuat sebelumnya dalam produk tersebut.

Otomatisasi dan skala

- Pola ini mencakup EventBridge aturan dan fungsi Lambda, yang dapat menangani beberapa permintaan pembuatan cabang hotfix secara paralel. Fungsi Lambda menyediakan produk Service Catalog untuk aturan acara yang cocok.

- Penyiapan pipeline ditangani dengan menggunakan produk Service Catalog, yang menyediakan kemampuan kontrol versi. Solusi ini juga menskalakan secara otomatis untuk menangani beberapa pengembangan perbaikan terbaru untuk aplikasi yang sama secara paralel.
- Fungsi prcreation-lambda memastikan bahwa perubahan hotfix ini juga digabungkan kembali ke dalam `main` dan `develop` cabang melalui pembuatan permintaan tarik otomatis. Pendekatan ini penting untuk menjaga agar `develop` cabang tetap up to date dengan semua perbaikan dan menghindari potensi regresi kode. `main` Proses ini membantu menjaga konsistensi di seluruh cabang dan mencegah regresi kode dengan memastikan bahwa semua cabang berumur panjang memiliki perbaikan terbaru.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri. AWS CodeCommit tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS CodeCommit dapat terus menggunakan layanan seperti biasa. Untuk informasi selengkapnya, lihat Cara memigrasikan AWS CodeCommit repositori Anda ke penyedia Git lain.
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.

- AWS Service Catalogmembantu Anda mengelola katalog layanan TI secara terpusat yang disetujui. AWS Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- CloudFormation Linter (cfn-lint) adalah linter yang memeriksa template CloudFormation YAMAL atau JSON terhadap spesifikasi sumber daya. CloudFormation Ini juga melakukan pemeriksaan lain, seperti memeriksa nilai yang valid untuk properti sumber daya dan kepatuhan terhadap praktik terbaik.
- cfn-nag adalah alat open source yang mengidentifikasi potensi masalah keamanan dalam CloudFormation template dengan mencari pola.
- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah. Pola ini menggunakan Docker untuk membangun dan menguji gambar kontainer secara lokal.
- Git adalah sistem kontrol versi terdistribusi open-source.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub dynamic_hotfix_codepipeline.

Praktik terbaik

Tinjau dan sesuaikan peran IAM dan kebijakan kontrol layanan (SCP) di lingkungan Anda untuk memastikan bahwa mereka membatasi akses dengan tepat. Ini sangat penting untuk mencegah tindakan apa pun yang dapat mengesampingkan langkah-langkah keamanan yang termasuk dalam pola ini. Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Mengatur lingkungan kerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori sampel ke direktori baru di lokasi kerja Anda, jalankan perintah berikut: `git clone git@github.com:aws-samples/dynamic_hotfix_codepipeline.git`	AWS DevOps
Ekspor variabel lingkungan untuk penyebaran CloudFormation tumpukan.	Tentukan variabel lingkungan berikut yang akan digunakan sebagai masukan ke CloudFormation tumpukan nanti dalam pola ini. • `ApplicationName` - Variabel ini digunakan untuk memberi nama sumber daya yang dibuat untuk aplikasi, sehingga lebih mudah untuk melacaknya. Gunakan perintah berikut, ganti `Applicationname` dengan nama aplikasi Anda yang sebenarnya: `export ApplicationName=<Applicationname>` • `BucketStartName` - Variabel ini untuk penamaan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bucket Amazon S3. Nama bucket S3 harus unik secara global di semua Akun AWS. Gunakan perintah berikut, ganti `BucketName` dengan nama unik untuk bucket S3 Anda: `export BucketStartName=<BucketName>` • Nomor akun dan Wilayah — Variabel ini menyimpan Akun AWS nomor untuk lingkungan yang berbeda dan Wilayah penyebaran. Gunakan perintah berikut, ganti placeholder (seperti `prodaccountnumber` dan`region`) dengan Akun AWS nomor aktual Anda dan Wilayah AWS yang Anda gunakan. **Note** `QAAccount` adalah opsional. Jika Anda ingin menggunak an`QAAccount` , atur dengan menggunakan parameter untuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tumpukan pipa utama. `export ProdAccou nt=<prodaccountnum ber> export StageAcco unt=<stage/preprod accountnumber> export QAAccount =<qaccountnumber> export ToolsAcco unt=<toolsaccountn umber> export DepRegion =<region>`	

Mengatur prasyarat yang diperlukan di Akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sumber daya yang diperlukan untuk CI/CD di akun alat.	Untuk menyebarkan CloudFormation tumpukan di akun alat, gunakan perintah berikut. (Hapus `QAAccount` parameter jika Anda tidak menggunakan akun QA untuk pengaturan.) `#InToolsAccount aws cloudformation deploy \ --template-file pre-requisites/pre-reqs.yaml \`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--stack-name prereqs \
	--parameter-overrides BucketStartName=${BucketStartName} \
	ApplicationName=${ApplicationName} ProdAccount=${ProdAccount} \
	StageAccount=${StageAccount} ToolsAccount=${ToolsAccount} \
	QAAccount=${QAAccount} \
	--capabilities CAPABILITY_IAM CAPABILITY_NAMED_IAM --region ${DepRegion}</pre>
	Catat sumber daya yang dibuat oleh CodeCommit repositori dan Amazon ECR dari tumpukan sebelumnya. Parameter ini diperlukan untuk mengatur `main` cabang pipa di langkah mendatang.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat sumber daya yang diperlukan untuk CI/CD akun beban kerja.	1. Untuk mengemas CloudFormation template di setiap akun beban kerja (tahap, produksi, dan QA opsional), gunakan perintah berikut. Pada perintah berikut, ganti `S3bucketpackage` dengan nama bucket Amazon S3 yang ingin Anda gunakan untuk pengemasan.
	
	`#InStageAccount`
	`aws cloudformation package \`
	`--template-file pre-requisites/infrastack.yaml \`
	`--s3-bucket <S3bucketpackage> \`
	`--s3-prefix infraStack \`
	`--region ${DepRegion} \`
	`--output-template-file pre-requisites/infrastructure_stage.template`
	
	`#InProdAccount`
	`aws cloudformation package \`
	`--template-file pre-requisites/infrastack.yaml \`
	`--s3-bucket <S3bucketpackage> \`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--s3-prefix
	infraStack \
	--region ${DepRegi
	on} \
	--output-
	template-file
	pre-requisites/inf
	rastructure_prod.t
	emplate</pre>
	2. Untuk menyebarkan CloudFormation template di setiap akun beban kerja, gunakan perintah berikut:
	<pre>#InStageAccount
	aws cloudformation
	deploy --stack-name
	inframainstack \
	--parameter-over
	rides Applicati
	onName=${Applicati
	onName} ToolsAcco
	unt=${ToolsAccount
	} DepRegion=${DepReg
	ion} \
	--template-file
	pre-requisites/inf
	rastructure_stage.
	template --region
	${DepRegion}
	--capabilities
	CAPABILITY_NAMED_I
	AM
	
	#InProdAccount
	aws cloudformation
	deploy --stack-name
	inframainstack \
	--parameter-over
	rides Applicati</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`onName=${Applicati onName} ToolsAcco unt=${ToolsAccount } DepRegion=${DepReg ion} \ --template-file pre-requisites/inf rastructure_prod.t emplate --region ${DepRegion} --capabilities CAPABILITY_NAMED_I AM`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui kebijakan bucket artefak S3 untuk mengizinkan akses akun beban kerja.	Untuk memperbarui prasyarat CloudFormation tumpukan di akun alat, gunakan perintah berikut untuk menambahkan semua izin yang diperluka n untuk akun beban kerja Tahap dan Produksi. (Hapus `QAAccount` parameter jika Anda tidak menggunakannya untuk pengaturan.)
	`#InToolsAccount`
	`aws cloudformation deploy \`
	`--template-file pre-requisites/pre-reqs.yaml \`
	`--stack-name prereqs \`
	`--parameter-overrides BucketStartName=${BucketStartName} \`
	`ApplicationName=${ApplicationName} ProdAccount=${ProdAccount} \`
	`StageAccount=${StageAccount} ToolsAccount=${ToolsAccount} \`
	`QAAccount=${QAAccount} PutPolicy=true \`
	`--capabilities CAPABILITY_IAM CAPABILITY_NAMED_IAM --region ${DepRegion}`
		AWS DevOps

Siapkan fungsi Lambda dan sumber daya Service Catalog di akun alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan portofolio dan produk Service Catalog.	Untuk menyiapkan portofolio dan produk Service Catalog, lakukan hal berikut:
	
	1. Unggah template pipeline-main.yaml dan pipeline-hotfix.yaml dari repositori di direktori CodePipeline ke bucket Amazon S3 () yang ada di Wilayah yang ingin Anda terapkan ke (). `Bucketname` `DepRegion`
	
	`#InToolsAccount`
	`aws s3 cp ./codepipeline/pipeline-main.yaml s3://<Bucketname>/pipeline-main.yaml`
	`aws s3 cp ./codepipeline/pipeline-hotfix.yaml s3://<Bucketname>/pipeline-hotfix.yaml`
	
	2. Siapkan portofolio dan produk Service Catalog yang akan mengelola pipeline untuk `main` dan `hotfix` cabang. Catat detail dari `Outputs` bagian untuk `MainProductId` dan `MainProductArtifactId` dari
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tumpukan berikut. Informasi diperlukan pada langkah-l angkah selanjutnya selama pengaturan pipa untuk `main` cabang. `#InToolsAccount aws cloudformation deploy \ --template-file pre-requisites/servicecatalogsetup.yaml \ --stack-name servicecatalogsetup \ --parameter-overrides TemplateBucket=<Bucketname> \ --capabilities CAPABILITY_IAM CAPABILITY_NAMED_IAM --region ${DepRegion}` 3. Menyediakan akses untuk peran IAM yang akan menyebarkan sumber daya di akun alat ke portofolio pipa utama portofolio Service Catalog. Gunakan portofolio ini untuk menyebarkan aplikasi dengan menggunakan `main` cabang. Untuk informasi selengkapnya tentang cara menyediakan akses, lihat Memberikan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Akses kepada Pengguna di dokumentasi Service Catalog.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan fungsi Lambda.	Solusi ini menggunakan fungsi Lambda berikut untuk mengelola alur kerja hotfix: • `hotfix-lambda-function` menangani penyediaan produk Service Catalog saat `hotfix` cabang dibuat. • `hotfix-cleanup-lambda-function` mengelola penghentian produk saat `hotfix` cabang dihapus. • `prcreation-lambda` membuat permintaan tarik dari `hotfix` cabang ke cabang `develop` dan `main` cabang. Untuk mengaktifkan fungsi Lambda untuk menyediakan dan menghentikan produk Service Catalog saat `hotfix` cabang dibuat atau dihapus melalui EventBridge aturan terkait, gunakan langkah-langkah berikut: 1. Untuk membuat peran dan izin IAM untuk fungsi Lambda, gunakan perintah berikut untuk menyebarkan tumpukan: CloudFormation	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`#InToolsAccount aws cloudformation deploy \ --templat e-file pre-requi sites/lambdasetup. yaml \ --stack-n ame prsclambdasetup \ --capabilities CAPABILITY_IAM CAPABILITY_NAMED_I AM --region ${DepRegi on}` 2. Setelah penerapan stack, berikan `hotfix-la mbda-execution-role` akses ke portofoli o pipeline hotfix portofoli o Service Catalog dengan menggunakan file. Konsol Manajemen AWS Akses ini memungkinkan fungsi Lambda untuk meluncurkan atau menghentikan produk Service Catalog untuk cabang hotfix.	

Buat pipeline untuk cabang utama dan terapkan aplikasi di akun beban kerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan pipa untuk `main` cabang.	Untuk mengatur pipeline untuk cabang utama, jalankan perintah berikut di akun alat. Ganti parameter untuk `MainProductId` dan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`MainProductArtifactId` dengan nilai dari output `servicecatalogsetup` tumpukan.
	
	`#InToolsAccount`
	`aws servicecatalog provision-product \`
	`--product-id <MainProductId> \`
	`--provisioning-artifact-id <MainProductArtifactId> \`
	`--provisioned-product-name "${ApplicationName}-main-pipeline" \`
	`--provisioning-parameters Key=CodeCommitRepoName,Value="${ApplicationName}-repository" Key=ECRRepository,Value="${ApplicationName}-app" \`
	`--region=${DepRegion}`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan aplikasi menggunakan `main` cabang.	1. Untuk mengkloning CodeCommit repositori yang dibuat di prasyarat , gunakan perintah. `git clone` Untuk informasi selengkapnya, lihat Connect ke CodeCommit repositori dengan mengkloning repositori seperti yang dijelaskan dalam dokumentasi. CodeCommit 2. Salin semua file aplikasi dari `repotemplates` direktori yang tersedia di repositori ke clone repositori lokal Anda (). `${ApplicationName}-repository` Ubah file berikut untuk memperbarui `ToolsAccount` ID. Di setiap file, cari `RegistryAccountid` parameter dan atur nilainya ke `ToolsAccount` ID Anda. Komit perubahan ke CodeCommit repositori, dan dorong file ke cabang `main` dan `develop` cabang. • ecs-configuration-prod.yaml • ecs-configuration-qa.yaml • ecs-configuration-stage.yaml	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Untuk memverifikasi penerapan aplikasi, pantau CodePipeline eksekusi dengan menggunakan file. Konsol Manajemen AWS Setelah penyebaran selesai, akses aplikasi dengan menggunakan Application Load Balancer FQDN di lingkungan panggung. Konfirmasikan bahwa aplikasi berfungsi seperti yang diharapkan. 4. Untuk menyetujui penerapan ke produksi, gunakan CodePipeline konsol untuk menemukan pipeline untuk aplikasi Anda. Temukan `ApprovalToStart` panggung. Tinjau perubahan dan, jika memuaskan, berikan persetujuan manual untuk melanjutkan penyebaran produksi.	

Buat pipeline untuk cabang hotfix-* dan terapkan perbaikan terbaru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat `hotfix-*` cabang dan komit perubahan.	Untuk membuat pipeline untuk `hotfix-*` cabang dan menerapkan perbaikan terbaru	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ke akun beban kerja, lakukan hal berikut: 1. Buat cabang menggunakan nama yang dimulai dengan kata kuncihotfix. Misalnya, pola ini menggunakan `hotfix-check1` cabang di repositori CodeCommit aplikasi ()`${ApplicationName}-repository` . Untuk langkah-langkah lebih rinci, lihat Connect to an AWS CodeCommit repositori dan perintah Git Dasar dalam dokumentasi. CodeCommit 2. Verifikasi bahwa produk Service Catalog berhasil `Hotfix CICD Pipeline` disediakan secara dinamis untuk cabang. `hotfix-check1` Nama produk yang disediakan dinamai setelah nama cabang hotfix ini dan nama repositori aplikasi. CodeCommit 3. Lakukan beberapa perubahan kecil dalam file index.html, dan dorong ke CodeCommit repositori. 4. Verifikasi bahwa CodePipeline eksekusi berhasil di lingkungan panggung.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk menyebarkan di lingkungan produksi, berikan persetujuan manual di CodePipeline. 5. Konfirmasikan perubahan terlihat di halaman beranda aplikasi dengan menggunakan Application Load Balancer full qualified domain name (FQDN). FQDN tersedia di bagian. `Outputs` `inframain stack-ALBStack-*`	
Hapus `hotfix-check1` cabang.	Untuk menghapus `hotfix-check1` cabang yang dibuat sebelumnya, lakukan hal berikut: 1. Hapus `hotfix-check1` cabang yang ada di repositori CodeCommit aplikasi. 2. Verifikasi bahwa produk Service Catalog yang disediakan untuk `hotfix-check1` cabang berhasil dihentikan.	AWS DevOps

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bersihkan sumber daya yang digunakan.	Untuk membersihkan sumber daya yang digunakan sebelumnya, lakukan hal berikut:
	1. Untuk mengurangi layanan Amazon ECS menjadi nol replika di akun beban kerja, gunakan perintah berikut:
	`aws ecs update-service --cluster ${ApplicationName}-Cluster --service ${ApplicationName}-Service-stage --desired-count 0 --region ${DepRegion}`
	`aws ecs update-service --cluster ${ApplicationName}-Cluster --service ${ApplicationName}-Service-prod --desired-count 0 --region ${DepRegion}`
	2. Mengakhiri produk Service Catalog yang disediakan untuk cabang. `main`
	3. Bersihkan objek yang dibuat di bucket Amazon S3 di akun alat. Hapus semua gambar Docker di Amazon ECR sebelum menghapus registri itu sendiri.
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Hapus peran IAM di bagian akses yang diberikan pada portofolio Service Catalog sebelum Anda menghapus portofolio Service Catalog.
	5. Hapus CloudFormation tumpukan yang digunakan di akun alat dan akun beban kerja.
	
	`##In Tools Account##`
	`aws cloudformation delete-stack --stack-name servicecatalogsetup --region ${DepRegion}`
	`aws cloudformation delete-stack --stack-name prlambdasetup --region ${DepRegion}`
	`aws cloudformation delete-stack --stack-name prereqs --region ${DepRegion}`
	
	`##In Workload Accounts##`
	`aws cloudformation delete-stack --stack-name inframainstack --region ${DepRegion}`
	
	Untuk informasi selengkapnya, lihat Menghapus produk yang disediakan di dokumentasi Service Catalog.	

Pemecahan masalah

Isu	Solusi
Perubahan yang Anda komit ke CodeCommit repositori tidak diterapkan.	Periksa CodeBuild log untuk kesalahan dalam tindakan build Docker. Lihat informasi yang lebih lengkap dalam dokumentasi CodeBuild .
Produk Service Catalog tidak sedang disediaka n.	Tinjau CloudFormation tumpukan terkait untuk peristiwa yang gagal. Lihat informasi yang lebih lengkap dalam dokumentasi CloudFormation .

Sumber daya terkait

- Perintah Git Dasar
- Mengonfigurasi kebijakan IAM untuk membatasi push dan merge ke cabang
- Connect ke AWS CodeCommit repositori
- Memberikan akses ke pengguna
- Mendorong gambar Docker ke repositori pribadi Amazon ECR
- Pemecahan Masalah AWS CodeBuild
- Apa itu AWS CodePipeline?

Informasi tambahan

Pola ini dirancang untuk lingkungan dengan pengaturan Gitflow yang diadopsi untuk alur kerja pengembangan dalam proses CI/CD . Jaringan pipa mengikuti siklus penyebaran yang dimulai dari pengembangan dan bergerak melalui jaminan kualitas (QA), tahap, dan lingkungan produksi. CI/CD Penyiapan mencakup dua cabang git dengan penerapan promosi ke lingkungan sebagai berikut:

- `develop`Cabang menyebar ke lingkungan pengembangan.
- `main`Cabang menyebar ke lingkungan QA, panggung, dan produksi.

Dalam pengaturan ini, merupakan tantangan untuk menerapkan perbaikan terbaru atau patch keamanan lebih cepat daripada siklus penerapan biasa sementara pengembangan aktif fitur baru

sedang berlangsung. Proses khusus diperlukan untuk mengatasi perbaikan terbaru atau permintaan keamanan, memastikan bahwa lingkungan hidup tetap berfungsi dengan baik dan aman.

Namun, Anda dapat menggunakan opsi lain yang tersedia tanpa memerlukan proses penyebaran khusus jika:

- CI/CD Proses ini dilengkapi dengan pengujian otomatis, seperti fungsional dan end-to-end pengujian, yang menghilangkan kebutuhan untuk pengujian manual dan mencegah penundaan penerapan ke produksi. Namun, jika pengujian otomatis tidak terintegrasi dengan baik ke dalam CI/CD proses, mendorong perbaikan kecil ke lingkungan produksi dapat menjadi kompleks dan rumit bagi pengembang. Ini karena mungkin ada fitur baru yang menunggu di lingkungan QA untuk persetujuan dan sign-off. Perbaikan terbaru atau perbaikan keamanan tidak dapat didorong ke produksi secara langsung secara bersamaan.
- Tim pengembangan terus menerapkan fitur baru ke dalam lingkungan produksi, mengintegrasikan perbaikan terbaru atau patch keamanan ke dalam penerapan terjadwal dari setiap fitur baru. Dengan kata lain, pembaruan fitur berikutnya ke lingkungan produksi terdiri dari dua komponen: Penambahan fitur baru dan dimasukkannya hotfix atau patch keamanan. Namun, jika siklus penerapan tidak berkelanjutan, mungkin ada beberapa fitur baru yang sudah menunggu persetujuan di lingkungan QA. Mengelola versi yang berbeda dan memastikan perubahan yang benar diterapkan kembali kemudian dapat menjadi kompleks dan rawan kesalahan.

> **Note**
>
> Jika Anda menggunakan versi 2 AWS CodePipeline dengan pemicu yang tepat yang disiapkan di `hotfix` cabang, Anda masih memerlukan proses khusus untuk menangani permintaan yang tidak terjadwal. Di versi 2, Anda dapat mengatur pemicu untuk permintaan push atau pull. Eksekusi akan segera diantri atau dieksekusi, tergantung pada keadaan pipa sebelumnya. Namun, dengan pipa khusus, perbaikan diterapkan segera ke lingkungan produksi, memastikan bahwa masalah mendesak diselesaikan tanpa penundaan.

Mengotomatiskan penghapusan AWS CloudFormation tumpukan dan sumber daya terkait

SANDEEP SINGH dan James Jacob, Amazon Web Services

Ringkasan

AWS CloudFormationadalah layanan yang banyak digunakan untuk mengelola infrastruktur cloud sebagai kode (IAc). Saat Anda menggunakan CloudFormation, Anda mengelola sumber daya terkait sebagai satu unit yang disebut tumpukan. Anda membuat, memperbarui, dan menghapus kumpulan sumber daya dengan membuat, memperbarui, dan menghapus tumpukan.

Terkadang, Anda tidak lagi membutuhkan sumber daya dalam CloudFormation tumpukan. Bergantung pada sumber daya dan konfigurasinya, mungkin rumit untuk menghapus tumpukan dan sumber daya yang terkait. Dalam sistem produksi dunia nyata, penghapusan terkadang gagal atau memakan waktu lama karena kondisi yang saling bertentangan atau pembatasan yang tidak dapat mengesampingkan. CloudFormation Ini dapat memerlukan perencanaan dan pelaksanaan yang cermat untuk memastikan bahwa semua sumber daya dihapus dengan benar secara efisien dan konsisten. Pola ini menjelaskan cara menyiapkan kerangka kerja yang membantu Anda mengelola penghapusan CloudFormation tumpukan yang melibatkan kompleksitas berikut:

- Sumber daya dengan perlindungan hapus - Beberapa sumber daya mungkin mengaktifkan perlindungan penghapusan. Contoh umum adalah tabel Amazon DynamoDB dan bucket Amazon Simple Storage Service (Amazon S3). Hapus perlindungan mencegah penghapusan otomatis, seperti penghapusan melalui. CloudFormation Jika Anda ingin menghapus sumber daya ini, Anda harus secara manual atau terprogram mengganti atau menonaktifkan sementara perlindungan hapus. Anda harus hati-hati mempertimbangkan implikasi dari menghapus sumber daya ini sebelum melanjutkan.
- Sumber daya dengan kebijakan penyimpanan — Sumber daya tertentu, seperti kunci AWS Key Management Service (AWS KMS) dan bucket Amazon S3, mungkin memiliki kebijakan penyimpanan yang menentukan berapa lama sumber daya tersebut harus dipertahankan setelah penghapusan diminta. Anda harus memperhitungkan kebijakan ini dalam strategi pembersihan untuk menjaga kepatuhan terhadap kebijakan organisasi dan persyaratan peraturan.
- Penghapusan fungsi Lambda yang tertunda yang dilampirkan ke VPC — Menghapus AWS Lambdafungsi yang dilampirkan ke cloud pribadi virtual (VPC) dapat memakan waktu 5—40 menit, tergantung beberapa dependensi yang saling berhubungan yang terlibat dalam proses. Jika Anda

melepaskan fungsi dari VPC sebelum menghapus tumpukan, Anda dapat mengurangi penundaan ini menjadi kurang dari 1 menit.

- Sumber daya tidak dibuat secara langsung oleh CloudFormation — Dalam desain aplikasi tertentu, sumber daya mungkin dibuat di luar CloudFormation tumpukan asli, baik oleh aplikasi itu sendiri atau oleh sumber daya yang disediakan melalui tumpukan. Berikut ini adalah dua contoh:
  - CloudFormation mungkin menyediakan instance Amazon Elastic Compute Cloud (Amazon EC2) yang menjalankan skrip data pengguna. Kemudian, skrip ini mungkin membuat AWS Systems Managerparameter untuk menyimpan data terkait aplikasi. Parameter ini tidak dikelola melalui CloudFormation.
  - CloudFormation mungkin menyediakan fungsi Lambda yang secara otomatis menghasilkan grup Amazon CloudWatch Logs untuk menyimpan log. Grup log ini tidak dikelola melalui CloudFormation.

  Meskipun sumber daya ini tidak dikelola secara langsung oleh CloudFormation, mereka sering perlu dibersihkan ketika tumpukan dihapus. Jika dibiarkan tidak dikelola, mereka dapat menjadi yatim piatu dan menyebabkan konsumsi sumber daya yang tidak perlu.

Meskipun pagar pembatas ini dapat menyebabkan kompleksitas, mereka disengaja dan kritis. Mengizinkan CloudFormation untuk mengesampingkan semua kendala dan menghapus sumber daya tanpa pandang bulu dapat menyebabkan konsekuensi yang merugikan dan tidak terduga dalam banyak skenario. Namun, sebagai seorang DevOps atau insinyur cloud yang bertanggung jawab untuk mengelola lingkungan, ada kalanya mengesampingkan kendala ini mungkin diperlukan, terutama dalam pengembangan, pengujian, atau lingkungan pementasan.

Hasil bisnis yang ditargetkan

Dengan menerapkan kerangka kerja ini, Anda dapat mencapai manfaat berikut:

- Manajemen biaya — Pembersihan lingkungan sementara yang teratur dan efisien, seperti end-to-end atau lingkungan pengujian penerimaan pengguna, membantu mencegah sumber daya berjalan lebih lama dari yang diperlukan. Ini dapat secara signifikan mengurangi biaya.
- Keamanan — Pembersihan otomatis sumber daya yang sudah ketinggalan zaman atau tidak terpakai mengurangi permukaan serangan dan membantu menjaga lingkungan yang aman. AWS
- Efisiensi operasional - Pembersihan rutin dan otomatis dapat memberikan manfaat operasional berikut:

- Skrip otomatis yang menghapus grup log lama atau bucket Amazon S3 kosong dapat meningkatkan efisiensi operasional dengan menjaga lingkungan tetap bersih dan mudah dikelola.
- Menghapus dan membuat ulang tumpukan dengan cepat mendukung iterasi cepat untuk desain dan implementasi, yang dapat menghasilkan arsitektur yang lebih kuat dan tangguh.
- Menghapus dan membangun kembali lingkungan secara teratur dapat membantu Anda mengidentifikasi dan memperbaiki potensi masalah. Ini dapat membantu Anda memastikan bahwa infrastruktur dapat bertahan dalam skenario dunia nyata.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Python versi 3.6 atau yang lebih baru, diinstal
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi

Batasan

- Konvensi penamaan digunakan untuk mengidentifikasi sumber daya yang harus dihapus. Kode sampel dalam pola ini menggunakan awalan untuk nama sumber daya, tetapi Anda dapat menentukan konvensi penamaan Anda sendiri. Sumber daya yang tidak menggunakan konvensi penamaan ini tidak akan diidentifikasi atau kemudian dihapus.

Arsitektur

Diagram berikut menunjukkan bagaimana kerangka kerja ini mengidentifikasi CloudFormation tumpukan target dan sumber daya tambahan yang terkait dengannya.

Diagram menunjukkan alur kerja berikut:

1. Kumpulkan sumber daya — Kerangka kerja otomatisasi menggunakan konvensi penamaan untuk mengembalikan semua CloudFormation tumpukan yang relevan, repositori Amazon Elastic Container Registry (Amazon ECR), tabel DynamoDB, dan bucket Amazon S3.

> **Note**
>
> Fungsi untuk tahap ini menggunakan paginator, fitur di Boto3 yang mengabstraksi proses iterasi di atas set hasil API terpotong. Ini memastikan bahwa semua sumber daya diproses. Untuk lebih mengoptimalkan kinerja, pertimbangkan untuk menerapkan pemfilteran sisi server atau pertimbangkan JMESPath untuk menggunakan untuk melakukan pemfilteran sisi klien.

2. Pra-pemrosesan — Kerangka otomatisasi mengidentifikasi dan mengatasi kendala layanan yang harus diganti agar memungkinkan untuk menghapus sumber daya. CloudFormation Misalnya, mengubah `DeletionProtectionEnabled` pengaturan untuk tabel DynamoDB menjadi. `False` Di antarmuka baris perintah, untuk setiap sumber daya, Anda menerima prompt yang menanyakan apakah Anda ingin mengganti kendala.
3. Hapus tumpukan - Kerangka otomatisasi menghapus CloudFormation tumpukan. Di antarmuka baris perintah, Anda menerima prompt yang menanyakan apakah Anda ingin menghapus tumpukan.
4. Pasca-pemrosesan - Kerangka otomatisasi menghapus sumber daya terkait yang tidak disediakan secara langsung CloudFormation sebagai bagian dari tumpukan. Contoh jenis sumber daya ini termasuk parameter Systems Manager dan grup CloudWatch log. Fungsi terpisah mengumpulkan sumber daya ini, memprosesnya terlebih dahulu, dan kemudian menghapusnya. Di antarmuka baris perintah, untuk setiap sumber daya, Anda menerima prompt yang menanyakan apakah Anda ingin menghapus sumber daya.

> **Note**
>
> Fungsi untuk tahap ini menggunakan paginator, fitur di Boto3 yang mengabstraksi proses iterasi di atas set hasil API terpotong. Ini memastikan bahwa semua sumber daya diproses. Untuk lebih mengoptimalkan kinerja, pertimbangkan untuk menerapkan pemfilteran sisi server atau pertimbangkan JMESPath untuk menggunakan untuk melakukan pemfilteran sisi klien.

Otomatisasi dan skala

Jika CloudFormation tumpukan Anda menyertakan sumber daya lain yang tidak termasuk dalam kode sampel, atau jika tumpukan memiliki kendala yang belum ditangani dalam pola ini, maka Anda

dapat menyesuaikan kerangka kerja otomatisasi untuk kasus penggunaan Anda. Ikuti metodologi yang sama untuk mengumpulkan sumber daya, pra-pemrosesan, menghapus tumpukan, dan kemudian pasca-pemrosesan.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- CloudFormation Command Line Interface (CFN-CLI) adalah alat open source yang membantu Anda mengembangkan dan menguji dan ekstensi pihak ketiga AWS dan kemudian mendaftarkannya untuk digunakan. CloudFormation
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS

Alat lainnya

- Klik adalah alat Python yang membantu Anda membuat antarmuka baris perintah.
- Puisi adalah alat untuk manajemen ketergantungan dan pengemasan dengan Python.
- Pyenv adalah alat yang membantu Anda mengelola dan beralih antar versi Python.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di GitHub cloudformation-stack-cleanuprepositori.

Praktik terbaik

- Menandai sumber daya untuk identifikasi yang mudah — Menerapkan strategi penandaan untuk mengidentifikasi sumber daya yang dibuat untuk lingkungan dan tujuan yang berbeda. Tag dapat menyederhanakan proses pembersihan dengan membantu Anda memfilter sumber daya berdasarkan tag mereka.
- Siapkan siklus hidup sumber daya — Tentukan siklus hidup sumber daya untuk menghapus sumber daya secara otomatis setelah periode tertentu. Praktik ini membantu Anda memastikan bahwa lingkungan sementara tidak menjadi kewajiban biaya permanen.

Epik

Instal alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	1. Buat folder di lingkungan virtual Anda. Beri nama dengan nama proyek Anda. 2. Buka terminal di mesin lokal Anda, dan arahkan ke folder ini. 3. Masukkan perintah berikut untuk mengkloning cloudformation-stack-cleanup repositori ke direktori proyek Anda: `git clone https://github.com/aws-samples/cloudformation-stack-cleanup.git`	DevOps insinyur
Instal Puisi.	Ikuti instruksi (Dokumentasi puisi) untuk menginstal Puisi di lingkungan virtual target.	DevOps insinyur

Instal dependensi.	1. Masukkan perintah berikut untuk menavigasi ke direktori proyek:
	`cd cloudformation-stack-cleanup`
	2. Masukkan perintah berikut:
	`poetry install`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Ini menginstal semua dependensi yang diperlukan, seperti Boto3, klik, dan kode sumber untuk CLI. CloudFormation	
(Opsional) Instal Pyenv.	Ikuti instruksi (GitHub) untuk menginstal Pyenv.	DevOps insinyur

(Opsional) Sesuaikan kerangka kerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi yang mengumpulkan, pra-proses, dan menghapus sumber daya target.	1. Di repositori kloning, masukkan perintah berikut untuk menavigasi ke direktori: `cli` `cd cfncli/cli` 2. Buka file cleanup_environment.py. 3. Buat fungsi Python baru yang mengumpulkan jenis sumber daya yang ingin Anda modifikasi. Sebagai contoh, lihat `gather_ddb_tables` fungsi dalam file ini. 4. Buat fungsi Python baru yang mengesampingkan kendala layanan untuk sumber daya target. Sebagai contoh, lihat	DevOps insinyur, Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`remove_ddb_deletion_protection` fungsi dalam file ini. 5. Buat fungsi Python baru yang mengumpulkan sumber daya target yang tidak dikelola. Sebagai contoh, lihat `gather_log_groups` fungsi dalam file ini. 6. Buat fungsi Python baru yang menghapus sumber daya target yang tidak dikelola. Sebagai contoh, lihat `delete_log_group` fungsi dalam file ini. 7. Simpan dan tutup file cleanup_enviornment.py.	

Buat sumber daya sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan.	1. Arahkan ke direktori proyek . 2. Masukkan perintah berikut untuk membuat CloudFormation tumpukan yang menyediakan tabel DynamoDB dan grup keamanan. Perbarui nilai untuk`<VPCID>`:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws cloudformation create-stack \ --stack-name sampleforcleanup-Stack \ --template-body file://samples/sample-cfn-stack.yaml \ --parameters ParameterKey=VpcId,ParameterValue=<VPCID> \ --region us-east-1`	
Buat parameter Systems Manager.	Masukkan perintah berikut untuk membuat parameter Systems Manager yang tidak disediakan melalui: CloudFormation `aws ssm put-parameter \ --name "/samplefor cleanup/database/password" \ --value "your_db_password" \ --type "SecureString" \ --description "Database password for my app" \ --tier "Standard" \ --region "us-east-1"`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket Amazon S3.	Masukkan perintah berikut untuk membuat bucket Amazon S3 yang tidak disediakan melalui: CloudFormation `aws s3api create-bucket \ --bucket samplesor cleanup-unmanagedb ucket-<UniqueIdent ifier> \ --region us-east-1 \ --create-bucket-co nfiguration LocationC onstraint=us-east-1`	AWS DevOps

Hapus sumber daya sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus CloudFormation tumpukan.	1. Masukkan perintah berikut untuk menghapus CloudFormation tumpukan sampel, parameter Systems Manager, dan bucket Amazon S3 yang Anda buat: `cfncli --region us-east-1 \ dev cleanup-env \ --prefix-list sampleforcleanup`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Saat diminta, masukkan Y untuk melanjutkan.	
Validasi penghapusan sumber daya.	Dalam output, konfirmasikan bahwa semua sumber daya sampel telah dihapus. Untuk keluaran sampel, lihat bagian Sumber daya tambahan dari pola ini.	AWS DevOps

Sumber daya terkait

- Hapus tumpukan (CloudFormation dokumentasi)
- Pemecahan masalah CloudFormation (dokumentasi) CloudFormation
- Memberikan fungsi Lambda akses ke sumber daya di Amazon VPC (dokumentasi Lambda)
- Bagaimana cara menghapus AWS CloudFormation tumpukan yang macet dalam status DELETE_FAILED? (Pusat AWS Pengetahuan)

Informasi tambahan

Berikut ini adalah contoh output dari `cfncli` perintah:

```
cfncli --region aus-east-1  dev cleanup-env --prefix-list sampleforcleanup


https://sts.us-east-1.amazonaws.com
Cleaning up: ['sampleforcleanup'] in xxxxxxxxxx:us-east-1
Do you want to proceed? [Y/n]: Y
No S3 buckets
No ECR repositories
No Lambda functions in VPC
The following DynamoDB tables will have their deletion protection removed:
sampleforcleanup-MyDynamoDBTable
Do you want to proceed with removing deletion protection from these tables? [Y/n]: Y
Deletion protection disabled for DynamoDB table 'sampleforcleanup-MyDynamoDBTable'.
The following CloudFormation stacks will be deleted:
```

```
sampleforcleanup-Stack
Do you want to proceed with deleting these CloudFormation stacks? [Y/n]: Y
Initiated deletion of CloudFormation stack: `sampleforcleanup-Stack`
Waiting for stack `sampleforcleanup-Stack` to be deleted...
CloudFormation stack `sampleforcleanup-Stack` deleted successfully.
The following ssm_params will be deleted:
/sampleforcleanup/database/password
Do you want to proceed with deleting these ssm_params? [Y/n]: Y
Deleted SSM Parameter: /sampleforcleanup/database/password
Cleaned up: ['sampleforcleanup']
```

Otomatiskan konsumsi dan visualisasi metrik kustom Amazon MWAA di Amazon Managed Grafana dengan menggunakan Terraform

Faisal Abdullah dan Satya Vajrapu, Amazon Web Services

Ringkasan

Pola ini membahas cara menggunakan Grafana Terkelola Amazon untuk membuat dan memantau metrik khusus yang dicerna oleh Alur Kerja Terkelola Amazon untuk Apache Airflow (Amazon MWAA). Amazon MWAA berfungsi sebagai orkestrator untuk alur kerja, menggunakan Directed Acyclic Graphs () yang ditulis dengan Python. DAGs Pola ini berpusat pada pemantauan metrik kustom, termasuk jumlah total DAGs berjalan dalam satu jam terakhir, hitungan lulus dan gagal DAGs setiap jam, dan durasi rata-rata proses ini. Analisis ini menunjukkan bagaimana Grafana Terkelola Amazon berintegrasi dengan Amazon MWAA untuk memungkinkan pemantauan dan wawasan komprehensif tentang orkestrasi alur kerja dalam lingkungan ini.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan izin pengguna yang diperlukan untuk membuat dan mengelola hal berikut: Layanan AWS
  - AWS Identity and Access Management (IAM) peran dan kebijakan
  - AWS Lambda
  - Amazon Managed Grafana
  - Amazon Managed Workflows for Apache Airflow (Amazon MWAA)
  - Amazon Simple Storage Service (Amazon S3)
  - Amazon Timestream
- Akses ke lingkungan shell yang dapat menjadi terminal pada mesin lokal Anda atau AWS CloudShell.
- Lingkungan shell dengan Git diinstal dan versi terbaru dari AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLI dokumentasi. AWS CLI
- Versi Terraform berikut diinstal: `required_version = ">= 1.6.1, < 2.0.0"` Anda dapat menggunakan tfswitch untuk beralih di antara berbagai versi Terraform.

- Sumber identitas yang dikonfigurasi AWS IAM Identity Center untuk Anda Akun AWS. Untuk informasi selengkapnya, lihat Konfirmasi sumber identitas Anda di Pusat Identitas IAM di dokumentasi Pusat Identitas IAM. Anda dapat memilih dari default Direktori Pusat Identitas IAM, Active Directory, atau penyedia Identitas eksternal (iDP) seperti Okta. Untuk informasi selengkapnya, lihat Sumber daya terkait.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Terraform `required_version = ">= 1.6.1, < 2.0.0"`
- Grafana yang Dikelola Amazon versi 9.4 atau yang lebih baru. Pola ini diuji pada versi 9.4.

Arsitektur

Diagram arsitektur berikut menyoroti yang Layanan AWS digunakan dalam solusi.

Diagram sebelumnya melangkah melalui alur kerja berikut:

1. Metrik khusus dalam Amazon MWAA berasal dari DAGs yang dijalankan di dalam lingkungan. Metrik diunggah ke bucket Amazon S3 dalam format file CSV. Berikut ini DAGs menggunakan kemampuan kueri database Amazon MWAA:
   - `run-example-dag`— DAG ini berisi contoh kode Python yang mendefinisikan satu atau lebih tugas. Ini berjalan setiap 7 menit dan mencetak tanggal. Setelah mencetak tanggal, DAG menyertakan tugas untuk tidur, atau menjeda, eksekusi untuk durasi tertentu.
   - `other-sample-dag`— DAG ini berjalan setiap 10 menit dan mencetak tanggalnya. Setelah mencetak tanggal, DAG menyertakan tugas untuk tidur, atau menjeda, eksekusi untuk durasi tertentu.
   - `data-extract`— DAG ini berjalan setiap jam dan menanyakan database Amazon MWAA dan mengumpulkan metrik. Setelah metrik dikumpulkan, DAG ini menuliskannya ke bucket Amazon S3 untuk diproses dan dianalisis lebih lanjut.

2. Untuk merampingkan pemrosesan data, fungsi Lambda berjalan saat dipicu oleh peristiwa Amazon S3, yang memfasilitasi pemuatan metrik ke Timestream.
3. Timestream terintegrasi sebagai sumber data dalam Grafana Terkelola Amazon tempat semua metrik khusus dari Amazon MWAA disimpan.
4. Pengguna dapat melakukan kueri data dan membuat dasbor khusus untuk memvisualisasikan indikator kinerja utama dan mendapatkan wawasan tentang orkestrasi alur kerja dalam Amazon MWAA.

Alat

Layanan AWS

- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua aplikasi Anda Akun AWS dan cloud.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan. Dalam pola ini, AWS Lambda jalankan kode Python sebagai respons terhadap peristiwa Amazon S3 dan mengelola sumber daya komputasi secara otomatis.
- Grafana Terkelola Amazon adalah layanan visualisasi data terkelola sepenuhnya yang dapat Anda gunakan untuk menanyakan, mengkorelasikan, dan memvisualisasikan, serta memperingatkan metrik, log, dan jejak Anda. Pola ini menggunakan Grafana Terkelola Amazon untuk membuat dasbor visualisasi dan peringatan metrik.
- Amazon Managed Workflows for Apache Airflow (Amazon MWAA) adalah layanan orkestrasi terkelola untuk Apache Airflow yang dapat Anda gunakan untuk menyiapkan dan mengoperasikan pipeline data di cloud dalam skala besar. Apache Airflow adalah alat open source yang digunakan untuk secara terprogram menulis, menjadwalkan, dan memantau urutan proses dan tugas yang disebut sebagai alur kerja. Dalam pola ini, sampel DAGs dan ekstraktor metrik DAG digunakan di Amazon MWAA.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Dalam pola ini, Amazon S3 digunakan untuk menyimpan DAGs, skrip, dan metrik khusus dalam format CSV.
- Amazon Timestream for LiveAnalytics is adalah database deret waktu yang cepat, terukur, dikelola sepenuhnya, dan dibuat khusus yang memudahkan penyimpanan dan analisis triliunan titik data deret waktu per hari. Timestream for LiveAnalytics juga terintegrasi dengan layanan yang umum

digunakan untuk pengumpulan data, visualisasi, dan pembelajaran mesin. Dalam pola ini, ini digunakan untuk menelan metrik kustom Amazon MWAA yang dihasilkan.

Alat-alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud. Pola ini menggunakan modul Terraform untuk mengotomatiskan penyediaan infrastruktur di. AWS

Repositori kode

Kode untuk pola ini tersedia GitHub di repositori visualize-amazon-mwaa-custom-metrics-grafana. `stacks/InfraFolder` berisi yang berikut:

- File konfigurasi Terraform untuk semua sumber daya AWS
- Grafana dasbor file.json di folder `grafana`
- Alur Kerja Terkelola Amazon untuk Apache DAGs Airflow di folder `mwaa/dags`
- Kode Lambda untuk mengurai file.csv dan menyimpan metrik di database Timestream di folder `src`
- Kebijakan IAM. File json di folder `templates`

Praktik terbaik

Terraform harus menyimpan status tentang infrastruktur dan konfigurasi terkelola Anda sehingga dapat memetakan sumber daya dunia nyata ke konfigurasi Anda. Secara default, Terraform menyimpan status secara lokal dalam file bernama. `terraform.tfstate` Sangat penting untuk memastikan keamanan dan integritas file status Terraform Anda karena file tersebut mempertahankan status infrastruktur Anda saat ini. Untuk informasi selengkapnya, lihat Remote State dalam dokumentasi Terraform.

Epik

Terapkan infrastruktur menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan infrastruktur.	Untuk menerapkan infrastru ktur solusi, lakukan hal berikut:
	1. Buka terminal atau command prompt di komputer lokal Anda atau dengan menggunakan AWS CloudShell.
	2. Arahkan ke direktori tempat Anda ingin mengkloning repositori.
	3. Untuk mengkloning repositori, jalankan perintah berikut:
	`git clone https://github.com/aws-samples/visualize-amazon-mwaa-custom-metrics-grafana`
	4. Setelah proses kloning selesai, jalankan perintah berikut untuk menavigasi ke direktori repositori kloning:
	`cd visualize-amazon-mwaa-custom-metrics-grafana/stacks/infra`
	5. Untuk mengunduh dan menginisialisasi penyedia
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	yang diperlukan, jalankan perintah berikut:
	`terraform init`
	6. Untuk mendapatkan tampilan komprehensif dari semua sumber daya yang akan dibuat Terraform, jalankan perintah berikut:
	`terraform plan`
	Terraform menyediakan sumber daya berikut:
	• Amazon Virtual Private Cloud (Amazon VPC) dan komponen jaringan terkait
	• Sumber daya Amazon S3
	• AWS Lambda fungsi
	• Sumber daya Grafana yang Dikelola Amazon (ruang kerja, dasbor, sumber data)
	• Mendukung sumber daya IAM (peran dan kebijakan)
	7. Untuk membuat AWS sumber daya dari output rencana, jalankan perintah berikut:
	`terraform apply -auto-approve`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Penyediaan infrastruktur selesai dalam waktu kurang lebih 20 menit. 8. Untuk membuat AWS sumber daya yang ditentukan sesuai dengan konfigurasi yang ditentukan dalam file Terraform Anda, jalankan perintah berikut: `terraform apply`	

Validasi sumber daya infrastruktur yang digunakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi lingkungan Amazon MWAA.	Untuk memvalidasi lingkungan Amazon MWAA, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, navigasikan ke konsol dasbor Amazon MWAA dan pilih Open Airflow UI. 2. Anda akan melihat tiga berikut DAGs dalam status Aktif: • data-ekstrak • run-example-dag • other-sample-dag	AWS DevOps, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Jika DAG tidak aktif, Anda dapat mengaktifkannya dengan mengaktifkan sakelar sakelar di sebelah nama DAG.	
Verifikasi jadwal DAG.	Untuk melihat setiap jadwal DAG, buka tab Jadwal di UI Aliran Udara. Masing-masing dari berikut ini DAGs memiliki jadwal pra-konfigurasi, yang berjalan di lingkungan Amazon MWAA dan menghasilkan metrik khusus: • run-example-dag- Berjalan setiap 7 menit • other-sample-dag- Berjalan setiap 10 menit • data-extract - Berjalan setiap jam Anda juga dapat melihat keberhasilan menjalankan setiap DAG di bawah kolom Runs.	Insinyur data, AWS DevOps

Konfigurasikan lingkungan Grafana yang Dikelola Amazon

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan akses ke ruang kerja Grafana Terkelola Amazon.	Skrip Terraform membuat ruang kerja, dasbor, dan metrik Grafana Terkelola Amazon yang diperluka n. Untuk mengonfigurasi akses sehingga Anda dapat melihatnya, lakukan hal berikut: 1. Buka konsol Grafana yang Dikelola Amazon. 2. Di Ruang kerja, pilih ruang kerja`grafana-ws-dev`, dan arahkan ke tab Otentikasi di panel bawah. 3. Pilih tombol Tetapkan pengguna atau grup baru. 4. Tambahkan grup Anda di tab Grup atau pengguna di tab Pengguna, lalu pilih tombol Tetapkan pengguna dan grup. 5. Setelah pengguna (atau grup) ditambahkan, jadikan pengguna ini (atau grup) admin. Pilih pengguna di Pengguna atau grup yang ditugaskan di tab grup pengguna yang ditugaska n dan pilih Buat admin dari menu tarik-turun. Untuk informasi selengkap	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nya, lihat Menggunakan AWS IAM Identity Center dengan ruang kerja Grafana Terkelola Amazon di dokumentasi Grafana Terkelola Amazon. 6. Arahkan ke Ruang Kerja, lalu pilih URL ruang kerja Grafana. Untuk masuk ke Grafana Terkelola Amazon sebagai admin, pilih Masuk dengan. AWS IAM Identity Center	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal plugin Amazon Timestream.	Metrik kustom Amazon MWAA dimuat ke database Timestream. Anda menggunakan plugin Timestream untuk memvisualisasikan metrik dengan dasbor Grafana Terkelola Amazon. Untuk menginstal plugin Timestream, lakukan hal berikut: 1. Di konsol Grafana yang Dikelola Amazon, perluas menu di panel navigasi kiri dan buka Administrasi, Plugin. 2. Cari dan kemudian instal versi terbaru dari plugin Amazon Timestream. 3. Setelah plugin diinstal, buka Administrasi, Sumber data untuk melihat sumber data Timestream. Jika sumber data tidak terdaftar, segarkan halaman. Untuk informasi selengkapnya, lihat Memperluas ruang kerja Anda dengan plugin di dokumentasi Grafana Terkelola Amazon.	AWS DevOps, DevOps insinyur

Visualisasikan metrik khusus di dasbor Grafana Terkelola Amazon

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat dasbor Grafana Terkelola Amazon.	Untuk melihat metrik yang dimasukkan ke dalam ruang kerja Grafana Terkelola Amazon, lakukan hal berikut: 1. Di konsol Grafana Terkelola Amazon, pilih Dasbor di panel navigasi kiri. 2. Untuk melihat metrik, pilih dasbor peristiwa MWAA lalu pilih mwaa_metrics. Halaman metrik dasbor menampilkan informasi berikut: • Total DAG berjalan dalam satu jam terakhir • Total berhasil, gagal, dan menjalankan DAG berjalan dalam satu jam terakhir • Durasi rata-rata untuk semua, berhasil, dan gagal DAG berjalan	AWS DevOps
Sesuaikan dasbor Grafana Terkelola Amazon.	Untuk menyesuaikan dasbor untuk penyempurnaan masa depan lebih lanjut, lakukan hal berikut: 1. Di `mwaa_metrics` halaman dasbor Grafana	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Terkelola Amazon, pilih ikon Pengaturan Dasbor. 2. Untuk melihat struktur data yang mendefinisikan dasbor, pilih model JSON. Anda dapat menyesuaikan dasbor dengan mengedit model JSON ini langsung di konsol. Atau, kode sumber untuk dasbor ini tersedia dalam `dashboard.json` file di `stacks/infra/grafana` folder di GitHub repositori.	

Bersihkan AWS sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jeda Amazon MWAA DAG berjalan.	Untuk menjeda DAG berjalan, lakukan hal berikut: 1. Di konsol Amazon MWAA, navigasikan ke lingkungan Airflow dan pilih Open Airflow UI. 2. Untuk menjeda DAG, gunakan sakelar sakelar di sebelah setiap DAG. 3. Segarkan halaman UI Airflow, yang seharusnya	AWS DevOps, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mencantumkan tiga DAGs di bagian Dijeda.	
Hapus objek di bucket Amazon S3.	Untuk menghapus bucket Amazon S3 mwaa-events-bucket-* dan mwaa-metrics-bucket-*, ikuti petunjuk penggunaan konsol Amazon S3 di Menghapus bucket di dokumentasi Amazon S3.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hancurkan sumber daya yang dibuat oleh Terraform.	Untuk menghancurkan sumber daya yang dibuat oleh Terraform dan file status Terraform lokal terkait, lakukan hal berikut: 1. (Opsional) Sebelum menghapus sumber daya, Anda dapat melihat pratinjau perubahan yang akan dilakukan Terraform . Untuk membuat rencana, jalankan perintah berikut: `terraform plan -destroy` Output perintah menunjukkan bahwa `destroy` perintah akan menghapus semua AWS sumber daya yang dibuat sebelumnya. 2. `terraform destroy -auto-approve` Perintah ini membutuhkan waktu sekitar 20 menit untuk menghancurkan infrastruktur. **Note** Untuk menghancurkan semua sumber daya yang dikelola	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	oleh Terraform, jalankan perintah berikut. : `-auto-approve` Tag tidak menunggu konfirmasi pengguna untuk mulai menghancurkan sumber daya.
	3. Untuk menghapus file status Terraform lokal, jalankan perintah berikut:
	`rm .terraform.lock.hcl`
	`rm -rf .terraform`
	`rm terraform.tfstate*`	

Pemecahan masalah

Isu	Solusi
`null_resource.plugin_mgmt (local-exec): aws: error: argument operation: Invalid choice, valid choices are:`	Tingkatkan versi Anda AWS CLI ke versi terbaru.
Memuat sumber data kesalahan - `Fetch error: 404 Not Found Instantiating…`	Kesalahannya terputus-putus. Tunggu beberapa menit, lalu segarkan sumber data Anda untuk melihat sumber data Timestream yang terdaftar.

Sumber daya terkait

AWS dokumentasi

- Grafana yang Dikelola Amazon untuk dasbor dan visualisasi
- Konfigurasikan Grafana Terkelola Amazon untuk menggunakan Okta
- Gunakan AWS IAM Identity Center dengan ruang kerja Grafana Terkelola Amazon Anda
- Bekerja dengan DAGs di Amazon MWAA

AWS video

- Konfigurasikan Pusat Identitas IAM dengan Grafana Terkelola Amazon untuk otentikasi, seperti yang ditunjukkan dalam video berikut.

https://www.youtube-nocookie.com/embed/XX2Xcz-Ps9U? kontrol=0

- Jika Pusat Identitas IAM tidak tersedia, Anda juga dapat mengintegrasikan autentikasi Grafana Terkelola Amazon dengan menggunakan penyedia Identitas eksternal (iDP) seperti Okta, seperti yang ditunjukkan dalam video berikut.

https://www.youtube-nocookie.com/embed/Z4 JHxl2xp Og? kontrol=0

Informasi tambahan

Anda dapat membuat solusi pemantauan dan peringatan komprehensif untuk lingkungan Amazon MWAA Anda, memungkinkan manajemen proaktif dan respons cepat terhadap potensi masalah atau anomali. Grafana yang Dikelola Amazon mencakup kemampuan berikut:

Peringatan — Anda dapat mengonfigurasi lansiran di Grafana Terkelola Amazon berdasarkan ambang batas atau kondisi yang telah ditentukan sebelumnya. Siapkan notifikasi email untuk mengingatkan pemangku kepentingan terkait ketika metrik tertentu melebihi atau jatuh di bawah ambang batas yang ditentukan. Untuk informasi selengkapnya, lihat Peringatan Grafana di dokumentasi Grafana Terkelola Amazon.

Integrasi — Anda dapat mengintegrasikan Grafana Terkelola Amazon dengan berbagai alat pihak ketiga seperti OpsGenie, PagerDuty, atau Slack untuk kemampuan notifikasi yang ditingkatkan. Misalnya, Anda dapat mengatur webhook atau mengintegrasikan dengan APIs untuk memicu insiden dan pemberitahuan di platform ini berdasarkan peringatan yang dihasilkan di Amazon Managed

Grafana. Selain itu, pola ini menyediakan GitHub repositori untuk membuat AWS sumber daya. Anda dapat mengintegrasikan kode ini lebih lanjut dengan alur kerja penerapan infrastruktur Anda.

Otomatiskan penerapan set tumpukan dengan menggunakan AWS dan AWS CodePipeline CodeBuild

Thiyagarajan Mani, Mihir Borkar, dan Raghu Gowda, Amazon Web Services

Ringkasan

Dalam integrasi berkelanjutan dan pengiriman berkelanjutan Anda (CI/CD) processes, you might want to deploy applications automatically into all your existing AWS accounts and into new accounts that you add to your organization in AWS Organizations. When you architect a CI/CDsolusi untuk persyaratan ini, kemampuan administrator set tumpukan AWS CloudFormation yang didelegasikan berguna karena memungkinkan lapisan keamanan dengan membatasi akses ke akun manajemen. Namun, AWS CodePipeline menggunakan model izin yang dikelola layanan untuk menyebarkan aplikasi ke beberapa akun dan Wilayah. Anda harus menggunakan akun manajemen AWS Organizations untuk menerapkan dengan kumpulan tumpukan, karena AWS CodePipeline tidak mendukung fitur administrator kumpulan tumpukan yang didelegasikan.

Pola ini menjelaskan bagaimana Anda dapat mengatasi batasan ini. Pola ini menggunakan AWS CodeBuild dan skrip khusus untuk mengotomatiskan penerapan set tumpukan dengan AWS. CodePipeline Ini mengotomatiskan aktivitas penyebaran aplikasi ini:

- Menyebarkan aplikasi sebagai set tumpukan ke unit organisasi yang ada () OUs
- Memperluas penyebaran aplikasi ke tambahan OUs dan Wilayah
- Menghapus aplikasi yang digunakan dari semua atau spesifik OUs atau Wilayah

Prasyarat dan batasan

Prasyarat

Sebelum Anda mengikuti langkah-langkah dalam pola ini:

- Buat organisasi di akun manajemen AWS Organizations Anda. Untuk petunjuk, lihat dokumentasi AWS Organizations.
- Aktifkan akses tepercaya antara AWS Organizations dan CloudFormation untuk menggunakan izin yang dikelola layanan. Untuk petunjuk, lihat Mengaktifkan akses tepercaya dengan AWS Organizations dalam CloudFormation dokumentasi.

Batasan

Kode yang disertakan dengan pola ini memiliki batasan berikut:

- Anda hanya dapat menerapkan satu CloudFormation templat untuk aplikasi; beberapa penerapan templat saat ini tidak didukung.
- Menyesuaikan implementasi saat ini membutuhkan DevOps keahlian.
- Pola ini tidak menggunakan kunci AWS Key Management System (AWS KMS). Namun, Anda dapat mengaktifkan fungsi ini dengan mengkonfigurasi ulang CloudFormation template yang disertakan dengan pola ini.

Arsitektur

Arsitektur untuk pipeline CI/CD penerapan ini menangani hal berikut:

- Membatasi akses langsung ke akun manajemen dengan mendelegasikan tanggung jawab penyebaran set tumpukan ke CI/CD akun khusus sebagai administrator set tumpukan untuk penerapan aplikasi.
- Menggunakan model izin yang dikelola layanan untuk menyebarkan aplikasi secara otomatis setiap kali akun baru dibuat dan dipetakan di bawah OU.
- Memastikan konsistensi versi aplikasi di semua akun di tingkat lingkungan.
- Menggunakan beberapa tahap persetujuan di tingkat repositori dan pipa untuk menyediakan lapisan keamanan dan tata kelola tambahan untuk aplikasi yang digunakan.
- Mengatasi batasan saat ini CodePipeline dengan menggunakan skrip penerapan yang dibuat khusus CodeBuild untuk menerapkan atau menghapus kumpulan tumpukan dan instance tumpukan secara otomatis. Untuk ilustrasi kontrol aliran dan hierarki panggilan API yang diterapkan oleh skrip kustom, lihat bagian Informasi tambahan.
- Membuat set tumpukan individu untuk pengembangan, pengujian, dan lingkungan produksi. Selain itu, Anda dapat membuat kumpulan tumpukan yang menggabungkan beberapa OUs dan Wilayah di setiap tahap. Misalnya, Anda dapat menggabungkan sandbox dan pengembangan OUs dalam tahap penyebaran pengembangan.
- Mendukung penerapan aplikasi ke, atau pengecualian dari, subset akun atau daftar. OUs

Otomatisasi dan skala

Anda dapat menggunakan kode yang disediakan dengan pola ini untuk membuat CodeCommit repositori AWS dan pipeline kode untuk aplikasi Anda. Anda kemudian dapat menerapkan ini sebagai

set tumpukan ke beberapa akun di tingkat OU. Kode ini juga mengotomatiskan komponen seperti topik Amazon Simple Notification Service (Amazon SNS) untuk memberi tahu pemberi persetujuan, peran AWS Identity and Access Management (IAM) yang diperlukan, dan kebijakan kontrol layanan (SCP) untuk diterapkan di akun manajemen.

Alat

Layanan AWS

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS CodeBuild adalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodeDeploy mengotomatiskan penerapan ke Amazon Elastic Compute Cloud (Amazon EC2) atau instans lokal, fungsi AWS Lambda, atau layanan Amazon Elastic Container Service (Amazon ECS).
- AWS CodePipeline membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS Organizations adalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub automated-code-pipeline-stackset-deployment. Untuk struktur folder dan detail lainnya, lihat file readme untuk repositori.

Praktik terbaik

Pola ini membatasi akses langsung ke akun manajemen saat menerapkan aplikasi di tingkat OU. Menambahkan beberapa tahapan persetujuan ke proses pipeline dan repositori membantu

memberikan keamanan dan tata kelola tambahan untuk aplikasi dan komponen yang Anda terapkan dengan menggunakan pendekatan ini.

Epik

Konfigurasikan akun di AWS Organizations

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan semua fitur di akun manajemen.	Aktifkan semua fitur di akun manajemen untuk organisasi Anda dengan mengikuti petunjuk dalam dokumentasi AWS Organizations.	Administrator AWS, Administrator platform
Buat CI/CD akun.	Di AWS Organizations, di organisasi Anda, buat CI/CD akun khusus, dan tetapkan tim untuk memiliki dan mengontrol akses ke akun.	Administrator AWS
Tambahkan administrator yang didelegasikan.	Di akun manajemen, daftarkan CI/CD akun yang Anda buat di langkah sebelumnya sebagai administrator kumpulan tumpukan yang didelegasikan. Untuk petunjuk, lihat CloudFormation dokumentasi AWS.	Administrator AWS, Administrator platform

Buat repositori aplikasi dan pipeline CI/CD

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode.	1. Kloning repositori kode yang disediakan dengan pola ini ke komputer Anda:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` git clone https://github.com/aws-samples/automated-code-pipeline-stackset-deployment.git ``` 2. Tinjau file readme untuk memahami struktur direktori dan detail lainnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat topik SNS.	Anda dapat menggunakan `sns-template.yaml` template yang disediakan di GitHub repositori untuk membuat topik SNS dan mengonfigurasi langganan untuk permintaan persetujuan. 1. Di konsol AWS, masuk ke CI/CD akun. 2. Buka CloudFormation konsol di at https://console.aws.amazon.com/cloudformation. 3. Buat tumpukan baru dengan sumber daya baru (opsi standar). 4. Untuk Tentukan templat, pilih Unggah file templat, Pilih file, lalu pilih `sns-template.yaml` file dari `templates` folder GitHub repositori kloning. Pilih Berikutnya. 5. Berikan nama tumpukan aplikasi yang berarti. 6. Tentukan awalan untuk sumber daya. 7. Pilih Berikutnya, Berikutnya, dan Kirim. 8. Jika tumpukan telah berhasil dibuat, pilih tab Output, dan catat Amazon	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Resource Names (ARNs) topik SNS untuk permintaan tarik, lingkungan pengujian , dan lingkungan produksi. Anda akan menggunakan informasi ini dalam langkah selanjutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk CI/CD komponen.	Anda dapat menggunakan `cicd-role-template.yaml` template yang disediakan di GitHub repositori untuk membuat peran dan kebijakan IAM yang diperlukan oleh komponen. CI/CD 1. Di konsol AWS, masuk ke CI/CD akun. 2. Buka CloudFormation konsol di at https://console.aws.amazon.com/cloudformation. 3. Buat tumpukan baru dengan sumber daya baru (opsi standar). 4. Untuk Tentukan templat, pilih Unggah file templat, Pilih file, lalu pilih `cicd-role-template.yaml` file dari `templates` folder GitHub repositori kloning. Pilih Berikutnya. 5. Berikan nama tumpukan aplikasi yang berarti. 6. Masukkan nilai untuk parameter berikut: • ARN untuk kebijakan batas izin. Anda dapat memperoleh ARN ini dari bagian Detail kebijakan kebijakan kebijakan batas izin Anda di konsol IAM.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• ARN untuk topik persetujuan produksi SNS yang Anda catat sebelumnya. • ARN untuk topik persetujuan tes SNS yang Anda catat sebelumnya. • Awalan untuk sumber daya yang dibuat oleh template. 7. Pilih Berikutnya, Berikutnya, dan Kirim. 8. Ketika tumpukan telah berhasil dibuat, pilih tab Output, dan catat peran IAM yang dibuat. ARNs Anda akan menggunakan informasi ini dalam langkah selanjutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodeCommit repositori dan pipeline kode untuk aplikasi Anda.	Anda dapat menggunakan `cicd-pipeline-template.yaml` template yang disediakan di GitHub repositori untuk membuat CodeCommit repositori dan pipeline kode untuk aplikasi Anda. 1. Di konsol AWS, masuk ke CI/CD akun. 2. Buka CloudFormation konsol di at https://console.aws.amazon.com/cloudformation. 3. Buat tumpukan baru dengan sumber daya baru (opsi standar). 4. Untuk Tentukan templat, pilih Unggah file templat, Pilih file, lalu pilih `cicd-pipeline-template.yaml` file dari `templates` folder GitHub repositori kloning. Pilih Berikutnya. 5. Berikan nama tumpukan aplikasi yang berarti. 6. Masukkan nilai untuk parameter berikut: • AppRepositoryName—Nama CodeCommit repositori yang akan dibuat untuk aplikasi.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• AppRepositoryDescription— Penjelasan singkat tentang CodeCommit repositori yang akan dibuat untuk aplikasi. • ApplicationName— Nama aplikasi Anda. String ini digunakan sebagai nama CodeCommit repositori dan sebagai awalan dari pipeline. CI/CD • CloudWatchEventRoleARN — ARN dari peran CloudWatch acara dari tugas sebelumnya. • CodeBuildProjectRoleARN — ARN peran CodeBuild proyek dari tugas sebelumnya. • CodePipelineRoleARN — ARN CodePipeline peran dari tugas sebelumnya. • DeploymentConfigBucket— Nama bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) tempat file konfigurasi deployment dan file skrip.zip akan disimpan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• DeploymentConfigKey— Jalur dan nama file.zip (kunci Amazon S3). • PRApprovalSNSARN - ARN untuk topik SNS untuk pemberitahuan permintaan tarik. • ProdApprovalSNSARN — ARN untuk topik SNS untuk persetujuan produksi. • TESTApprovalSNSARN - ARN untuk topik SNS untuk persetujuan tes. • TemplateBucket— Nama bucket S3 di CI/CD akun tempat template pembuatan CI/CD pipeline akan disimpan. 7. Pilih Berikutnya, Berikutnya, dan Kirim. 8. Ketika tumpukan berhasil diselesaikan, ia menciptakan CodeCommit repositori yang memiliki nama yang ditentukan dan struktur direktori default, file konfigurasi penerapan, skrip, dan pipa kode untuk repositori.	

Menyebarkan set tumpukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori aplikasi.	Template CI/CD pipeline yang Anda gunakan sebelumnya membuat contoh repositor i aplikasi dan pipeline kode. Untuk mengkloning dan memverifikasi repositori:
	1. Masuk ke CI/CD akun.
	2. Temukan repositori aplikasi dan CI/CD pipeline yang Anda buat di epik sebelumnya.
	3. Salin URL untuk repositor i dan gunakan perintah git clone untuk mengkloning repositori pada mesin lokal Anda.
	4. Verifikasi bahwa struktur direktori dan file cocok dengan yang berikut:
	`root`
	`|- deploy_configs`
	`|  |- deploymen t_config.json`
	`|- parameters`
	`|  |- template- parameter-dev.json`
	`|  |- template- parameter-test.json`
	`|  |- template- parameter-prod.json`
	`|- templates`
	`|  |- template. yml`
		Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`|- buildspec.yml` di mana `deploy_configs` folder berisi file konfigurasi penyebaran, dan `parameters` folder `templates` dan menyertakan file default yang akan Anda ganti dengan file CloudFormation template dan parameter Anda sendiri. **Important** Jangan menyesuaikan struktur folder. 5. Buat cabang fitur.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan artefak aplikasi.	Perbarui repositori aplikasi dengan menggunakan template. CloudFormation **Note** Solusi ini mendukung penyebaran hanya satu CloudFormation template. 1. Buat CloudFormation template Anda untuk menerapkan perubahan kode aplikasi Anda, dan beri nama. `<application-name>.yaml` 2. Ganti `template.yml` file di `templates` folder repositori aplikasi dengan CloudFormation template yang Anda buat di langkah 1. 3. Siapkan file parameter untuk setiap lingkungan (pengembangan, pengujian, dan produksi). 4. Beri nama file parameter dengan menggunakan format`<cloudformation-template-name>-parameter-<env`	Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ironment-name>.jso n` . 5. Ganti file parameter default di `parameters` folder dengan file Anda dari langkah 4.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file konfigurasi penyebaran.	Perbarui `deployment_config.json` file: 1. Di repositori aplikasi, arahkan ke folder. `deploy_configs` 2. Buka file`deployment_config.json` : `{ "deployment_action": "<deploy/delete>", "stack_set_name": "<stack set name>", "stack_set_desciption": "<stack set description>", "deployment_targets": { "dev": { "org_units": ["list of OUs"], "regions": ["list of regions"], "filter_accounts": ["list of accounts"], "filter_type":`	Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"<DIFFERENCE/INTERSECTION/UNION>"
	},
	"test": {
	"org_units": ["list of OUs"],
	"regions": ["list of regions"],
	"filter_accounts": ["list of accounts"],
	"filter_type": "<DIFFERENCE/INTERSECTION/UNION>"
	},
	"prod": {
	"org_units": ["list of OUs"],
	"regions": ["list of regions"],
	"filter_accounts": ["list of accounts"],</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"filter_type": "<DIFFERENCE/INTERSECTION/UNION>"
	}
	},
	"cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
	"auto_deployement": "<True/False>",
	"retain_stacks_on_account_removal": "<True/False>",
	"region_deployment_concurrency": "<SEQUENTIAL/PARALLEL>"
	}</pre> 3. Perbarui nilai untuk tindakan penerapan, nama set tumpukan, deskripsi kumpulan tumpukan, dan target penerapan.
	
	Misalnya, Anda dapat mengatur `deployment_action` `delete` untuk menghapus seluruh kumpulan tumpukan dan instance tumpukan yang terkait. Gunakan `deploy`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk membuat kumpulan tumpukan baru, memperbarui kumpulan tumpukan yang ada, atau untuk menambah atau menghapus instance tumpukan untuk tambahan OUs atau Wilayah. Untuk contoh lainnya, lihat bagian Informasi tambahan. Pola ini membuat kumpulan tumpukan individual untuk setiap lingkungan dengan menambahkan nama lingkungan ke nama set tumpukan yang Anda berikan dalam file konfigurasi penerapan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Komit perubahan dan terapkan set tumpukan.	Komit perubahan yang Anda tentukan dalam template aplikasi Anda, dan gabungkan serta terapkan kumpulan tumpukan ke beberapa lingkungan tahap demi tahap: 1. Simpan semua file Anda dan komit perubahan ke cabang fitur repositori aplikasi lokal Anda. 2. Dorong cabang fitur ke repositori jarak jauh. 3. Buat permintaan tarik untuk menggabungkan perubahan ke cabang utama. Ketika permintaan tarik telah disetujui dan perubahan telah digabungkan ke cabang utama, CI/CD pipeline akan dimulai. 4. Jika tahap Dev-Deployment telah berhasil diselesaikan, periksa CloudFormation konsol,, tab StackSets Service-Managed. Anda akan melihat set tumpukan baru dengan akhiran`dev`. 5. Periksa CodeBuild log untuk tahap penyebaran Dev untuk masalah apa pun.	Pengembang aplikasi, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Terapkan set tumpukan ke lingkungan pengujian dan produksi dengan meminta pemberi persetujuan Anda untuk menyetujui penerapan untuk tahapan tersebut dan mengulangi langkah 5 dan 6. Set tumpukan untuk lingkungan pengujian dan produksi memiliki sufiks `test` dan. `prod`	

Pemecahan masalah

Isu	Solusi
Penerapan gagal dengan pengecualian: Ubah Nama file Parameter Template sebagai -parameter- .json dengan, nama default tidak diperbolehkan <application name><evn>	File parameter CloudFormation template harus mengikuti konvensi penamaan yang ditentukan. Perbarui nama file parameter dan coba lagi.
Penerapan gagal dengan pengecualian: Mengubah Nama CloudFormation Template sebagai.yl, default template.yml/template.yaml tidak diperbolehkan <application name>	Nama CloudFormation template harus mengikuti konvensi penamaan yang ditentukan. Perbarui nama file dan coba lagi.
Penerapan gagal dengan pengecualian: Tidak ada CloudFormation Template yang valid dan File Parameternya yang ditemukan untuk lingkungan {nama lingkungan}	Periksa konvensi penamaan file untuk CloudFormation templat dan file parameternya untuk lingkungan yang ditentukan.

Isu	Solusi
Penerapan gagal dengan pengecualian: Tindakan penerapan tidak valid yang disediaka n dalam file konfigurasi penerapan. Opsi yang valid adalah 'menyebarkan' dan 'hapus'.	Anda menetapkan nilai yang tidak valid untuk `deployment_action` parameter dalam file konfigurasi deployment. Parameter memiliki dua nilai yang valid: `deploy` dan`delete`. Gunakan `deploy` untuk membuat dan memperbarui kumpulan tumpukan dan instance tumpukan terkait. Gunakan `delete` hanya ketika Anda ingin menghapus seluruh kumpulan tumpukan dan instance tumpukan terkait.

Sumber daya terkait

- GitHub automated-code-pipeline-stackset-repositori penyebaran
- Mengaktifkan semua fitur di organisasi Anda (dokumentasi AWS Organizations)
- Daftarkan administrator yang didelegasikan (CloudFormation dokumentasi AWS)
- Target tingkat akun untuk Set Stack yang dikelola layanan (dokumentasi CloudFormation AWS)

Informasi tambahan

Diagram alir

Diagram alir berikut menggambarkan kontrol aliran dan hierarki panggilan API yang diimplementasikan oleh skrip khusus untuk mengotomatiskan penerapan set tumpukan.

Contoh file konfigurasi penyebaran

Membuat set tumpukan baru

File konfigurasi penerapan berikut membuat kumpulan tumpukan baru yang disebut `sample-stack-set` di Wilayah AWS `us-east-1` dalam tiga OUs.

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
```

```
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                            "dev": {
                                        "org_units": ["dev-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                           },
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
 }
```

Menyebarkan tumpukan yang ada disetel ke OU lain

Jika Anda menerapkan konfigurasi yang ditunjukkan pada contoh sebelumnya dan Anda ingin menerapkan kumpulan tumpukan ke OU tambahan yang dipanggil `dev-org-unit-2` di lingkungan pengembangan, file konfigurasi penerapan mungkin terlihat seperti berikut.

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                            "dev": {
                                        "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
```

```
                                    "filter_type": ""
                                },
                            "test": {
                                    "org_units": ["test-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": [],
                                    "filter_type": ""
                                },
                            "prod": {
                                    "org_units": ["prod-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": [],
                                    "filter_type": ""
                                }
                        },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}
```

Menerapkan tumpukan yang ada yang disetel ke Wilayah AWS lain

Jika Anda menerapkan konfigurasi yang ditunjukkan pada contoh sebelumnya dan Anda ingin menerapkan kumpulan tumpukan ke AWS Region (`us-east-2`) tambahan di lingkungan pengembangan untuk dua OUs (`dev-org-unit-1`dan`dev-org-unit-2`), file konfigurasi penerapan mungkin terlihat seperti berikut.

> **Note**
>
> Sumber daya dalam CloudFormation template harus valid dan spesifik Wilayah.

```
{
    "deployment_action": "deploy",
    "stack_set_name": "sample-stack-set",
    "stack_set_desciption": "this is a sample stack set",
   "deployment_targets": {
                            "dev": {
                                    "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                    "regions": ["us-east-1", "us-east-2"],
```

```
                                    "filter_accounts": [],
                                    "filter_type": ""
                                },
                            "test": {
                                    "org_units": ["test-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": [],
                                    "filter_type": ""
                                },
                            "prod": {
                                    "org_units": ["prod-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": [],
                                    "filter_type": ""
                                }
                        },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}
```

Menghapus instans tumpukan dari Wilayah OU atau AWS

Katakanlah konfigurasi penerapan yang ditunjukkan pada contoh sebelumnya telah diterapkan. File konfigurasi berikut menghapus instance tumpukan dari kedua Wilayah OU`dev-org-unit-2`.

```
{
    "deployment_action": "deploy",
    "stack_set_name": "sample-stack-set",
    "stack_set_desciption": "this is a sample stack set",
   "deployment_targets": {
                            "dev": {
                                    "org_units": ["dev-org-unit-1"],
                                    "regions": ["us-east-1", "us-east-2"],
                                    "filter_accounts": [],
                                    "filter_type": ""
                                },
                            "test": {
                                    "org_units": ["test-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": [],
                                    "filter_type": ""
```

```
                                    },
                                "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                            },
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
 }
```

File konfigurasi berikut menghapus instance tumpukan dari Wilayah AWS `us-east-1` untuk keduanya OUs di lingkungan pengembangan.

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                                "dev": {
                                        "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                        "regions": ["us-east-2"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                                "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                                "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                            },
```

```
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
 }
```

Menghapus seluruh set tumpukan

File konfigurasi penerapan berikut menghapus seluruh kumpulan tumpukan dan semua instance tumpukan terkait.

```
{
     "deployment_action": "delete",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                            "dev": {
                                        "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                        "regions": ["us-east-2"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                           },
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
 }
```

Mengecualikan akun dari penerapan

File konfigurasi penyebaran berikut mengecualikan akun111122223333, yang merupakan bagian dari OUdev-org-unit-1, dari penerapan.

```
{
    "deployment_action": "deploy",
    "stack_set_name": "sample-stack-set",
    "stack_set_desciption": "this is a sample stack set",
   "deployment_targets": {
                           "dev": {
                                      "org_units": ["dev-org-unit-1"],
                                      "regions": ["us-east-1"],
                                      "filter_accounts": ["111122223333"],
                                      "filter_type": "DIFFERENCE"
                                   },
                           "test": {
                                      "org_units": ["test-org-unit-1"],
                                      "regions": ["us-east-1"],
                                      "filter_accounts": [],
                                      "filter_type": ""
                                   },
                           "prod": {
                                      "org_units": ["prod-org-unit-1"],
                                      "regions": ["us-east-1"],
                                      "filter_accounts": [],
                                      "filter_type": ""
                                   }
                         },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}
```

Menyebarkan aplikasi ke subset akun di OU

File konfigurasi deployment berikut menyebarkan aplikasi ke hanya tiga akun (111122223333,444455556666, dan777788889999) di OU. dev-org-unit-1

```
{
    "deployment_action": "deploy",
    "stack_set_name": "sample-stack-set",
    "stack_set_desciption": "this is a sample stack set",
   "deployment_targets": {
```

```
                            "dev": {
                                        "org_units": ["dev-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": ["111122223333",
"444455556666", "777788889999"],
                                        "filter_type": "INTERSECTION"
                                    },
                            "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                           },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}
```

Secara otomatis melampirkan kebijakan terkelola AWS untuk Systems Manager ke profil EC2 instans menggunakan Cloud Custodian dan AWS CDK

Ali Asfour dan Aaron Lennon, Amazon Web Services

Ringkasan

Anda dapat mengintegrasikan instans Amazon Elastic Compute Cloud (Amazon EC2) dengan AWS Systems Manager untuk mengotomatiskan tugas operasional serta memberikan lebih banyak visibilitas dan kontrol. Untuk berintegrasi dengan Systems Manager, EC2 instans harus memiliki Agen AWS Systems Manager (Agen SSM) yang diinstal dan kebijakan `AmazonSSMManagedInstanceCore` AWS Identity and Access Management (IAM) yang terpasang pada profil instans mereka.

Namun, jika Anda ingin memastikan bahwa semua profil EC2 instans memiliki `AmazonSSMManagedInstanceCore` kebijakan yang dilampirkan, Anda dapat menghadapi tantangan dalam memperbarui EC2 instans baru yang tidak memiliki profil EC2 instans atau instance yang memiliki profil instans tetapi tidak memiliki kebijakan. `AmazonSSMManagedInstanceCore` Mungkin juga sulit untuk menambahkan kebijakan ini di beberapa akun Amazon Web Services (AWS) dan Wilayah AWS.

Pola ini membantu mengatasi tantangan ini dengan menerapkan tiga kebijakan Cloud Custodian di akun AWS Anda:

- Kebijakan Cloud Custodian pertama memeriksa EC2 instans yang ada yang memiliki profil instans tetapi tidak memiliki kebijakan. `AmazonSSMManagedInstanceCore` `AmazonSSMManagedInstanceCore`Kebijakan tersebut kemudian dilampirkan.
- Kebijakan Cloud Custodian kedua memeriksa instans yang ada tanpa profil EC2 instans dan menambahkan profil instans default yang memiliki kebijakan yang dilampirkan. `AmazonSSMManagedInstanceCore`
- Kebijakan Cloud Custodian ketiga membuat fungsi AWS Lambda di akun Anda untuk memantau pembuatan EC2 instans dan profil instans. Ini memastikan bahwa `AmazonSSMManagedInstanceCore` kebijakan dilampirkan secara otomatis saat EC2 instance dibuat.

Pola ini menggunakan DevOps alat AWS untuk mencapai penerapan kebijakan Cloud Custodian yang berkelanjutan dan berskala besar ke lingkungan multi-akun, tanpa menyediakan lingkungan komputasi terpisah.

Prasyarat dan batasan

Prasyarat

- Dua atau lebih akun AWS aktif. Satu akun adalah akun keamanan dan yang lainnya adalah akun anggota.
- Izin untuk menyediakan sumber daya AWS di akun keamanan. Pola ini menggunakan izin administrator, tetapi Anda harus memberikan izin sesuai dengan persyaratan dan kebijakan organisasi Anda.
- Kemampuan untuk mengambil peran IAM dari akun keamanan ke akun anggota dan membuat peran IAM yang diperlukan. Untuk informasi selengkapnya tentang hal ini, lihat Mendelegasikan akses di seluruh akun AWS menggunakan peran IAM dalam dokumentasi IAM.
- **Important**

  AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. Untuk tujuan pengujian, Anda dapat mengonfigurasi AWS CLI dengan menggunakan `aws configure` perintah atau menyetel variabel lingkungan. : Ini tidak disarankan untuk lingkungan produksi dan kami menyarankan agar akun ini hanya diberikan akses hak istimewa paling sedikit. Untuk informasi lebih lanjut tentang ini, lihat Berikan hak istimewa paling sedikit dalam dokumentasi IAM.
- `devops-cdk-cloudcustodian.zip`File (terlampir), diunduh ke komputer lokal Anda.
- Keakraban dengan Python.
- Alat yang diperlukan (Node.js, AWS Cloud Development Kit (AWS CDK), dan Git), diinstal dan dikonfigurasi. Anda dapat menggunakan `install-prerequisites.sh` file dalam `devops-cdk-cloudcustodian.zip` file untuk menginstal alat-alat ini. Pastikan Anda menjalankan file ini dengan hak akses root.

Batasan

- Meskipun pola ini dapat digunakan dalam lingkungan produksi, pastikan bahwa semua peran dan kebijakan IAM memenuhi persyaratan dan kebijakan organisasi Anda.

Versi Package

- Cloud Custodian versi 0.9 atau yang lebih baru
- TypeScript versi 3.9.7 atau yang lebih baru
- Node.js versi 14.15.4 atau yang lebih baru
- npmversi 7.6.1 atau yang lebih baru
- AWS CDK versi 1.96.0 atau yang lebih baru

Arsitektur

Diagram menunjukkan alur kerja berikut:

1. Kebijakan Cloud Custodian didorong ke CodeCommit repositori AWS di akun keamanan. Aturan CloudWatch Acara Amazon secara otomatis memulai CodePipeline pipeline AWS.
2. Pipeline mengambil kode terbaru dari CodeCommit dan mengirimkannya ke bagian integrasi berkelanjutan dari pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang ditangani oleh AWS. CodeBuild
3. CodeBuild melakukan DevSecOps tindakan lengkap, termasuk validasi sintaks kebijakan pada kebijakan Cloud Custodian, dan menjalankan kebijakan ini dalam `--dryrun` mode untuk memeriksa sumber daya mana yang diidentifikasi.
4. Jika tidak ada kesalahan, tugas berikutnya memberi tahu administrator untuk meninjau perubahan dan menyetujui penyebaran ke akun anggota.

Tumpukan teknologi

- AWS CDK
- CodeBuild
- CodeCommit
- CodePipeline
- IAM
- Penjaga Awan

Otomatisasi dan skala

Modul pipeline AWS CDK menyediakan CI/CD pipeline yang digunakan CodePipeline untuk mengatur pembuatan dan pengujian kode sumber CodeBuild, selain penerapan sumber daya AWS dengan tumpukan AWS. CloudFormation Anda dapat menggunakan pola ini untuk semua akun anggota dan Wilayah di organisasi Anda. Anda juga dapat memperluas `Roles creation` tumpukan untuk menerapkan peran IAM lainnya di akun anggota Anda.

Alat

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak untuk mendefinisikan infrastruktur cloud dalam kode dan menyediakannya melalui AWS. CloudFormation
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang memungkinkan Anda berinteraksi dengan layanan AWS menggunakan perintah di shell baris perintah Anda.
- AWS CodeBuild adalah layanan build yang dikelola sepenuhnya di cloud.
- AWS CodeCommit adalah layanan kontrol versi yang dapat Anda gunakan untuk menyimpan dan mengelola aset secara pribadi.
- AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perangkat lunak Anda.
- AWS Identity and Access Management adalah layanan web yang membantu Anda mengontrol akses ke sumber daya AWS dengan aman.
- Cloud Custodian adalah alat yang menyatukan lusinan alat dan skrip yang digunakan sebagian besar organisasi untuk mengelola akun cloud publik mereka menjadi satu alat sumber terbuka.
- Node.js adalah JavaScript runtime yang dibangun di mesin V8 JavaScript Google Chrome.

Kode

Untuk daftar rinci modul, fungsi akun, file, dan perintah penerapan yang digunakan dalam pola ini, lihat `README` file dalam `devops-cdk-cloudcustodian.zip` file (terlampir).

Epik

Siapkan pipeline dengan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan CodeCommit repositori.	1. Buka zip `devops-cdk-cloudcustodian.zip` file (terlampir) di direktori kerja di komputer lokal Anda. 2. Masuk ke AWS Management Console untuk akun keamanan Anda, buka CodeCommit konsol, lalu buat `devops-cdk-cloudcustodian` repositori baru. 3. Ubah ke direktori proyek dan atur CodeCommit repositori sebagai asal, komit perubahan, lalu dorong ke cabang asal dengan menjalankan perintah berikut: • `cd devops-cdk-cloudcustodian` • `git init --initial-branch=main` • `git add . git commit -m 'initial commit'` • `git remote add origin https://git-codecommit.us-east-1.ama`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`zonaws.com/v1/devops-cdk-cloudcustodian` • `git push origin main` Untuk informasi selengkapnya tentang ini, lihat Membuat CodeCommit repositori di dokumentasi CodeCommit AWS.	
Instal alat yang diperlukan.	Gunakan `install-prerequisites.sh` file untuk menginstal semua alat yang diperlukan di Amazon Linux. Ini tidak termasuk AWS CLI karena sudah diinstal sebelumnya. Untuk informasi selengkapnya tentang ini, lihat bagian Prasyarat Memulai AWS CDK dalam dokumentasi AWS CDK.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal paket AWS CDK yang diperlukan.	1. Siapkan lingkungan virtual Anda dengan menjalankan perintah berikut di AWS CLI: `$ python3 -m venv .env` 2. Aktifkan lingkungan virtual Anda dengan menjalankan perintah berikut: `$ source .env/bin/activate` 3. Setelah lingkungan virtual diaktifkan, instal dependensi yang diperlukan dengan menjalankan perintah berikut: `$ pip install -r requirements.txt` 4. Untuk menambahkan dependensi tambahan (misalnya, library AWS CDK lainnya), tambahkan dependensi tersebut ke `requirements.txt` file, lalu jalankan perintah berikut: `pip install -r requirements.txt` Paket-paket berikut diperlukan oleh AWS CDK dan disertakan dalam `requirements.txt` file: • `aws-cdk.aws-cloudwatch`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `aws-cdk.aws-codebuild` • `aws-cdk.aws-codecommit` • `aws-cdk.aws-codedeploy` • `aws-cdk.aws-codepipeline` • `aws-cdk.aws-codepipeline-actions` • `aws-cdk.aws-events` • `aws-cdk.aws-events-targets` • `aws-cdk.aws-iam` • `aws-cdk.aws-logs` • `aws-cdk.aws-s3` • `aws-cdk.aws-sns` • `aws-cdk.aws-sns-subscriptions` • `aws-cdk.aws-sqs` • `aws-cdk.core`	

Konfigurasikan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui variabel yang diperlukan.	Buka `vars.py` file di folder root CodeCommit repositori Anda dan perbarui variabel berikut:	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Perbarui `var_deploy_region = ‘us-east-1’` dengan Wilayah AWS tempat Anda ingin pipeline diterapkan. • Perbarui `var_codecommit_repo_name = "cdk-cloudcustodian"` dengan nama CodeCommit repositori Anda. • Perbarui `var_codecommit_branch_name = "main"` dengan nama CodeCommit cabang. • Perbarui `var_adminEmail=notifyadmin@email.com’` dengan alamat email untuk administrator yang menyetujui perubahan. • Perbarui `var_slackWebHookUrl = https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXX` 'dengan webhook Slack yang digunakan untuk mengirim pemberitahuan Cloud Custodian saat perubahan dilakukan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Perbarui `var_orgId = ‘o-yyyyyyyyyy’` dengan ID organisasi Anda. • Perbarui `security_account = ‘123456789011’` dengan ID akun AWS untuk akun tempat pipeline digunakan. • Perbarui `member_accounts = [‘111111111111’,’111111111112’,’111111111113’]` dengan akun anggota tempat Anda ingin mem-bootstrap tumpukan AWS CDK dan menerapkan peran IAM yang diperlukan. • Setel `cdk_bootstrap_member_accounts = True` ke `True` jika Anda ingin pipeline secara otomatis mem-bootstrap AWS CDK ke akun anggota Anda. Jika disetel ke `True` ini juga memerlukan nama peran IAM yang ada di akun anggota yang dapat diasumsikan dari akun keamanan. Peran IAM ini juga harus memiliki izin yang diperlukan untuk mem-bootstrap AWS CDK.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• **Note** Perbarui `cdk_bootstrap_role = 'AWSControlTowerExecution'` dengan peran IAM yang ada di akun anggota yang dapat diasumsikan dari akun keamanan. Peran ini juga harus mengizinkan untuk mem-bootstrap AWS CDK. : Ini hanya berlaku jika `cdk_bootstrap_member_accounts` disetel ke`True`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui file account.y mldengan informasi akun anggota.	Untuk menjalankan alat Cloud Custodian c7n-org terhadap beberapa akun, Anda harus menempatkan file `accounts.yml` konfigurasi di root repositori. Berikut ini adalah contoh file konfigurasi Cloud Custodian untuk AWS:
	`accounts:`
	`- account_id: '123123123123'`
	`name: account-1`
	`regions:`
	`- us-east-1`
	`- us-west-2`
	`role: arn:aws:iam::123123123123:role/CloudCustodian`
	`vars:`
	`charge_code: xyz`
	`tags:`
	`- type:prod`
	`- division:some division`
	`- partition:us`
	`- scope:pci`
		Developer

Bootstrap akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Boostrap akun keamanan.	Bootstrap `deploy_account` dengan `cloudcustodian_stack` aplikasi	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan menjalankan perintah berikut: `cdk bootstrap -a 'python3 cloudcustodian/cloudcustodian_stack.py`	
Opsi 1 - Secara otomatis bootstrap akun anggota.	Jika `cdk_bootstrap_member_accounts` variabel diatur ke `True` dalam `vars.py` file, akun yang ditentukan dalam `member_accounts` variabel secara otomatis di-bootstrap oleh pipeline. Jika diperlukan, Anda dapat memperbarui `*cdk_bootstrap_role*` dengan peran IAM yang dapat Anda asumsikan dari akun keamanan dan yang memiliki izin yang diperlukan untuk mem-bootstrap AWS CDK. Akun baru yang ditambahkan ke `member_accounts` variabel secara otomatis di-bootstrap oleh pipeline sehingga peran yang diperlukan dapat digunakan.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 2 - Bootstrap akun anggota secara manual.	Meskipun kami tidak menyarankan menggunakan pendekatan ini, Anda dapat mengatur nilai `cdk_bootstrap_member_accounts` to `False` dan melakukan langkah ini secara manual dengan menjalankan perintah berikut: `$ cdk bootstrap -a 'python3 cloudcustodian/member_account_roles_stack.py' \ --trust {security_account_id} \ --context assume-role-credentials:writeIamRoleName={role_name} \ --context assume-role-credentials:readIamRoleName={role_name} \ --mode=ForWriting \ --context bootstrap=true \ --cloudformation-execution-policies arn:aws:iam::aws:policy/AdministratorAccess`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important Pastikan Anda memperbarui `{security_account_id}` dan `{role_name}` nilai dengan nama peran IAM yang dapat Anda asumsikan dari akun keamanan dan yang memiliki izin yang diperlukan untuk mem-bootstrap AWS CDK. Anda juga dapat menggunakan pendekatan lain untuk mem-bootstrap akun anggota, misalnya, dengan AWS CloudFormation. Untuk informasi selengkapnya tentang ini, lihat Bootstrapping di dokumentasi AWS CDK.	

Menerapkan tumpukan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM di akun anggota.	Jalankan perintah berikut untuk menyebarkan `member_account_roles_stack` tumpukan dan	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membuat peran IAM di akun anggota: `cdk deploy --all -a 'python3 cloudcustodian/member_account_roles_stack.py' --require-approval never`	
Terapkan tumpukan pipeline Cloud Custodian.	Jalankan perintah berikut untuk membuat `cloudcustodian_stack.py` pipeline Cloud Custodian yang disebarkan ke akun keamanan: `cdk deploy -a 'python3 cloudcustodian/cloudcustodian_stack.py'`	Developer

Sumber daya terkait

- Memulai dengan AWS CDK

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Membuat CI/CD pipeline dan kluster Amazon ECS secara otomatis untuk layanan mikro menggunakan AWS CDK

Varsha Raju, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara membuat integrasi berkelanjutan dan pengiriman berkelanjutan secara otomatis (CI/CD) pipelines and underlying infrastructure for building and deploying microservices on Amazon Elastic Container Service (Amazon ECS). You can use this approach if you want to set up proof-of-concept CI/CD pipelines to show your organization the benefits of CI/CD, microservices, and DevOps. You can also use this approach to create initial CI/CDpipeline yang kemudian dapat Anda sesuaikan atau ubah sesuai dengan kebutuhan organisasi Anda.

Pendekatan pola menciptakan lingkungan produksi dan lingkungan non-produksi yang masing-masing memiliki virtual private cloud (VPC) dan cluster Amazon ECS yang dikonfigurasi untuk berjalan di dua Availability Zone. Lingkungan ini dibagi oleh semua layanan mikro Anda dan Anda kemudian membuat CI/CD pipeline untuk setiap layanan mikro. CI/CD Pipeline ini menarik perubahan dari repositori sumber di AWS CodeCommit, membuat perubahan secara otomatis, lalu menerapkannya ke lingkungan produksi dan non-produksi Anda. Ketika pipa berhasil menyelesaikan semua tahapannya, Anda dapat menggunakannya URLs untuk mengakses layanan mikro di lingkungan produksi dan non-produksi.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif.
- Bucket Amazon Simple Storage Service (Amazon S3) yang sudah ada yang berisi file (`starter-code.zip`terlampir).
- AWS Cloud Development Kit (AWS CDK), diinstal dan dikonfigurasi di akun Anda. Untuk informasi selengkapnya tentang ini, lihat Memulai AWS CDK dalam dokumentasi AWS CDK.
- Python 3 dan`pip`, diinstal dan dikonfigurasi. Untuk informasi lebih lanjut tentang ini, lihat dokumentasi Python.
- Keakraban dengan AWS CDK, AWS CodeBuild, CodePipeline AWS, CodeCommit Amazon Elastic Container Registry (Amazon ECR), Amazon ECS, dan AWS Fargate.
- Keakraban dengan Docker.

- Pemahaman tentang CI/CD dan DevOps.

Batasan

- Batas akun AWS umum berlaku. Untuk informasi selengkapnya tentang hal ini, lihat kuota layanan AWS di dokumentasi AWS General Reference.

Versi produk

- Kode diuji menggunakan Node.js versi 16.13.0 dan AWS CDK versi 1.132.0.

Arsitektur

Diagram menunjukkan alur kerja berikut:

1. Pengembang aplikasi melakukan kode ke CodeCommit repositori.
2. Sebuah pipa dimulai.
3. CodeBuild membangun dan mendorong image Docker ke repositori Amazon ECR
4. CodePipeline menyebarkan gambar baru ke layanan Fargate yang ada di kluster Amazon ECS non-produksi.
5. Amazon ECS menarik gambar dari repositori Amazon ECR ke layanan Fargate non-produksi.
6. Pengujian dilakukan dengan menggunakan URL non-produksi.
7. Manajer rilis menyetujui penerapan produksi.
8. CodePipeline menyebarkan gambar baru ke layanan Fargate yang ada di cluster Amazon ECS produksi
9. Amazon ECS menarik gambar dari repositori Amazon ECR ke dalam layanan Fargate produksi.
10. Pengguna produksi mengakses fitur Anda dengan menggunakan URL produksi.

Tumpukan teknologi

- AWS CDK
- CodeBuild
- CodeCommit

- CodePipeline
- Amazon ECR
- Amazon ECS
- Amazon VPC

Otomatisasi dan skala

Anda dapat menggunakan pendekatan pola ini untuk membuat pipeline untuk layanan mikro yang diterapkan di tumpukan AWS bersama. CloudFormation Otomatisasi dapat membuat lebih dari satu cluster Amazon ECS di setiap VPC dan juga membuat pipeline untuk layanan mikro yang digunakan di cluster Amazon ECS bersama. Namun, ini mengharuskan Anda memberikan informasi sumber daya baru sebagai input ke tumpukan pipeline.

Alat

- AWS CDK — AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak untuk mendefinisikan infrastruktur cloud dalam kode dan menyediakannya melalui AWS. CloudFormation
- AWS CodeBuild — AWS CodeBuild adalah layanan build yang dikelola sepenuhnya di cloud. CodeBuild mengompilasi kode sumber Anda, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit — AWS CodeCommit adalah layanan kontrol versi yang memungkinkan Anda menyimpan dan mengelola repositori Git secara pribadi di AWS Cloud. CodeCommit menghilangkan kebutuhan bagi Anda untuk mengelola sistem kontrol sumber Anda sendiri atau khawatir tentang penskalaan infrastrukturnya.
- AWS CodePipeline — AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perangkat lunak Anda. Anda dapat dengan cepat memodelkan dan mengonfigurasi berbagai tahapan proses rilis perangkat lunak. CodePipeline mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak Anda secara terus menerus.
- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat yang digunakan untuk menjalankan, menghentikan, dan mengelola kontainer di cluster. Anda dapat menjalankan tugas dan layanan Anda pada infrastruktur tanpa server yang dikelola oleh AWS Fargate. Atau, untuk kontrol lebih lanjut atas

infrastruktur, Anda dapat menjalankan tugas dan layanan Anda di klaster instans Amazon Elastic Compute Cloud (Amazon EC2) yang Anda kelola.

- Docker — Docker membantu pengembang untuk mengemas, mengirim, dan menjalankan aplikasi apa pun sebagai wadah yang ringan, portabel, dan mandiri.

Kode

Kode untuk pola ini tersedia di `starter-code.zip` file `cicdstarter.zip` dan (terlampir).

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan direktori kerja untuk AWS CDK.	1. Buat direktori bernama `cicdproject` di mesin lokal Anda. 2. Unduh `cicdstarter.zip` file (terlampir) ke dalam `cicdproject` direktori dan unzip. Ini membuat folder bernama`cicdstarter` . 3. Jalankan perintah `cd <user-home>/cicdproject/cicdstarter` . 4. Siapkan lingkungan virtual Python dengan menjalankan perintah. `python3 -m venv .venv` 5. Jalankan perintah `source ./.venv/bin/activate` . 6. Konfigurasikan lingkungan AWS Anda dengan	AWS DevOps, infrastruktur Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menjalankan `aws configure` perintah atau dengan menggunakan variabel lingkungan berikut: • `AWS_ACCESS_KEY_ID` • `AWS_SECRET_ACCESS_KEY` • `AWS_DEFAULT_REGION`	

Buat infrastruktur bersama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat infrastruktur bersama.	1. Di direktori kerja Anda, jalankan `cd cicdvpcecs` perintah. 2. Jalankan `pip3 install -r requirements.txt` perintah untuk menginstal semua dependensi Python yang diperlukan 3. Jalankan `cdk bootstrap command` untuk mengatur lingkungan AWS untuk AWS CDK. 4. Jalankan perintah `cdk synth --context aws_account=<aws_account_ID> --context aws_region=<aws-region> .`	AWS DevOps, infrastruktur Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Jalankan perintah `cdk deploy --context aws_account=<aws_account_ID> --context aws_region=<aws-region>` . 6. CloudFormation Tumpukan AWS membuat infrastruktur berikut: • VPC non-produksi bernama `cicd-vpc-ecs/cicd-vpc-nonprod` • VPC produksi bernama `cicd-vpc-ecs/cicd-vpc-prod` • Cluster Amazon ECS non-produksi bernama `cicd-ecs-nonprod` • Cluster Amazon ECS produksi bernama `cicd-ecs-prod`	
Pantau CloudFormation tumpukan AWS.	1. Masuk ke AWS Management Console, buka CloudFormation konsol AWS, lalu pilih `cicd-vpc-ecs` tumpukan dari daftar. 2. Di panel detail tumpukan, pilih tab Peristiwa dan pantau kemajuan pembuatan tumpukan Anda.	AWS DevOps, infrastruktur Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji CloudFormation tumpukan AWS.	1. Setelah CloudFormation tumpukan `cicd-vpc-ecs` AWS dibuat, pastikan bahwa `cicd-vpc-ecs/cicd-vpc-nonprod` dan `cicd-vpc-ecs/cicd-vpc-prod` VPCs dibuat. 2. Pastikan bahwa cluster ECS `cicd-ecs-nonprod` dan `cicd-ecs-prod` Amazon dibuat. **Important** Pastikan Anda merekam IDs untuk keduanya VPCs dan grup keamanan IDs untuk grup keamanan default di keduanya VPCs.	AWS DevOps, infrastruktur Cloud

Buat CI/CD pipeline untuk microservice

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat infrastruktur untuk layanan mikro.	1. Beri nama layanan mikro Anda. Misalnya, pola ini digunakan `myservice1` sebagai nama microservice.	AWS DevOps, infrastruktur Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di direktori kerja Anda jalankan `cd <working-directory>/cdkpipeline` perintah. 3. Jalankan perintah `pip3 install -r requirements.txt` . 4. Jalankan `cdk synth` perintah lengkap yang tersedia di bagian Informasi tambahan dari pola ini. 5. Jalankan `cdk deploy` perintah lengkap yang tersedia di bagian Informasi tambahan dari pola ini. **Note** Anda juga dapat memberikan nilai untuk kedua perintah dengan menggunakan `cdk.json` file di direktori.	
Pantau CloudFormation tumpukan AWS.	Buka CloudFormation konsol AWS dan pantau kemajuan `myservice1-cicd-stack` tumpukan. Akhirnya, status berubah menjadi `CREATE_COMPLETE` .	AWS DevOps, infrastruktur Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji CloudFormation tumpukan AWS.	1. Di CodeCommit konsol AWS, verifikasi bahwa repositori bernama `myservice1` ada dan berisi kode starter. 2. Di CodeBuild konsol AWS, verifikasi bahwa proyek build bernama `myservice1` ada. 3. Di konsol Amazon ECR, verifikasi bahwa repositori Amazon ECR bernama ada. `myservice1` 4. Di konsol Amazon ECS, verifikasi bahwa layanan Fargate `myservice1` bernama ada di klaster Amazon ECS non-produksi dan produksi. 5. Di konsol Amazon Elastic Compute Cloud (Amazon EC2), verifikasi bahwa Application Load Balancer non-produksi dan produksi dibuat. Catat nama DNS dari. ALBs 6. Di CodePipeline konsol AWS, verifikasi bahwa pipeline bernama `myservice1` ada. Itu harus memiliki`Source`,, `BuildDeploy-NonProd` , dan `Deploy-Prod`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tahapan. Pipa juga harus memiliki `in progress` status. 7. Pantau pipa sampai semua tahapan selesai. 8. Menyetujui secara manual untuk produksi. 9. Di jendela browser, masukkan nama DNS dari file. ALBs 10 Aplikasi harus ditampilkan `Hello World` dalam non-produksi dan produksi URLs.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan pipa.	1. Buka CodeCommit repositori yang Anda buat sebelumnya dan buka file. `index.js` 2. Ganti `Hello World` dengan`Hello CI/CD`. 3. Simpan dan komit perubahan ke cabang utama. 4. Verifikasi bahwa pipa dimulai dan bahwa perubahan melewati`Build`,`Deploy-NonProd` , dan `Deploy-Prod` tahapan. 5. Menyetujui produksi secara manual. 6. Baik produksi maupun non-produksi sekarang URLs harus ditampilkan*`Hello CICD`*.	AWS DevOps, infrastruktur Cloud
Ulangi epik ini untuk setiap layanan mikro.	Ulangi tugas dalam epik ini untuk membuat CI/CD pipeline untuk setiap layanan mikro Anda.	AWS DevOps, infrastruktur Cloud

Sumber daya terkait

- Menggunakan Python dengan AWS CDK
- Referensi AWS CDK Python
- Membuat layanan AWS Fargate menggunakan AWS CDK

Informasi tambahan

Perintah **cdk synth**

```
cdk synth --context aws_account=<aws_account_number> --context
 aws_region=<aws_region> --context vpc_nonprod_id=<id_of_non_production
 VPC> --context vpc_prod_id=<id_of_production_VPC> --context
 ecssg_nonprod_id=< default_security_group_id_of_non-production_VPC>
 --context ecssg_prod_id=<default_security_group_id_of_production_VPC>
 --context code_commit_s3_bucket_for_code=<S3 bucket name> --context
 code_commit_s3_object_key_for_code=<Object_key_of_starter_code> --context
 microservice_name=<name_of_microservice>
```

cdk deploy perintah

```
cdk deploy --context aws_account=<aws_account_number> --context
 aws_region=<aws_region> --context vpc_nonprod_id=<id_of_non_production_VPC>
 --context vpc_prod_id=<id_of_production_VPC> --context ecssg_nonprod_id=<
 default_security_group_id_of_non-production_VPC> --context
 ecssg_prod_id=<default_security_group_id_of_production_VPC> --
context code_commit_s3_bucket_for_code=<S3 bucket name> --context
 code_commit_s3_object_key_for_code=<Object_key_of_starter_code> --context
 microservice_name=<name_of_microservice>
```

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat dan dorong gambar Docker ke Amazon ECR menggunakan GitHub Actions dan Terraform

Ruchika Modi, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana Anda dapat membuat GitHub alur kerja yang dapat digunakan kembali untuk membangun Dockerfile Anda dan mendorong gambar yang dihasilkan ke Amazon Elastic Container Registry (Amazon ECR). Pola mengotomatiskan proses pembuatan Dockerfiles Anda dengan menggunakan Terraform dan Actions. GitHub Ini meminimalkan kemungkinan kesalahan manusia dan secara substansional mengurangi waktu penyebaran.

Tindakan GitHub push ke cabang utama GitHub repositori Anda memulai penyebaran sumber daya. Alur kerja membuat repositori Amazon ECR unik berdasarkan kombinasi nama GitHub organisasi dan repositori. Kemudian mendorong gambar Dockerfile ke repositori Amazon ECR.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- GitHub Akun aktif.
- Sebuah GitHub repositori.
- Terraform versi 1 atau yang lebih baru diinstal dan dikonfigurasi.
- Bucket Amazon Simple Storage Service (Amazon S3) untuk backend Terraform.
- Tabel Amazon DynamoDB untuk penguncian dan konsistensi status Terraform. Tabel harus memiliki kunci partisi bernama `LockID` dengan tipe`String`. Jika ini tidak dikonfigurasi, penguncian status akan dinonaktifkan.
- Peran AWS Identity and Access Management (IAM) yang memiliki izin untuk menyiapkan backend Amazon S3 untuk Terraform. Untuk petunjuk konfigurasi, lihat dokumentasi Terraform.

Keterbatasan

Kode yang dapat digunakan kembali ini telah diuji hanya dengan GitHub Tindakan.

Arsitektur

Tumpukan teknologi target

- Repositori Amazon ECR
- GitHub Tindakan
- Terraform

Arsitektur target

Diagram ini menggambarkan hal sebagai berikut:

1. Seorang pengguna menambahkan template Dockerfile dan Terraform ke repositori. GitHub

2. Penambahan ini memulai alur kerja GitHub Actions.

3. Alur kerja memeriksa apakah ada repositori Amazon ECR. Jika tidak, itu membuat repositori berdasarkan nama GitHub organisasi dan repositori.

4. Alur kerja membangun Dockerfile dan mendorong gambar ke repositori Amazon ECR.

Alat

Layanan Amazon

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri kontainer terkelola yang aman, terukur, dan andal.

Alat-alat lainnya

- GitHub Tindakan diintegrasikan ke dalam GitHub platform untuk membantu Anda membuat, berbagi, dan menjalankan alur kerja dalam GitHub repositori Anda. Anda dapat menggunakan GitHub Tindakan untuk mengotomatiskan tugas seperti membangun, menguji, dan menerapkan kode Anda.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola infrastruktur cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di repositori Alur Kerja Tindakan ECR GitHub Docker.

- Saat Anda membuat GitHub Actions, file alur kerja Docker disimpan di `/.github/workflows/` folder repositori ini. Alur kerja untuk solusi ini ada di file workflow.yaml.
- `e2e-test`Folder menyediakan contoh Dockerfile untuk referensi dan pengujian.

Praktik terbaik

- Untuk praktik terbaik dalam menulis Dockerfiles, lihat dokumentasi Docker.
- Gunakan titik akhir VPC untuk Amazon ECR. Titik akhir VPC didukung oleh AWS PrivateLink, teknologi yang memungkinkan Anda mengakses Amazon APIs ECR secara pribadi melalui alamat IP pribadi. Untuk tugas Amazon ECS yang menggunakan tipe peluncuran Fargate, titik akhir VPC memungkinkan tugas untuk menarik gambar pribadi dari Amazon ECR tanpa menetapkan alamat IP publik ke tugas tersebut.

Epik

Siapkan penyedia dan repositori OIDC GitHub

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan OpenID Connect.	Buat penyedia OpenID Connect (OIDC). Anda akan menggunakan penyedia dalam kebijakan kepercayaan untuk peran IAM yang digunakan dalam tindakan ini. Untuk petunjuk, lihat Mengonfigurasi OpenID Connect di Amazon Web Services GitHub dalam dokumentasi.	Administrator AWS, AWS DevOps, AWS Umum
Kloning GitHub repositori.	Kloning repositori Alur Kerja Tindakan ECR GitHub Docker ke folder lokal Anda:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`$git clone https://github.com/aws-samples/docker-ecr-actions-workflow`	

Sesuaikan alur kerja yang GitHub dapat digunakan kembali dan terapkan gambar Docker

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan acara yang memulai alur kerja Docker.	Alur kerja untuk solusi ini ada di workflow.yaml. Skrip ini saat ini dikonfigurasi untuk menyebarkan sumber daya saat menerima `workflow_dispatch` acara. Anda dapat menyesuaikan konfigurasi ini dengan mengubah acara ke `workflow_call` dan memanggil alur kerja dari alur kerja induk lain.	DevOps insinyur
Sesuaikan alur kerja.	File workflow.yaml dikonfigurasi untuk membuat alur kerja yang dinamis dan dapat digunakan kembali. GitHub Anda dapat mengedit file ini untuk menyesuaikan konfigurasi default, atau Anda dapat meneruskan nilai input dari konsol GitHub Tindakan jika Anda menggunakan `workflow_dispatch` acara untuk memulai penerapan secara manual.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Pastikan untuk menentukan ID akun AWS dan Wilayah target yang benar.
	• Buat kebijakan siklus hidup Amazon ECR (lihat kebijakan sampel) dan perbarui jalur default (`e2e-test/policy.json`) yang sesuai.
	• File alur kerja membutuhkan dua peran IAM sebagai masukan:
	  • Peran IAM yang memiliki izin untuk menyiapkan backend Amazon S3 untuk Terraform (lihat bagian Prasyarat). Anda dapat memperbarui nama peran default `workload-assumable-role` di. `yaml`berkas yang sesuai.
	  • Peran IAM yang memiliki izin untuk mengakses . GitHub Peran ini juga digunakan dalam kebijakan Amazon ECR untuk membatasi operasi ECR Amazon. Untuk informasi selengkapnya, lihat file data.tf.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan templat Terraform.	Alur kerja secara otomatis menyebarkan templat Terraform yang membuat repositori ECR Amazon, berdasarkan peristiwa yang Anda konfigurasikan. GitHub Template ini tersedia sebagai `.tf` file di root repositori Github.	AWS DevOps, DevOps insinyur

Pemecahan masalah

Isu	Solusi
Masalah atau kesalahan saat Anda mengonfig urasi Amazon S3 dan DynamoDB sebagai backend jarak jauh Terraform.	Ikuti petunjuk dalam dokumentasi Terraform untuk menyiapkan izin yang diperlukan pada sumber daya Amazon S3 dan DynamoDB untuk konfigurasi backend jarak jauh.
Tidak dapat menjalankan atau memulai alur kerja dengan `workflow_dispatch` acara tersebut.	Alur kerja yang dikonfigurasi untuk diterapka n dari `workflow_dispatch` acara akan berfungsi hanya jika alur kerja dikonfigurasi di cabang utama juga.

Sumber daya terkait

- Menggunakan kembali alur kerja (dokumentasi) GitHub
- Memicu alur kerja (dokumentasi) GitHub

Buat dan uji aplikasi iOS dengan AWS CodeCommit, AWS CodePipeline, dan AWS Device Farm

Abdullahi Olaoye, Amazon Web Services

Ringkasan

Pola ini menguraikan langkah-langkah untuk membuat pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang menggunakan AWS CodePipeline untuk membangun dan menguji aplikasi iOS pada perangkat nyata di AWS. Pola ini menggunakan AWS CodeCommit untuk menyimpan kode aplikasi, alat sumber terbuka Jenkins untuk membangun aplikasi iOS, dan AWS Device Farm untuk menguji aplikasi yang dibangun pada perangkat nyata. Ketiga fase ini diatur bersama dalam pipeline dengan menggunakan AWS. CodePipeline

Pola ini didasarkan pada posting Membangun dan menguji aplikasi iOS dan iPadOS dengan AWS DevOps dan layanan seluler di blog DevOps AWS. Untuk petunjuk rinci, lihat posting blog.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Akun pengembang Apple
- Membangun server (macOS)
- Xcode versi 11.3 (diinstal dan diatur di server build)
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi di workstation
- Pengetahuan dasar tentang Git

Batasan

- Server build aplikasi harus menjalankan macOS.
- Server build harus memiliki alamat IP publik, sehingga CodePipeline dapat menghubungkannya dari jarak jauh untuk memulai build.

Arsitektur

Tumpukan teknologi sumber

- Proses pembuatan aplikasi iOS lokal yang melibatkan penggunaan simulator atau pengujian manual pada perangkat fisik

Tumpukan teknologi target

- CodeCommit Repositori AWS untuk menyimpan kode sumber aplikasi
- Server Jenkins untuk membangun aplikasi menggunakan Xcode
- Kumpulan perangkat AWS Device Farm untuk menguji aplikasi pada perangkat nyata

Arsitektur target

Saat pengguna melakukan perubahan ke repositori sumber, pipeline (AWS CodePipeline) mengambil kode dari repositori sumber, memulai build Jenkins, dan meneruskan kode aplikasi ke Jenkins. Setelah build, pipeline mengambil artefak build dan memulai pekerjaan AWS Device Farm untuk menguji aplikasi terhadap kumpulan perangkat.

Alat

- AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dikelola sepenuhnya yang membantu Anda mengotomatiskan saluran pipa rilis untuk pembaruan aplikasi dan infrastruktur yang cepat dan andal. CodePipeline mengotomatiskan fase build, test, dan deploy dari proses rilis Anda setiap kali ada perubahan kode, berdasarkan model rilis yang Anda tentukan.
- AWS CodeCommit adalah layanan kontrol sumber terkelola penuh yang menampung repositori berbasis Git yang aman. Ini memudahkan tim untuk berkolaborasi dalam kode dalam ekosistem yang aman dan sangat skalabel. CodeCommit menghilangkan kebutuhan untuk mengoperasikan sistem kontrol sumber Anda sendiri atau khawatir tentang penskalaan infrastrukturnya.
- AWS Device Farm adalah layanan pengujian aplikasi yang memungkinkan Anda meningkatkan kualitas web dan aplikasi seluler dengan mengujinya di berbagai browser desktop dan perangkat seluler nyata, tanpa harus menyediakan dan mengelola infrastruktur pengujian apa pun.
- Jenkins adalah server otomatisasi sumber terbuka yang memungkinkan pengembang untuk membangun, menguji, dan menyebarkan perangkat lunak mereka.

Epik

Siapkan lingkungan build

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Jenkins di server build yang menjalankan macOS.	Jenkins akan digunakan untuk membangun aplikasi, jadi Anda harus menginstalnya terlebih dahulu di server build. Untuk mendapatkan petunjuk terperinci untuk tugas ini dan selanjutnya, lihat postingan blog AWS Membangun dan menguji aplikasi iOS dan iPadOS dengan AWS DevOps dan layanan seluler serta sumber daya lainnya di bagian Sumber daya terkait di akhir pola ini.	DevOps
Konfigurasikan Jenkins.	Ikuti petunjuk di layar untuk mengkonfigurasi Jenkins.	DevOps
Instal CodePipeline plugin AWS untuk Jenkins.	Plugin ini harus diinstal pada server Jenkins agar Jenkins dapat berinteraksi dengan layanan CodePipeline AWS.	DevOps
Buat proyek gaya bebas Jenkins.	Di Jenkins, buat proyek gaya bebas. Konfigurasikan proyek untuk menentukan pemicu dan opsi konfigurasi build lainnya.	DevOps

Konfigurasikan AWS Device Farm

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat proyek Device Farm.	Buka konsol AWS Device Farm. Buat proyek dan kumpulan perangkat untuk pengujian. Untuk instruksi, lihat posting blog.	Developer

Konfigurasikan repositori sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodeCommit repositori.	Buat repositori tempat kode sumber akan disimpan.	DevOps
Komit kode aplikasi Anda ke repositori.	Connect ke CodeCommit repositori yang Anda buat. Dorong kode dari mesin lokal Anda ke repositori.	DevOps

Konfigurasikan pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pipeline di AWS CodePipeline.	Buka CodePipeline konsol AWS, dan buat pipeline. Pipa mengatur semua fase proses. CI/CD Untuk petunjuk, lihat postingan blog AWS Membangun dan menguji aplikasi iOS dan iPadOS dengan AWS DevOps dan layanan seluler.	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan tahap uji ke pipa.	Untuk menambahkan tahap pengujian dan menginteg rasikannya dengan AWS Device Farm, edit pipeline.	DevOps
Memulai pipa.	Untuk memulai pipeline dan CI/CD prosesnya, pilih Rilis perubahan.	DevOps

Lihat hasil tes aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau hasil tes.	Di konsol AWS Device Farm, pilih project yang Anda buat, dan tinjau hasil pengujian. Konsol akan menampilkan detail setiap tes.	Developer

Sumber daya terkait

Step-by-step instruksi untuk pola ini

- Membangun dan menguji aplikasi iOS dan iPadOS dengan AWS DevOps dan layanan seluler (postingan DevOps blog AWS)

Konfigurasikan AWS Device Farm

- Konsol AWS Device Farm

Konfigurasikan repositori sumber

- Buat repositori CodeCommit AWS
- Connect ke CodeCommit repositori AWS

Konfigurasikan pipa

- CodePipeline Konsol AWS

Sumber daya tambahan

- CodePipeline Dokumentasi AWS
- CodeCommit Dokumentasi AWS
- Dokumentasi AWS Device Farm
- Dokumentasi Jenkins
- Instalasi Jenkins di macOS
- AWS CodePipeline Plugin untuk Jenkins
- Instalasi Xcode
- Instalasi dan konfigurasi AWS CLI
- Dokumentasi Git

Konfigurasikan otentikasi TLS timbal balik untuk aplikasi yang berjalan di Amazon EKS

Mahendra Revanasiddappa, Amazon Web Services

Ringkasan

Mutual Transport Layer Security (TLS) berbasis sertifikat adalah komponen TLS opsional yang menyediakan otentikasi peer dua arah antara server dan klien. Dengan TLS timbal balik, klien harus memberikan sertifikat X.509 selama proses negosiasi sesi. Server menggunakan sertifikat ini untuk mengidentifikasi dan mengotentikasi klien.

Mutual TLS adalah persyaratan umum untuk aplikasi Internet of Things (IoT) dan dapat digunakan business-to-business untuk aplikasi atau standar seperti Open Banking.

Pola ini menjelaskan cara mengonfigurasi TLS timbal balik untuk aplikasi yang berjalan pada cluster Amazon Elastic Kubernetes Service (Amazon EKS) dengan menggunakan pengontrol ingress NGINX. Anda dapat mengaktifkan fitur TLS timbal balik bawaan untuk pengontrol ingress NGINX dengan membuat anotasi sumber daya ingress. Untuk informasi selengkapnya tentang anotasi TLS timbal balik pada pengontrol NGINX, lihat Autentikasi sertifikat klien dalam dokumentasi Kubernetes.

> **Important**
>
> Pola ini menggunakan sertifikat yang ditandatangani sendiri. Kami menyarankan Anda menggunakan pola ini hanya dengan cluster pengujian, dan bukan di lingkungan produksi. Jika Anda ingin menggunakan pola ini di lingkungan produksi, Anda dapat menggunakan AWS Private Certificate Authority (AWS Private CA) atau standar infrastruktur kunci publik (PKI) yang ada untuk menerbitkan sertifikat pribadi.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif.
- Sebuah klaster Amazon EKS yang sudah ada.
- AWS Command Line Interface (AWS CLI) versi 1.7 atau yang lebih baru, diinstal dan dikonfigurasi di macOS, Linux, atau Windows.

- Utilitas baris perintah kubectl, diinstal dan dikonfigurasi untuk mengakses kluster Amazon EKS. Untuk informasi selengkapnya tentang ini, lihat Menginstal kubectl di dokumentasi Amazon EKS.
- Nama Domain Name System (DNS) yang ada untuk menguji aplikasi.

Batasan

- Pola ini menggunakan sertifikat yang ditandatangani sendiri. Kami menyarankan Anda menggunakan pola ini hanya dengan cluster pengujian, dan bukan di lingkungan produksi.

Arsitektur

Tumpukan teknologi

- Amazon EKS
- Amazon Route 53
- Kubectl

Alat

- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.
- Kubectl adalah utilitas baris perintah yang Anda gunakan untuk berinteraksi dengan cluster Amazon EKS.

Epik

Menghasilkan sertifikat yang ditandatangani sendiri

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan kunci CA dan sertifikat.	Hasilkan kunci dan sertifikat otoritas sertifikat (CA) dengan menjalankan perintah berikut.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`openssl req -x509 -sha256 -newkey rsa:4096 -keyout ca.key -out ca.crt -days 356 -nodes -subj '/CN=Test Cert Authority'`	
Hasilkan kunci server dan sertifikat, dan tandatangani dengan sertifikat CA.	Hasilkan kunci server dan sertifikat, dan tandatangani dengan sertifikat CA dengan menjalankan perintah berikut. `openssl req -new -newkey rsa:4096 -keyout server.key -out server.csr -nodes -subj '/CN= <your_domain_name> ' && openssl x509 -req -sha256 -days 365 -in server.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out server.crt` **Important** Pastikan Anda mengganti `<your_domain_name>` dengan nama domain yang ada.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kunci klien dan sertifikat, dan tandatangani dengan sertifikat CA.	Buat kunci klien dan sertifikat, dan tandatangani dengan sertifikat CA dengan menjalankan perintah berikut. `openssl req -new -newkey rsa:4096 -keyout client.key -out client.csr -nodes -subj '/CN=Test' && openssl x509 -req -sha256 -days 365 -in client.csr -CA ca.crt -CAkey ca.key -set_serial 02 -out client.crt`	DevOps insinyur

Terapkan pengontrol masuk NGINX

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan pengontrol ingress NGINX di kluster Amazon EKS Anda.	Menyebarkan pengontrol ingress NGINX dengan menggunakan perintah berikut. `kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.7.0/deploy/static/provider/aws/deploy.yaml`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa layanan pengontrol ingress NGINX sedang berjalan.	Verifikasi bahwa layanan pengontrol ingress NGINX berjalan dengan menggunakan perintah berikut. `kubectl get svc -n ingress-nginx` **Important** Pastikan bahwa bidang alamat layanan berisi nama domain Network Load Balancer.	DevOps insinyur

Buat namespace di cluster Amazon EKS untuk menguji TLS timbal balik

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat namespace di kluster Amazon EKS.	Buat namespace yang dipanggil `mtls` di cluster Amazon EKS Anda dengan menjalankan perintah berikut. `kubectl create ns mtls` Ini menyebarkan aplikasi sampel untuk menguji TLS timbal balik.	DevOps insinyur

Buat penyebaran dan layanan untuk aplikasi sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat deployment dan service Kubernetes di namespace mtls.	Buat file bernama `mtls.yaml` . Tempel kode berikut ke file.
	
	`kind: Deployment`
	`apiVersion: apps/v1`
	`metadata:`
	`  name: mtls-app`
	`  labels:`
	`  app: mtls`
	`spec:`
	`  replicas: 1`
	`  selector:`
	`  matchLabels:`
	`  app: mtls`
	`  template:`
	`  metadata:`
	`  labels:`
	`  app: mtls`
	`  spec:`
	`  containers:`
	`  - name: mtls-app`
	`  image: hashicorp/http-echo`
	`  args:`
	`  - "-text=mTLS is working"`
	
	`---`
	
	`kind: Service`
	`apiVersion: v1`
	`metadata:`
	`  name: mtls-service`
	`spec:`
	`  selector:`
	`  app: mtls`
	`  ports:`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`- port: 5678 # Default port for image`
	
	Buat deployment dan service Kubernetes di `mtls` namespace dengan menjalankan perintah berikut.
	
	`kubectl create -f mtls.yaml -n mtls`	

Verifikasi bahwa penerapan Kubernetes dibuat.	Jalankan perintah berikut untuk memverifikasi bahwa penerapan dibuat dan memiliki satu pod dalam status yang tersedia. `kubectl get deploy -n mtls`	DevOps insinyur
Verifikasi bahwa layanan Kubernetes dibuat.	Verifikasi bahwa layanan Kubernetes dibuat dengan menjalankan perintah berikut. `kubectl get service -n mtls`	DevOps insinyur

Buat rahasia di namespace mtls

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat rahasia untuk sumber daya ingress.	Jalankan perintah berikut untuk membuat rahasia untuk pengontrol ingress NGINX	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan menggunakan sertifikat yang Anda buat sebelumnya. `kubectl create secret generic mtls-certs --from-file=tls.crt=server.crt --from-file=tls.key=server.key --from-file=ca.crt=ca.crt -n mtls` Rahasia Anda memiliki sertifikat server untuk klien untuk mengidentifikasi server dan sertifikat CA untuk server untuk memverifikasi sertifikat klien.	

Buat sumber daya ingress di namespace mtls

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat sumber daya ingress di namespace mtls.	Buat file bernama `ingress.yaml` . Rekatkan kode berikut ke dalam file (ganti `<your_domain_name>` dengan nama domain yang ada).
	
	`apiVersion: networking.k8s.io/v1`
	`kind: Ingress`
	`metadata:`
	`  annotations:`
	`  nginx.ingress.kubernetes.io/auth-tls-verify-client: "on"`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> nginx.ingress.kube
	rnetes.io/auth-tls-
	secret: mtls/mtls-certs
	  name: mtls-ingress
	spec:
	  ingressClassName:
	 nginx
	  rules:
	  - host: "*.<your_
	domain_name>"
	    http:
	      paths:
	      - path: /
	        pathType: Prefix
	        backend:
	          service:
	            name: mtls-
	service
	            port:
	              number:
	 5678
	  tls:
	  - hosts:
	    - "*.<your_
	domain_name>"
	    secretName: mtls-
	certs</pre>
	Buat sumber daya ingress di `mtls` namespace dengan menjalankan perintah berikut.
	<pre>kubectl create -f
	 ingress.yaml -n mtls</pre>
	Ini berarti bahwa pengontro l ingress NGINX dapat merutekan lalu lintas ke aplikasi sampel Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa sumber daya ingress dibuat.	Verifikasi bahwa sumber daya ingress dibuat dengan menjalankan perintah berikut. `kubectl get ing -n mtls` **Important** Pastikan bahwa alamat sumber daya ingress menunjukkan penyeimbang beban yang dibuat untuk pengontrol ingress NGINX.	DevOps insinyur

Konfigurasikan DNS untuk mengarahkan nama host ke penyeimbang beban

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat catatan CNAME yang menunjuk ke penyeimbang beban untuk pengontrol ingress NGINX.	Masuk ke AWS Management Console, buka konsol Amazon Route 53, dan buat catatan Canonical Name (CNAME) yang mengarah `mtls.<your_domain_name>` ke penyeimbang beban untuk pengontrol ingress NGINX. Untuk informasi selengkapnya, lihat Membuat catatan menggunakan konsol Route	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	53 dalam dokumentasi Route 53.	

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji pengaturan TLS timbal balik tanpa sertifikat.	Jalankan perintah berikut. `curl -k https://m tls.<your_domain_n ame>` Anda harus menerima respons kesalahan “400 Tidak diperlukan sertifikat SSL dikirim”.	DevOps insinyur
Uji pengaturan TLS timbal balik dengan sertifikat.	Jalankan perintah berikut. `curl -k https://m tls.<your_domain_n ame> --cert client.crt --key client.key` Anda harus menerima respons “mTLS berfungsi”.	DevOps insinyur

Sumber daya terkait

- Membuat rekaman dengan menggunakan konsol Amazon Route 53
- Menggunakan Network Load Balancer dengan pengontrol ingress NGINX di Amazon EKS
- Otentikasi Sertifikat Klien

Otomatiskan pembuatan sumber daya WorkSpaces Aplikasi Amazon menggunakan AWS CloudFormation

Ram Kandaswamy, Amazon Web Services

Ringkasan

Pola ini menyediakan contoh kode dan langkah-langkah untuk mengotomatiskan pembuatan sumber daya WorkSpaces Aplikasi Amazon di AWS Cloud dengan menggunakan AWS CloudFormationtemplat. Pola ini menunjukkan cara menggunakan CloudFormation tumpukan untuk mengotomatiskan pembuatan sumber daya WorkSpaces aplikasi Aplikasi Anda, termasuk pembuat gambar, gambar, instance armada, dan tumpukan. Anda dapat melakukan streaming WorkSpaces aplikasi Aplikasi Anda ke pengguna akhir pada browser yang HTML5 sesuai dengan menggunakan mode pengiriman desktop atau aplikasi.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Penerimaan syarat dan ketentuan WorkSpaces Aplikasi
- Pengetahuan dasar tentang sumber daya WorkSpaces Aplikasi, seperti armada dan tumpukan dan pembuat gambar

Batasan

- Anda tidak dapat mengubah peran AWS Identity and Access Management(IAM) yang terkait dengan instance WorkSpaces Aplikasi setelah instance tersebut dibuat.
- Anda tidak dapat mengubah properti (seperti subnet atau grup keamanan) pada instance pembuat gambar WorkSpaces Aplikasi setelah pembuat gambar tersebut dibuat.

Arsitektur

Diagram berikut menunjukkan kepada Anda cara mengotomatiskan pembuatan sumber daya WorkSpaces Aplikasi dengan menggunakan CloudFormation templat.

Diagram menunjukkan alur kerja berikut:

1. Anda membuat CloudFormation template berdasarkan kode YAMB di bagian Informasi tambahan dari pola ini.
2. CloudFormation Template membuat tumpukan CloudFormation uji.
   a. (Opsional) Anda membuat instance pembuat gambar dengan menggunakan WorkSpaces Aplikasi.
   b. (Opsional) Anda membuat gambar Windows dengan menggunakan perangkat lunak khusus Anda.
3. CloudFormation Tumpukan membuat instance dan tumpukan armada WorkSpaces Aplikasi.
4. Anda menerapkan sumber daya WorkSpaces Aplikasi Anda ke pengguna akhir pada browser HTML5 yang sesuai.

Alat

- Amazon WorkSpaces Applications adalah layanan streaming aplikasi yang dikelola sepenuhnya yang memberi Anda akses instan ke aplikasi desktop Anda dari mana saja. WorkSpaces Aplikasi mengelola AWS sumber daya yang diperlukan untuk meng-host dan menjalankan aplikasi Anda, menskalakan secara otomatis, dan menyediakan akses ke pengguna sesuai permintaan.
- AWS CloudFormationmembantu Anda memodelkan dan menyiapkan AWS sumber daya Anda, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Anda dapat mengelola dan menyediakan tumpukan di beberapa Akun AWS dan Wilayah AWS.

Praktik terbaik

- Konfigurasikan akses jaringan untuk pembuat gambar dengan benar — Luncurkan pembuat gambar di subnet virtual private cloud (VPC) dengan akses internet yang tepat dengan menggunakan gateway NAT untuk akses internet khusus keluar.

  Uji konektivitas jaringan ke sumber daya yang diperlukan (seperti server aplikasi, database, dan server lisensi) sebelum membuat gambar. Verifikasi bahwa tabel rute VPC memungkinkan koneksi ke semua sumber daya jaringan yang diperlukan. Untuk informasi selengkapnya, lihat Akses Internet di dokumentasi WorkSpaces Aplikasi.

- Pantau kapasitas armada terhadap kuota layanan secara proaktif — Jenis dan kuota ukuran instans WorkSpaces aplikasi adalah per, per Akun AWS. Wilayah AWS Jika Anda memiliki beberapa armada di Wilayah yang sama yang menggunakan jenis dan ukuran instans yang sama, jumlah total instans di semua armada di Wilayah tersebut harus kurang dari atau sama dengan kuota yang berlaku. Untuk informasi selengkapnya, lihat Memecahkan Masalah Armada di dokumentasi Aplikasi. WorkSpaces

- Uji aplikasi dalam mode Image Builder Test sebelum penerapan armada — Selalu validasi aplikasi dalam mode Image Builder Test sebelum membuat gambar dan menyebarkan ke armada. Mode pengujian mensimulasikan izin terbatas yang dimiliki pengguna akhir pada instance armada. Untuk informasi selengkapnya, lihat Pemecahan Masalah Pembuat Gambar di dokumentasi WorkSpaces Aplikasi.

Epik

(Opsional) Buat gambar WorkSpaces Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal perangkat lunak khusus dan buat gambar.	1. Luncurkan pembuat gambar dan pilih jenis instans yang relevan, subnet, grup keamanan, dan sebagainya. 2. Gunakan mode admin saat menghubungkan ke instance. 3. Gunakan asisten Gambar atau PowerShell skrip untuk membuat gambar Windows baru untuk perangkat lunak khusus Anda.	AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Pertimbangkan untuk menggunakan AppLocker fitur Windows untuk mengunci gambar lebih lanjut.	

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui CloudFormation template.	1. Gunakan kode di bagian Informasi tambahan dari pola ini sebagai file YAMB. 2. Perbarui file YAMB dengan nilai yang diperlukan untuk parameter di lingkungan Anda.	Administrator sistem AWS, Administrator cloud, arsitek Cloud, AWS Umum, administrator AWS
Buat CloudFormation tumpukan dengan menggunakan template.	1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol. 2. Pada panel navigasi, pilih Stacks. 3. Pilih Buat tumpukan kemudian pilih Dengan sumber daya baru (standar). 4. Di bagian Prasyarat - Siapkan templat, pilih Template sudah siap.	Pemilik aplikasi, administrator sistem AWS, Windows Engineer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Di bagian Tentukan templat, pilih Unggah file templat. 6. Pilih Pilih file dan kemudian pilih CloudFormation template yang diperbarui. 7. Selesaikan langkah-langkah lainnya di wizard untuk membuat tumpukan Anda.	

Pemecahan masalah

Isu	Solusi
Berbagai masalah	Untuk informasi selengkapnya, lihat Pemecahan Masalah dalam dokumentasi WorkSpaces Aplikasi.

Sumber daya terkait

Referensi

- Memulai WorkSpaces Aplikasi Amazon: Siapkan dengan contoh aplikasi
- Buat armada dan tumpukan WorkSpaces Aplikasi Amazon

Tutorial dan video

- Alur Kerja Pengguna WorkSpaces Aplikasi Amazon
- Cara Memigrasi Aplikasi Formulir Windows Lama ke Aplikasi Amazon WorkSpaces
- AWS re: Invent 2018: Mengirimkan Aplikasi Desktop dengan Aman dengan Aplikasi Amazon WorkSpaces (01) BAP2

Informasi tambahan

Kode berikut adalah contoh CloudFormation template yang dapat Anda gunakan untuk secara otomatis membuat sumber daya WorkSpaces Aplikasi.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  SubnetIds:
    Type: 'List<AWS::EC2::Subnet::Id>'
  testSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup::Id'
  ImageName:
    Type: String
Resources:

  AppStreamFleet:
    Type: 'AWS::AppStream::Fleet'
    Properties:
      ComputeCapacity:
        DesiredInstances: 5
      InstanceType: stream.standard.medium
      Name: appstream-test-fleet
      DisconnectTimeoutInSeconds: 1200
      FleetType: ON_DEMAND
      IdleDisconnectTimeoutInSeconds: 1200
      ImageName: !Ref ImageName
      MaxUserDurationInSeconds: 345600
      VpcConfig:
        SecurityGroupIds:
          - !Ref testSecurityGroup
        SubnetIds: !Ref SubnetIds
  AppStreamStack:
    Type: 'AWS::AppStream::Stack'
    Properties:
      Description: AppStream stack for test
      DisplayName: AppStream test Stack
      Name: appstream-test-stack
      StorageConnectors:
        - ConnectorType: HOMEFOLDERS
      UserSettings:
        - Action: CLIPBOARD_COPY_FROM_LOCAL_DEVICE
          Permission: ENABLED
        - Action: CLIPBOARD_COPY_TO_LOCAL_DEVICE
```

```
        Permission: ENABLED
      - Action: FILE_DOWNLOAD
        Permission: ENABLED
      - Action: PRINTING_TO_LOCAL_DEVICE
        Permission: ENABLED
AppStreamFleetAssociation:
  Type: 'AWS::AppStream::StackFleetAssociation'
  Properties:
    FleetName: appstream-test-fleet
    StackName: appstream-test-stack
  DependsOn:
    - AppStreamFleet
    - AppStreamStack
```

Buat parser log khusus untuk Amazon ECS menggunakan router log Firelens

Varun Sharma, Amazon Web Services

Ringkasan

Firelens adalah router log untuk Amazon Elastic Container Service (Amazon ECS) dan AWS Fargate. Anda dapat menggunakan Firelens untuk merutekan log kontainer dari Amazon ECS ke Amazon CloudWatch dan tujuan lainnya (misalnya, Splunk atau Sumo Logic). Firelens bekerja dengan Fluentd atau Fluent Bit sebagai agen logging, yang berarti Anda dapat menggunakan parameter definisi tugas Amazon ECS untuk merutekan log.

Dengan memilih untuk mengurai log di tingkat sumber, Anda dapat menganalisis data logging Anda dan melakukan kueri untuk merespons masalah operasional secara lebih efisien dan efektif. Karena aplikasi yang berbeda memiliki pola logging yang berbeda, Anda perlu menggunakan parser khusus yang menyusun log dan membuat pencarian lebih mudah di tujuan akhir Anda.

Pola ini menggunakan router log Firelens dengan parser khusus untuk mendorong log CloudWatch dari contoh aplikasi Spring Boot yang berjalan di Amazon ECS. Anda kemudian dapat menggunakan Amazon CloudWatch Logs Insights untuk memfilter log berdasarkan bidang kustom yang dihasilkan oleh parser kustom.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi pada mesin lokal Anda.
- Docker, diinstal dan dikonfigurasi pada mesin lokal Anda.
- Aplikasi kontainer berbasis Spring Boot yang ada di Amazon Elastic Container Registry (Amazon ECR).

Arsitektur

Tumpukan teknologi

- CloudWatch
- Amazon ECR
- Amazon ECS
- Fargate
- Docker
- Fluent Bit

Alat

- Amazon ECR — Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola AWS yang aman, terukur, dan andal.
- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat yang memudahkan untuk menjalankan, menghentikan, dan mengelola kontainer di cluster.
- AWS Identity and Access Management (IAM) — IAM adalah layanan web untuk mengontrol akses ke layanan AWS secara aman.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat sumber terbuka yang memungkinkan Anda berinteraksi dengan layanan AWS menggunakan perintah di shell baris perintah Anda.
- Docker — Docker adalah platform terbuka untuk mengembangkan, mengirim, dan menjalankan aplikasi.

Kode

File-file berikut dilampirkan ke pola ini:

- `customFluentBit.zip`— Berisi file untuk menambahkan parsing dan konfigurasi khusus.
- `firelens_policy.json`— Berisi dokumen kebijakan untuk membuat kebijakan IAM.
- `Task.json`- Berisi definisi tugas sampel untuk Amazon ECS.

Epik

Buat gambar Fluent Bit kustom

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR.	Masuk ke AWS Management Console, buka konsol Amazon ECR, dan buat repositori yang disebut. `fluentbit_custom` Untuk informasi selengkapnya tentang ini, lihat Membuat repositori di dokumentasi Amazon ECR.	Administrator sistem, Pengembang
Buka zip customFluentBit paket.zip.	1. Unduh `customFluentBit.zip` paket (terlampir) ke mesin lokal Anda. 2. Unzip ke `customFluentBit` direktori dengan menjalankan perintah berikut: `unzip -d customFluentBit.zip` 3. Direktori berisi file-file berikut yang diperlukan untuk menambahkan parsing dan konfigurasi kustom: • `parsers/springboot_parser.conf` — Berisi direktif parser dan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mendefinisikan pola ekspresi reguler (regex) untuk parser kustom. Anda dapat menambahk an pola regex untuk parser spesifik Anda. • `conf/pars e_springb oot.conf` — Berisi filter dan arahan layanan. • Dockerfile	
Buat gambar Docker kustom.	1. Ubah direktori ke `customFluentBit` . 2. Buka konsol Amazon ECR, pilih `fluentbit_custom` repositori, lalu pilih Lihat perintah push. 3. Unggah proyek Anda. 4. Setelah unggahan selesai, salin URL build. URL ini diperlukan saat Anda membuat wadah di Amazon ECS Untuk informasi selengkapnya tentang ini, lihat Mendorong gambar Docker di dokumenta si Amazon ECR.	Administrator sistem, Pengembang

Siapkan cluster Amazon ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster Amazon ECS.	Buat klaster Amazon ECS dengan mengikuti petunjuk dari bagian Template Networking only pada Membuat klaster di dokumentasi Amazon ECS. **Note** Pastikan Anda memilih Create VPC untuk membuat virtual private cloud (VPC) baru untuk cluster Amazon ECS Anda.	Administrator sistem, Pengembang

Siapkan tugas Amazon ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan peran IAM eksekusi tugas Amazon ECS.	Buat peran IAM eksekusi tugas Amazon ECS dengan menggunakan kebijakan `AmazonECSTaskExecutionRolePolicy` terkelola. Untuk informasi selengkapnya tentang ini, lihat peran IAM eksekusi tugas Amazon ECS dalam dokumentasi Amazon ECS.	Administrator sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Pastikan Anda merekam Nama Sumber Daya Amazon (ARN) peran IAM.	
Lampirkan kebijakan IAM ke peran IAM eksekusi tugas Amazon ECS.	1. Buat kebijakan IAM dengan menggunak an dokumen kebijakan `firelens_policy.js` on (terlampir). Untuk informasi selengkapnya tentang ini, lihat Membuat kebijakan pada tab JSON di dokumentasi IAM. 2. Lampirkan kebijakan ini ke peran IAM eksekusi tugas Amazon ECS yang Anda buat sebelumny a. Untuk informasi selengkapnya tentang ini, lihat Menambahkan kebijakan IAM (AWS CLI) di dokumentasi IAM.	Administrator sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan definisi tugas Amazon ECS.	1. Perbarui bagian berikut dalam definisi tugas `Task.json` sampel (terlampir): • Perbarui `executionRoleArn` dan `taskRoleArn` dengan ARN dari peran IAM eksekusi tugas • Perbarui gambar `containerDefinitions` dengan gambar Fluent Bit Docker khusus yang Anda buat sebelumnya • Perbarui gambar `containerDefinitions` dengan nama gambar aplikasi Anda 2. Buka konsol Amazon ECS, pilih Definisi Tugas, pilih Buat definisi tugas baru, lalu pilih Fargate di halaman Pilih kompatibilitas. 3. Pilih Konfigurasi melalui Json, tempel `Task.json` file yang diperbarui ke area teks, lalu pilih Simpan. 4. Buat definisi tugas. Untuk informasi selengkap nya tentang ini, lihat	Administrator sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Membuat definisi tugas dalam dokumentasi Amazon ECS.	

Jalankan tugas Amazon ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan tugas Amazon ECS.	Di konsol Amazon ECS, pilih Cluster, pilih klaster yang Anda buat sebelumnya, lalu jalankan tugas mandiri. Untuk informasi selengkapnya tentang ini, lihat Menjalankan tugas mandiri di dokumentasi Amazon ECS.	Administrator sistem, Pengembang

Verifikasi CloudWatch log

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi log.	1. Buka CloudWatch konsol, pilih Grup log, lalu pilih `/aws/ecs/containerinsights/{{cluster_ARN}}/firelens/application` . 2. Verifikasi log, terutama bidang kustom yang ditambahkan oleh parser kustom.	Administrator sistem, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Gunakan CloudWatch untuk memfilter log berdasarkan bidang kustom.	

Sumber daya terkait

- Dasar-dasar Docker untuk Amazon ECS
- Amazon ECS di AWS Fargate
- Mengkonfigurasi parameter layanan dasar

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat kerangka orkestrasi sumber daya berbasis API menggunakan Actions dan Terragrunt GitHub

Tamilselvan P, Abhigyan Dandriyal, Sandeep Gawande, dan Akash Kumar, Amazon Web Services

Ringkasan

Pola ini memanfaatkan alur kerja GitHub Actions untuk mengotomatiskan penyediaan sumber daya melalui muatan JSON standar, sehingga tidak perlu konfigurasi manual. Pipeline otomatis ini mengelola siklus hidup penerapan lengkap dan dapat berintegrasi dengan mulus dengan berbagai sistem frontend, dari komponen UI kustom hingga. ServiceNow Fleksibilitas solusi memungkinkan pengguna untuk berinteraksi dengan sistem melalui antarmuka pilihan mereka sambil mempertahankan proses standar.

Arsitektur pipa yang dapat dikonfigurasi dapat disesuaikan untuk memenuhi persyaratan organisasi yang berbeda. Implementasi contoh berfokus pada penyediaan Amazon Virtual Private Cloud (Amazon VPC) dan Amazon Simple Storage Service (Amazon S3). Pola ini secara efektif mengatasi tantangan manajemen sumber daya cloud umum dengan menstandarisasi permintaan di seluruh organisasi dan memberikan poin integrasi yang konsisten. Pendekatan ini memudahkan tim untuk meminta dan mengelola sumber daya sambil memastikan standardisasi.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- GitHub Akun aktif dengan akses ke repositori yang dikonfigurasi

Batasan

- Sumber daya baru memerlukan penambahan `terragrunt.hcl` file secara manual ke konfigurasi repositori.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan komponen dan alur kerja pola ini.

Diagram arsitektur menunjukkan tindakan berikut:

1. Pengguna mengirimkan payload JSON ke GitHub Actions, memicu pipeline otomatisasi.
2. Pipeline GitHub Actions mengambil kode sumber daya yang diperlukan dari repositori Terragrunt dan Terraform, berdasarkan spesifikasi payload.
3. Pipeline mengasumsikan peran yang sesuai AWS Identity and Access Management (IAM) menggunakan ID yang ditentukan Akun AWS . Kemudian, pipeline menyebarkan sumber daya ke target Akun AWS dan mengelola status Terraform menggunakan bucket Amazon S3 khusus akun dan tabel Amazon DynamoDB.

Masing-masing Akun AWS berisi peran IAM untuk akses aman, bucket Amazon S3 untuk penyimpanan status Terraform, dan tabel DynamoDB untuk penguncian status. Desain ini memungkinkan penyebaran sumber daya yang terkontrol dan otomatis di seluruh Akun AWS. Proses penyebaran mempertahankan manajemen status dan kontrol akses yang tepat melalui bucket Amazon S3 khusus dan peran IAM di setiap akun.

Alat

Layanan AWS

- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat-alat lainnya

- GitHub Actions adalah platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang terintegrasi erat dengan GitHub repositori. Anda dapat menggunakan GitHub Actions untuk mengotomatiskan pipeline build, test, dan deployment.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.
- Terragrunt adalah alat orkestrasi yang memperluas keduanya dan kemampuan Terraform. OpenTofu Ini mengelola bagaimana pola infrastruktur generik diterapkan, membuatnya lebih mudah untuk skala dan memelihara perkebunan infrastruktur besar.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub sample-aws-orchestration-pipeline-terraform.

Praktik terbaik

- Simpan AWS kredensil dan data sensitif menggunakan rahasia GitHub repositori untuk akses aman.
- Konfigurasikan penyedia OpenID Connect (OIDC) untuk GitHub Actions untuk mengambil peran IAM, menghindari kredensi statis.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Buat dan konfigurasikan repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi GitHub repositori.	Untuk menginisialisasi GitHub repositori, gunakan langkah-l angkah berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Buat GitHub repositori baru untuk meng-host kode pipeline. 2. Impor `workflow-trigger.yml` file `deployment.yml` dan yang terletak di folder.github/workflows dari repositori sumber.	
Konfigurasikan peran dan izin IAM.	Untuk mengonfigurasi peran dan izin IAM, gunakan langkah-langkah berikut: 1. Buat peran IAM dengan hubungan kepercayaan yang diperlukan untuk GitHub Actions untuk terhubung ke tindakan dalam AWS menggunakan penyedia identitas OpenID Connect (OIDC) (iDP). 2. Lampirkan izin yang diperlukan ke peran IAM untuk membuat backend dan sumber daya yang diinginkan. Untuk informasi selengkapnya, lihat contoh kebijakan untuk membuat VPC dengan Amazon VPC beserta bucket Amazon S3 backend dan tabel DynamoDB.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan GitHub rahasia dan variabel.	Untuk petunjuk tentang cara mengatur rahasia dan variabel repositori di GitHub repositor i, lihat Membuat variabel konfigurasi untuk repositori dalam dokumentasi. GitHub Konfigurasikan variabel berikut:
	• Rahasia repositori
	  • `PAT_TOKEN` — Token akses pribadi Anda dengan izin untuk melakukan operasi GitHub
	• Variabel repositori
	  • `aws_role`— Peran IAM Nama Sumber Daya Amazon (ARN) dengan izin yang sesuai untuk menyebarkan sumber daya yang diperlukan dan GitHub menghubungkan tindakan ke tindakan di AWS
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat struktur repositori.	Untuk membuat struktur repositori, gunakan langkah-l angkah berikut: 1. Buat folder baru di `main` cabang untuk menyimpan `terragrunt.hcl` file, serta output dan changelog setelah pembuatan sumber daya. 2. Beri nama folder menggunakan huruf kecil sesuai dengan jenis sumber daya yang ingin Anda sediakan melaluinya. Folder digunakan apa adanya nanti di payload. Untuk informasi selengkapnya, lihat struktur sampel untuk Amazon S3 dan Amazon VPC di repositori.	DevOps insinyur

Memicu pipeline dan memvalidasi hasil

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan pipa menggunakan curl.	Untuk menjalankan pipeline dengan menggunakan curl, gunakan langkah-langkah berikut: 1. Buat file payload.json di direktori lokal Anda. Ikuti struktur yang ditentuka	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n untuk file payload, dan stringify sebelum mengirimk an permintaan. Untuk informasi selengkapnya, lihat contoh muatan di repositori. 2. Kirimkan permintaan di terminal Anda dengan menggunakan GitHub API seperti yang ditunjukkan pada contoh berikut: `curl -X POST \ -H "Accept: application/vnd.github.v3+json" \ -H "Authorization: token YOUR_GITHUB_TOKEN" \ -d @payload.json \ https://api.github.com/repos/OWNER/REPO/actions/workflows/workflow-trigger.yml/dispatches` Dalam contoh, berikan nilai Anda sendiri untuk hal-hal berikut: • `YOUR_GITHUB_TOKEN` dengan token akses GitHub pribadi Anda • `OWNER`dengan nama pemilik repositori	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `REPO`dengan nama repositori Untuk informasi selengkap nya tentang proses eksekusi pipeline, lihat Informasi tambahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi hasil eksekusi pipeline	Untuk memvalidasi hasil, gunakan langkah-langkah berikut: 1. Pantau eksekusi alur kerja GitHub Tindakan di tab Tindakan repositori Anda. 2. Setelah eksekusi alur kerja berhasil, verifikasi pembuatan sumber daya dengan menggunakan Konsol Manajemen AWS sebagai berikut: a. Untuk Amazon VPC: • Arahkan ke layanan Amazon VPC dalam yang ditentukan. Wilayah AWS • Periksa VPC baru dengan tag yang sesuai dengan parameter permintaan Anda. • Verifikasi blok CIDR dan konfigurasi lainnya. b. Untuk Amazon S3: • Arahkan ke ember Amazon S3. • Di bucket tujuan umum, periksa bucket baru yang cocok	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan parameter permintaan Anda. • Verifikasi nama bucket S3 dan konfigurasi lainnya. Anda juga dapat memverifikasi silang sumber daya yang dibuat dengan menggunakan `output.json` file yang dibuat di repositori yang berada di dalam sumber daya yang sama dengan file. `terragrunt.hcl`	

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim permintaan pembersihan.	Untuk menghapus sumber daya yang tidak lagi diperlukan, gunakan langkah-langkah berikut: 1. Kirim permintaan penghapusan dengan menggunakan titik akhir API yang sama tetapi ubah payload sebagai berikut: • Ubah `RequestType` ke `delete` dalam `RequestParameters` .	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Jaga agar semua parameter lain identik dengan `create` permintaan. 2. Pantau proses penghapus an dalam GitHub Tindakan. 3. Setelah alur kerja selesai, verifikasi bahwa `changelog.json` file di dalam folder sumber daya menunjukkan status. `deleted` 4. Verifikasi penghapusan sumber daya dengan menggunakan Konsol Manajemen AWS.	

Sumber daya terkait

AWS Blog

- Gunakan peran IAM untuk menghubungkan GitHub Tindakan ke tindakan di AWS

Layanan AWS dokumentasi

- Pembuatan peran IAM
- Memantau file CloudTrail log dengan Amazon CloudWatch Logs
- Praktik terbaik keamanan untuk Amazon S3

GitHub sumber daya

- Buat acara pengiriman repositori
- Membuat webhooks

- Menerapkan kontrol akses yang kuat pada GitHub repositori
- Secara teratur mengaudit akses repositori
- Pemeriksaan keamanan di dalam CI/CD pipa
- Gunakan otentikasi multi-faktor untuk akun GitHub

Informasi tambahan

Proses eksekusi pipa

Berikut ini adalah langkah-langkah eksekusi pipeline:

1. Memvalidasi format payload JSON - Memastikan bahwa konfigurasi JSON yang masuk terstruktur dengan benar dan berisi semua parameter yang diperlukan
2. Mengasumsikan peran IAM yang ditentukan - Mengautentikasi dan mengasumsikan peran IAM yang diperlukan untuk operasi AWS
3. Diperlukan unduhan kode Terraform dan Terragrunt - Mengambil versi kode sumber daya dan dependensi yang ditentukan
4. Menjalankan penyebaran sumber daya - Menerapkan konfigurasi untuk menyebarkan atau memperbarui AWS sumber daya di lingkungan target

Contoh payload yang digunakan untuk pembuatan VPC

Berikut ini adalah contoh kode untuk pembuatan bucket status backend Terraform:

```
state_bucket_name = "${local.payload.ApplicationName}-${local.payload.EnvironmentId}-
tfstate"
```

```
lock_table_name = "${local.payload.ApplicationName}-${local.payload.EnvironmentId}-
tfstate-lock"
```

Berikut ini adalah contoh payload untuk membuat VPC dengan Amazon VPC, `vpc_cidr` di mana mendefinisikan spesifikasi blok CIDR untuk VPC. Bucket status Terraform dipetakan ke variabel yang ditentukan dalam file. `terraform` `ref`Parameter berisi nama cabang kode yang akan dieksekusi.

```
{
    "ref": "main",
    "inputs": {
```

```
        "RequestParameters": {
            "RequestId": "1111111",
            "RequestType": "create",
            "ResourceType": "vpc",
            "AccountId": "1234567890",
            "AccountAlias": "account-alias",
            "RegionId": "us-west-2",
            "ApplicationName": "myapp",
            "DivisionName": "division-name",
            "EnvironmentId": "dev",
            "Suffix": "poc"
        },
        "ResourceParameters": [
            {
                "VPC": {
                    "vpc_cidr": "10.0.0.0/16"
                }
            }
        ]
    }
}
```

`RequestParameters`digunakan untuk melacak status permintaan di bagian pipeline dan `tfstate` dibuat berdasarkan informasi ini. Parameter berikut berisi metadata dan informasi kontrol:

- `RequestId`— Pengidentifikasi unik untuk permintaan
- `RequestType`— Jenis operasi (membuat, memperbarui, atau menghapus)
- `ResourceType`— Jenis sumber daya yang akan disediakan
- `AccountId`— Target Akun AWS untuk penyebaran
- `AccountAlias`— Nama ramah untuk Akun AWS
- `RegionId`— Wilayah AWS untuk penyebaran sumber daya
- `ApplicationName`— Nama aplikasi
- `DivisionName`— Divisi organisasi
- `EnvironmentId`— Lingkungan (misalnya, dev dan prod)
- `Suffix`— Pengenal tambahan untuk sumber daya

`ResourceParameters`berisi konfigurasi khusus sumber daya yang memetakan ke variabel yang ditentukan dalam file Terraform. Variabel khusus apa pun yang perlu diteruskan ke modul Terraform

harus disertakan di dalamnya. `ResourceParameters` Parameter `vpc_cidr` ini wajib untuk Amazon VPC.

Buat permintaan tarik otomatis untuk infrastruktur yang dikelola Terraform AWS dengan menggunakan Tindakan GitHub

Matt Padgett, Ashish Bhatt, Ashwin Divakaran, Sandip Gangapadhyay, dan Prafful Gupta, Amazon Web Services

Ringkasan

Pola ini menyajikan utilitas otomatisasi yang dirancang untuk menghilangkan pekerjaan manual dan berulang yang terlibat dalam mengelola perubahan di beberapa repositori Terraform. Banyak organisasi menggunakan repositori Terraform untuk mengelola infrastruktur mereka sebagai kode (IAc), seringkali dengan ratusan repositori terpisah yang mewakili lingkungan, layanan, atau tim yang berbeda. Mengelola repositori ini dalam skala besar menghadirkan tantangan operasional yang signifikan. Tugas rutin seperti memperbarui parameter, memutakhirkan versi modul, atau menerapkan perubahan konfigurasi sering kali memerlukan pembuatan dan pengelolaan pull request (PRs) di banyak repositori beberapa kali sehari.

Bahkan untuk perubahan sederhana, proses berulang dan manual ini memakan waktu dan rawan kesalahan. Insinyur harus secara konsisten menerapkan perubahan yang sama di semua repositori yang ditargetkan dan membuat judul dan deskripsi PR yang bermakna. Selain itu, mereka sering harus berinteraksi dengan alat eksternal seperti Jira untuk mengambil atau menyertakan referensi pelacakan masalah. Tugas-tugas ini, meskipun perlu, adalah angkat berat yang tidak berdiferensiasi yang menghabiskan waktu teknik yang berharga dan mengurangi efisiensi secara keseluruhan. Kurangnya otomatisasi dalam alur kerja ini menciptakan gesekan, memperlambat pengiriman, dan meningkatkan beban kognitif pada tim yang bertugas memelihara infrastruktur Terraform skala besar.

Ikhtisar solusi

Untuk mengatasi tantangan ini, pola ini menawarkan utilitas yang sepenuhnya digerakkan oleh konfigurasi, memungkinkan pengguna untuk menentukan perubahan yang diinginkan dalam file konfigurasi terstruktur. File ini menentukan repositori target, modul, parameter, dan nilai menggunakan skema yang jelas.

Setelah dikonfigurasi, utilitas melakukan langkah-langkah otomatis berikut:

1. Membaca konfigurasi yang ditentukan pengguna untuk menentukan ruang lingkup dan sifat perubahan
2. Membuat cabang baru di setiap repositori target dengan pembaruan yang diperlukan diterapkan

3. Menghasilkan PR untuk setiap perubahan, memastikan konsistensi di semua repositori
4. Mengirim pemberitahuan Slack (opsional) untuk mengingatkan pemangku kepentingan dengan tautan langsung ke yang dibuat PRs

Dengan mengotomatiskan tugas-tugas berulang ini, utilitas secara signifikan mengurangi waktu, tenaga, dan risiko yang terkait dengan pengelolaan pembaruan infrastruktur skala besar. Ini memungkinkan tim untuk fokus pada pekerjaan teknik bernilai lebih tinggi sambil membantu memastikan bahwa perubahan diterapkan secara konsisten dan dapat ditelusuri di semua repositori.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Python versi 3.8 atau yang lebih baru.
- Token akses GitHub pribadi (PAT). Untuk informasi selengkapnya, lihat Membuat token akses pribadi (klasik) dalam GitHub dokumentasi.
- GitHub PAT dapat mengakses repositori target Anda sehingga utilitas dapat melakukan operasi seperti membuat cabang dan permintaan tarik. Untuk informasi lebih lanjut, lihat repositori GitHub kode pola ini.

Batasan

- Kompleksitas konfigurasi menghadirkan tantangan utama. Efektivitas otomatisasi dibatasi oleh kemampuan file konfigurasinya. Meskipun sistem menangani perubahan standar secara efisien, modifikasi infrastruktur yang kompleks mungkin memerlukan intervensi manual dan kasus tepi tertentu tetap berada di luar cakupan penanganan otomatis.
- Keamanan dan akses menyajikan pertimbangan yang signifikan terutama dalam mengelola token GitHub akses dan batas tarif API. Organizations harus secara hati-hati menyeimbangkan kebutuhan akan otomatisasi dengan penyimpanan dan manajemen kredensi yang aman, memastikan kontrol akses yang tepat sambil mempertahankan efisiensi operasional.
- Kendala validasi menimbulkan batasan penting lainnya karena sistem otomatis memiliki kemampuan terbatas untuk memvalidasi logika bisnis dan persyaratan khusus lingkungan. Ketergantungan yang kompleks dan interaksi lintas layanan sering kali memerlukan pengawasan manusia karena validasi otomatis tidak dapat sepenuhnya menangkap semua nuansa kontekstual dan aturan bisnis.

- Masalah skala dan kinerja muncul ketika berhadapan dengan perubahan infrastruktur skala besar. Sistem harus beroperasi dalam batas GitHub API sambil mengelola banyak repositori secara bersamaan. Operasi intensif sumber daya di seluruh infrastruktur yang luas dapat menciptakan kemacetan kinerja yang memerlukan manajemen yang cermat.
- Batas integrasi membatasi fleksibilitas sistem karena terutama dirancang untuk bekerja dengan alat khusus seperti GitHub dan Slack. Organizations yang menggunakan alat yang berbeda mungkin memerlukan solusi kustom dan opsi penyesuaian alur kerja dari pola ini terbatas pada titik integrasi yang didukung.

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen untuk solusi ini.

Alur kerja terdiri dari langkah-langkah berikut:

1. Pengembang memicu GitHub Tindakan dengan menentukan repositori Terraform.
2. Utilitas otomatisasi membaca konfigurasi yang ditentukan.
3. Utilitas otomatisasi juga menarik repositori Terraform yang disediakan.
4. Utilitas otomatisasi membuat cabang baru dan membuat pembaruan ke templat Terraform secara lokal.
5. Utilitas otomatisasi mendorong cabang baru ke repositori dan membuat PR baru.
6. Utilitas otomatisasi menggunakan pemberitahuan Slack yang menyertakan tautan PR untuk memberi tahu pengembang dan mengaktifkan templat Terraform untuk penerapan. AWS Cloud

Alat

- GitHubadalah platform pengembang yang dapat digunakan pengembang untuk membuat, menyimpan, mengelola, dan membagikan kode mereka.
- GitHub Actions adalah platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang terintegrasi erat dengan GitHub repositori. Anda dapat menggunakan GitHub Actions untuk mengotomatiskan pipeline build, test, dan deployment.
- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

- Slack, penawaran Salesforce, adalah platform percakapan bertenaga AI yang menyediakan kolaborasi obrolan dan video, mengotomatiskan proses tanpa kode, dan mendukung berbagi informasi.

Repositori kode

Kode untuk pola ini tersedia di Alur Kerja Pembaruan Infrastruktur Terraform GitHub Otomatis menggunakan GitHub repositori Tindakan.

Praktik terbaik

- Manajemen perubahan yang efektif sangat penting untuk implementasi yang sukses. Organizations harus mengadopsi strategi peluncuran bertahap untuk perubahan skala besar. Pertahankan konvensi penamaan cabang dan deskripsi PR yang konsisten dan pastikan dokumentasi komprehensif dari semua perubahan.
- Kontrol keamanan harus diterapkan secara ketat, dengan fokus pada prinsip akses hak istimewa dan manajemen kredensi yang aman. Aktifkan aturan perlindungan cabang untuk mencegah perubahan yang tidak sah. Melakukan audit keamanan secara teratur untuk menjaga integritas sistem.
- Protokol pengujian yang kuat harus mencakup `terraform plan` eksekusi otomatis dalam pipeline continuous integration dan continuous deployment (CI/CD). Protokol juga harus mencakup pemeriksaan validasi pra-komit, dan lingkungan tinjauan khusus untuk perubahan kritis. Pendekatan pengujian berlapis-lapis ini membantu menangkap masalah lebih awal dan memastikan stabilitas infrastruktur.
- Strategi pemantauan perlu mencakup mekanisme peringatan yang komprehensif, pelacakan success/failure metrik terperinci, dan mekanisme coba ulang otomatis untuk operasi yang gagal. Strategi ini membantu memastikan visibilitas operasional dan memungkinkan respons cepat terhadap masalah apa pun yang muncul.
- Standar konfigurasi harus menekankan kontrol versi untuk semua konfigurasi, mempertahankan modularitas untuk kegunaan kembali dan skalabilitas. Dokumentasi skema dan contoh yang jelas membantu tim memahami dan menggunakan sistem otomasi secara efektif.

Epik

Instalasi dan pengaturan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan repositori.	Untuk mengatur repositori, jalankan perintah berikut:
	`# Clone the automation tool repository`
	`git clone https://github.com/aws-samples/sample-terraform-pr-automation-utility`
	`cd sample-terraform-pr-automation-utility`
	
	`# Copy example configuration`
	`cp config.example.yaml config.yaml`
		AWS DevOps

Instal dependensi.	Untuk menginstal dan memverifikasi dependensi Python, jalankan perintah berikut:
	`# Install Python dependencies`
	`pip3 install -r requirements.txt`
	
	`# Verify installation`
	`python3 -c "import github; import hcl2; import yaml; import requests; print('All packages installed successfully')"`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan GitHub token.	Untuk mengonfigurasi GitHub token dan kemudian memverifikasi bahwa itu berfungsi, jalankan perintah berikut:
	`# Set GitHub token environment variable`
	`export GITHUB_TOKEN="your_github_token_here"`
	
	`# Verify token works`
	`curl -H "Authorization: token $GITHUB_TOKEN" https://api.github.com/user`
		AWS DevOps

Siapkan file konfigurasi untuk perubahan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan `config.yaml` file.	Untuk menentukan repositori target Anda dan perubahan yang diinginkan, edit file `config.yaml` sebagai berikut:
	`repositories:`
	`  - owner: "your-org"`
	`  repo: "your-terraform-repo"`
	`  files:`
	`  - path: "variables.tf"`
	`  changes:`
	`  variables:`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>- app_versi
	on:
	default:
	
	update:
	-
	from: ["1.0.0"]
	
	to: "1.1.0"
	
	settings:
	pr_title_template:
	"Infrastructure Update
	- {{timestamp}}"
	slack:
	username: "Terrafor
	m Bot"
	icon_emoji:
	":terraform:"
	notify_on_success:
	true
	notify_on_error:
	true
	notify_batch_summa
	ry: true</pre>	

Uji dan validasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lakukan pengujian pra-penerbangan.	Selalu uji konfigurasi Anda sebelum menjalankannya di repositori produksi. Gunakan salah satu perintah berikut ini:
	<pre># 1. Test configuration
	syntax
	python3 -c "from
	main import get_confi</pre>
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`g_content; get_confi g_content()"`
	
	`# 2. Run in dry-run mode first`
	`DRY_RUN=true python3 main.py`
	
	`# 3. Test with minimal configuration`
	`# Use a simple config.ya ml with just one repository and one change`	

Verifikasi akses repositori.	Untuk memverifikasi bahwa GitHub token dapat mengakses repositori, jalankan perintah berikut:
	
	`# Test GitHub token access`
	`curl -H "Authoriz ation: token $GITHUB_T OKEN" \`
	`https://api.github .com/repos/owner/r epo-name`
	
	`# Should return repository informati on, not 404`
		AWS DevOps

Jalankan utilitas otomatisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan utilitas otomatisasi dengan menggunakan GitHub Actions UI.	Untuk menjalankan utilitas otomatisasi menggunakan GitHub Actions UI, lakukan hal berikut: 1. Arahkan ke repositori Anda di. GitHub 2. Pilih tab Tindakan. 3. Pilih alur kerja Otomasi Pembaruan Infrastruktur Terraform. 4. Pilih Jalankan alur kerja. 5. Konfigurasikan input alur kerja berikut: • Konfigurasi sumber: • Pilih cabang target untuk otomatisasi. • Tentukan jalur ke file konfigurasi (`config.yaml`) yang berisi perubahan yang diinginkan. • Kontrol pratinjau: • Pilih opsi Pratinjau untuk meninjau perubahan tanpa menerapkannya. • Manajemen cabang: • Masukkan cabang dasar untuk membuat cabang fitur baru.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Masukkan konfigurasi awalan cabang. • Pilih kotak centang Permintaan tarik usang secara otomatis. • Pengaturan pemberita huan: • Masukkan URL Anda untuk URL webhook Slack untuk notifikas i kaya (mengganti rahasia repositori). • Pilih kotak centang Test Slack integration before processing (dry run only). • Masukkan informasi Anda untuk saluran Slack default Override (misalnya, #infrastr ucture -test). • Pengaturan lanjutan: • Pilih kotak centang Aktifkan pencatata n debug untuk pemecahan masalah.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

(Alternatif) Jalankan utilitas otomatisasi dari baris perintah.	Jika mau, Anda dapat menjalankan utilitas otomatisasi dari baris perintah alih-alih dengan menggunakan UI GitHub Tindakan. Gunakan perintah berikut ini.
	`# Run actual automation`
	`python3 main.py`
		AWS DevOps

Validasi PRs dan perubahan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau yang dibuat PRs dan perubahan.	Untuk memantau hasil eksekusi GitHub alur kerja, lakukan hal berikut: • Periksa alur kerja menjalankan log untuk status pemrosesan. • Tinjau apa PRs yang dibuat oleh utilitas otomatisasi. • Pantau pemberitahuan Slack (jika dikonfigurasi).	AWS DevOps

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Bersihkan PRs.	Tutup ditinggalkan atau tidak perlu PRs.	AWS DevOps

Sumber daya terkait

AWS Bimbingan Preskriptif

- Menggunakan Terraform sebagai alat IAc untuk AWS Cloud

GitHub dokumentasi

- Tentang permintaan tarik
- Mengelola token akses pribadi Anda
- Memahami GitHub Tindakan
- Quickstart untuk Tindakan GitHub

Buat pipeline CI dinamis untuk proyek Java dan Python secara otomatis

Aromal Raj Jayarajan, Vijesh Vijayakumaran Nair, MAHESH RAGHUNANDANAN, dan Amarnath Reddy, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara membuat pipeline dynamic continuous integration (CI) untuk proyek Java dan Python secara otomatis dengan menggunakan alat pengembang AWS.

Ketika tumpukan teknologi terdiversifikasi dan aktivitas pengembangan meningkat, akan menjadi sulit untuk membuat dan memelihara jaringan pipa CI yang konsisten di seluruh organisasi. Dengan mengotomatiskan proses di AWS Step Functions, Anda dapat memastikan bahwa pipeline CI Anda konsisten dalam penggunaan dan pendekatannya.

Untuk mengotomatiskan pembuatan pipeline CI dinamis, pola ini menggunakan input variabel berikut:

- Bahasa pemrograman (hanya Java atau Python)
- Nama pipa
- Tahapan pipa yang dibutuhkan

> **Note**
>
> Step Functions mengatur pembuatan pipeline dengan menggunakan beberapa layanan AWS. Untuk informasi selengkapnya tentang layanan AWS yang digunakan dalam solusi ini, lihat bagian Alat dari pola ini.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Bucket Amazon S3 di Wilayah AWS yang sama dengan solusi ini sedang diterapkan
- Prinsip AWS Identity and Access Management (IAM) yang memiliki CloudFormation izin AWS yang diperlukan untuk membuat sumber daya yang diperlukan untuk solusi ini

Batasan

- Pola ini hanya mendukung proyek Java dan Python.
- Peran IAM yang disediakan dalam pola ini mengikuti prinsip hak istimewa paling sedikit. Izin peran IAM harus diperbarui berdasarkan sumber daya spesifik yang perlu dibuat oleh pipeline CI Anda.

Arsitektur

Tumpukan teknologi target

- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- IAM
- Amazon Simple Storage Service (Amazon S3)
- AWS Systems Manager
- AWS Step Functions
- AWS Lambda
- Amazon DynamoDB

Arsitektur target

Diagram berikut menunjukkan contoh alur kerja untuk membuat pipeline CI dinamis untuk proyek Java dan Python secara otomatis dengan menggunakan alat pengembang AWS.

Diagram menunjukkan alur kerja berikut:

1. Pengguna AWS menyediakan parameter input untuk pembuatan pipeline CI dalam format JSON. Input ini memulai alur kerja Step Functions (state machine) yang membuat pipeline CI dengan menggunakan alat pengembang AWS.
2. Fungsi Lambda membaca folder bernama input-reference, yang disimpan dalam bucket Amazon S3, dan kemudian menghasilkan file buildspec.yml. File yang dihasilkan ini menentukan tahapan pipeline CI dan disimpan kembali di bucket Amazon S3 yang sama yang menyimpan referensi parameter.

3. Step Functions memeriksa dependensi alur kerja pembuatan pipeline CI untuk setiap perubahan, dan memperbarui tumpukan dependensi sesuai kebutuhan.
4. Step Functions membuat sumber daya pipeline CI dalam CloudFormation tumpukan, termasuk CodeCommit repositori, CodeBuild proyek, dan pipeline. CodePipeline
5. CloudFormation Tumpukan menyalin kode sumber sampel untuk tumpukan teknologi yang dipilih (Java atau Python) dan file buildspec.yl ke repositori. CodeCommit
6. Rincian runtime pipeline CI disimpan dalam tabel DynamoDB.

Otomatisasi dan skala

- Pola ini hanya untuk digunakan dalam satu lingkungan pengembangan saja. Perubahan konfigurasi diperlukan untuk digunakan di beberapa lingkungan pengembangan.
- Untuk menambahkan dukungan untuk lebih dari satu CloudFormation tumpukan, Anda dapat membuat CloudFormation template tambahan. Untuk informasi selengkapnya, lihat Memulai AWS CloudFormation di CloudFormation dokumentasi.

Alat

Alat

- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi AWS Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS CodeBuild adalah layanan build terkelola penuh yang membantu Anda mengompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodePipeline membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Systems Manager Parameter Store menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.

Kode

Kode untuk pola ini tersedia di GitHub automated-ci-pipeline-creationrepositori. Repositori berisi CloudFormation template yang diperlukan untuk membuat arsitektur target yang diuraikan dalam pola ini.

Praktik terbaik

- Jangan masukkan kredensil (rahasia) seperti token atau kata sandi langsung ke CloudFormation templat atau konfigurasi tindakan Step Functions. Jika Anda melakukannya, informasi akan ditampilkan di log DynamoDB. Sebagai gantinya, gunakan AWS Secrets Manager untuk menyiapkan dan menyimpan rahasia. Kemudian, rujuk rahasia yang disimpan di Secrets Manager dalam CloudFormation template dan konfigurasi tindakan Step Functions sesuai kebutuhan. Untuk informasi selengkapnya, lihat Apa itu AWS Secrets Manager dalam dokumentasi Secrets Manager.
- Konfigurasikan enkripsi sisi server untuk CodePipeline artefak yang disimpan di Amazon S3. Untuk informasi selengkapnya, lihat Mengonfigurasi enkripsi sisi server untuk artefak yang disimpan di Amazon S3 dalam dokumentasi. CodePipeline CodePipeline
- Terapkan izin hak istimewa paling sedikit saat mengonfigurasi peran IAM. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM.
- Pastikan bucket Amazon S3 Anda tidak dapat diakses publik. Untuk informasi selengkapnya, lihat Mengonfigurasi setelan blokir akses publik untuk bucket S3 Anda di dokumentasi Amazon S3.

- Pastikan Anda mengaktifkan pembuatan versi untuk bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat Menggunakan pembuatan versi di bucket S3 di dokumentasi Amazon S3.
- Gunakan IAM Access Analyzer saat mengonfigurasi kebijakan IAM. Alat ini memberikan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan IAM yang aman dan fungsional. Untuk informasi selengkapnya, lihat Menggunakan AWS Identity and Access Management Access Analyzer dalam dokumentasi IAM.
- Jika memungkinkan, tentukan kondisi akses tertentu saat mengonfigurasi kebijakan IAM.
- Aktifkan CloudWatch pencatatan Amazon untuk tujuan pemantauan dan audit. Untuk informasi selengkapnya, lihat Apa itu Amazon CloudWatch Logs? dalam CloudWatch dokumentasi.

Epik

Konfigurasikan prasyarat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket Amazon S3.	Buat bucket Amazon S3 (atau gunakan bucket yang sudah ada) untuk menyimpan CloudFormation template, kode sumber, dan file input yang diperlukan untuk solusi tersebut. Untuk informasi selengkap nya, lihat Langkah 1: Membuat bucket S3 pertama Anda di dokumentasi Amazon S3. **Note** Bucket Amazon S3 harus berada di Wilayah AWS yang sama dengan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang Anda gunakan solusinya.	
Kloning GitHub repositori.	Kloning GitHub automated-ci-pipeline-creationrepositori dengan menjalankan perintah berikut di jendela terminal: `git clone https://github.com/aws-samples/automated-ci-pipeline-creation.git` Untuk informasi selengkapnya, lihat Mengkloning repositori dalam dokumentasi. GitHub	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah folder Solution Templates dari GitHub repositori kloning ke bucket Amazon S3 Anda.	Salin konten dari folder Solution-Templates yang dikloning dan unggah ke bucket Amazon S3 yang Anda buat. Untuk informasi selengkap nya, lihat Mengunggah objek di dokumentasi Amazon S3. **Note** Pastikan Anda mengunggah konten folder Solution-Templates saja. Anda dapat mengunggah file di tingkat root bucket Amazon S3 saja.	AWS DevOps

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan untuk menerapkan solusi dengan menggunakan file template.ymldi repositori kloning. GitHub	1. Masuk ke AWS Management Console lalu buka CloudFormation konsol AWS. 2. Pilih Buat tumpukan. Daftar dropdown muncul. 3. Dalam daftar dropdown, pilih Dengan sumber daya baru (standar). Halaman Buat tumpukan terbuka.	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Di bagian Tentukan templat, pilih kotak centang di samping Unggah file templat. 5. Pilih Pilih file. Kemudian, navigasikan ke folder root GitHub repositori yang dikloning dan pilih file template.yml. Kemudian, pilih Buka. 6. Pilih Berikutnya. Halaman Tentukan detail tumpukan terbuka. 7. Di bagian Parameter, tentukan parameter berikut: • Untuk S3 TemplateB ucketName, masukkan nama bucket Amazon S3 yang Anda buat sebelumnya, yang berisi kode sumber dan referensi untuk solusi ini. Pastikan parameter nama bucket dalam huruf kecil. • Untuk Dynamo DBTable, masukkan nama untuk tabel DynamoDB yang dibuat tumpukan. CloudFormation • Untuk StateMach ineName, masukkan nama untuk mesin status Step Functions yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dibuat oleh CloudForm ation stack. 8. Pilih Berikutnya. Halaman Configure stack options terbuka. 9. Pada halaman Konfigura sikan opsi tumpukan, pilih Berikutnya. Jangan mengubah nilai default apa pun. Halaman Review terbuka. 10. Tinjau pengaturan pembuatan tumpukan. Kemudian, pilih Buat tumpukan untuk meluncurk an tumpukan Anda. **Note** Saat tumpukan Anda sedang dibuat, itu terdaftar di halaman Stacks dengan status CREATE_IN _PROGRESS . Pastikan Anda menunggu status tumpukan berubah menjadi CREATE_CO MPLETE sebelum menyelesaikan langkah-langkah yang tersisa dalam pola ini.	

Uji penyiapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan fungsi langkah yang Anda buat.	1. Masuk ke AWS Management Console, lalu buka konsol Step Functions . 2. Buka fungsi langkah yang Anda buat. 3. Pilih Mulai Eksekusi. Kemudian, masukkan nilai input Anda untuk alur kerja dalam format JSON (lihat contoh input berikut). 4. Pilih Mulai Eksekusi. Pemformatan JSON `{ "details": { "tech_stack": "Name of the Tech Stack (python/java)", "project_name": "Name of the Project that you want to create with", "pre_build": "Choose the step if it required in the buildspec.yml file i.e., yes/no", "build": "Choose the step if it required in the buildspec.yml file i.e., yes/no", "post_build": "Choose the step if`	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

```
 it required in the
 buildspec.yml file
 i.e., yes/no",
    "reports": "Choose
 the step if it required
 in the buildspec.yml
 file i.e., yes/no",
  }
}
```

Contoh masukan JSON Java

```
{
  "details": {
    "tech_stack":
 "java",
    "project_name":
 "pipeline-java-pjt",
    "pre_build": "yes",
    "build": "yes",
    "post_build":
 "yes",
    "reports": "yes"
  }
}
```

Contoh masukan Python JSON

```
{
  "details": {
    "tech_stack":
 "python",
    "project_name":
 "pipeline-python-p
jt",
    "pre_build": "yes",
    "build": "yes",
    "post_build":
 "yes",
```

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"reports": "yes"
	}
	}</pre>	

Konfirmasikan bahwa CodeCommit repositori untuk pipa CI telah dibuat.	1. Masuk ke AWS Management Console lalu buka CodeCommit konsol. 2. Pada halaman Repositori, verifikasi bahwa nama CodeCommit repositori yang Anda buat muncul dalam daftar repositori. Nama repositori ditambahkan dengan yang berikut: - Repo pipeline-java-pjt 3. Buka CodeCommit repositori dan validasi bahwa kode sumber sampel bersama dengan file buildspec.yl didorong ke cabang utama.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa sumber daya CodeBuild proyek.	1. Masuk ke AWS Management Console lalu buka CodeBuild konsol. 2. Pada halaman Membangun proyek, verifikasi bahwa nama CodeBuild proyek yang Anda buat muncul dalam daftar proyek. Nama proyek ditambahk an dengan yang berikut: pipeline-java-pjt -Build 3. Pilih nama CodeBuild proyek Anda untuk membuka proyek. Kemudian, tinjau dan validasi konfigurasi berikut: • Konfigurasi Proyek • Sumber • Lingkungan • Buildspec • Konfigurasi Batch • Artefak	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi CodePipeline tahapan.	1. Masuk ke AWS Management Console lalu buka CodePipeline konsol. 2. Pada halaman Pipelines , verifikasi bahwa nama pipeline yang Anda buat muncul di daftar pipeline. Nama pipa ditambahk an dengan yang berikut: pipeline-java-pjt -Pipeline 3. Pilih nama pipeline Anda untuk membuka pipeline. Kemudian, tinjau dan validasi setiap tahap pipeline, termasuk Commit dan Deploy.	AWS DevOps
Konfirmasikan bahwa pipa CI berhasil berjalan.	1. Di CodePipeline konsol, pada halaman Pipelines, pilih nama pipeline Anda untuk melihat status pipeline. 2. Verifikasi bahwa setiap tahap pipa memiliki status Berhasil.	AWS DevOps

Bersihkan sumber daya Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus tumpukan sumber daya di CloudFormation.	Hapus tumpukan sumber daya pipa CI di CloudFormation.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya, lihat Menghapus tumpukan di CloudFormation konsol AWS dalam CloudFormation dokumentasi. **Note** Pastikan Anda menghapus tumpukan bernama - stack<project_name>.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus dependensi pipeline CI di Amazon S3 dan. CloudFormation	1. Kosongkan ember Amazon S3 bernama. DeploymentArtifactBucket Untuk informasi selengkapnya, lihat Mengosongkan bucket di dokumentasi Amazon S3. 2. Hapus tumpukan ketergantungan pipa CI di CloudFormation. Untuk informasi selengkapnya, lihat Menghapus tumpukan di CloudFormation konsol AWS dalam CloudFormation dokumentasi. **Note** Pastikan Anda menghapus tumpukan bernama pipeline-creation-dependencies-stack.	AWS DevOps
Hapus bucket template Amazon S3.	Hapus bucket Amazon s3 yang Anda buat di bagian Konfigurasikan prasyarat dari pola ini, yang menyimpan templat untuk solusi ini. Untuk informasi selengkapnya, lihat Menghapus bucket di dokumentasi Amazon S3.	AWS DevOps

Sumber daya terkait

- Membuat mesin status Step Functions yang menggunakan Lambda (dokumentasi AWS Step Functions)
- AWS Step Functions WorkFlow Studio (dokumentasi AWS Step Functions)
- DevOps dan AWS
- Bagaimana cara CloudFormation kerja AWS? (CloudFormation Dokumentasi AWS)
- Lengkap CI/CD dengan AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, dan AWS CodePipeline (postingan blog AWS)
- Kuota IAM dan AWS STS, persyaratan nama, dan batas karakter (dokumentasi IAM)

Menyebarkan kenari CloudWatch Synthetics dengan menggunakan Terraform

Dhrubajyoti Mukherjie dan Jean-Francois Landerau, Amazon Web Services

Ringkasan

Penting untuk memvalidasi kesehatan sistem dari perspektif pelanggan dan mengonfirmasi bahwa pelanggan dapat terhubung. Ini lebih sulit ketika pelanggan tidak terus-menerus memanggil titik akhir. Amazon CloudWatch Synthetics mendukung pembuatan kenari, yang dapat menguji titik akhir publik dan pribadi. Dengan menggunakan kenari, Anda dapat mengetahui status suatu sistem meskipun tidak digunakan. Burung kenari ini adalah skrip Dalang Node.js atau skrip Python Selenium.

Pola ini menjelaskan cara menggunakan HashiCorp Terraform untuk menyebarkan kenari yang menguji titik akhir pribadi. Ini menyematkan skrip Dalang yang menguji apakah URL kembali. `200-OK` Skrip Terraform kemudian dapat diintegrasikan dengan skrip yang menyebarkan titik akhir pribadi. Anda juga dapat memodifikasi solusi untuk memantau titik akhir publik.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) aktif dengan virtual private cloud (VPC) dan subnet pribadi
- URL titik akhir yang dapat dicapai dari subnet pribadi
- Terraform diinstal di lingkungan penerapan

Keterbatasan

Solusi saat ini berfungsi untuk versi runtime CloudWatch Synthetics berikut:

- syn-nodejs-puppeteer-3,4
- syn-nodejs-puppeteer-3,5
- syn-nodejs-puppeteer-3,6
- syn-nodejs-puppeteer-3,7

Saat versi runtime baru dirilis, Anda mungkin perlu memperbarui solusi saat ini. Anda juga perlu memodifikasi solusi untuk mengikuti pembaruan keamanan.

Versi produk

- Terraform 1.3.0

Arsitektur

Amazon CloudWatch Synthetics didasarkan pada, CloudWatch Lambda, dan Amazon Simple Storage Service (Amazon S3). Amazon CloudWatch menawarkan wizard untuk membuat kenari dan dasbor yang menampilkan status kenari berjalan. Fungsi Lambda menjalankan skrip. Amazon S3 menyimpan log dan tangkapan layar dari kenari berjalan.

Pola ini mensimulasikan titik akhir pribadi melalui instans Amazon Elastic Compute Cloud (Amazon EC2) yang diterapkan di subnet yang ditargetkan. Fungsi Lambda memerlukan antarmuka jaringan elastis di VPC tempat titik akhir pribadi digunakan.

Diagram menunjukkan yang berikut:

1. Canary Synthetics memulai fungsi Lambda kenari.
2. Fungsi Canary Lambda terhubung ke interface elastic network.
3. Fungsi Lambda kenari memonitor status titik akhir.
4. Canary Synthetics mendorong data run ke bucket dan metrik S3. CloudWatch
5. CloudWatch Alarm dimulai berdasarkan metrik.
6. CloudWatch Alarm memulai topik Amazon Simple Notification Service (Amazon SNS).

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber daya AWS Anda dan aplikasi yang Anda jalankan di AWS secara real time.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan. Pola ini menggunakan titik akhir VPC dan antarmuka jaringan elastis.

Layanan lainnya

- HashiCorp Terraform adalah alat infrastruktur sumber terbuka sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud. Pola ini menggunakan Terraform untuk menyebarkan infrastruktur.
- Puppeteer adalah perpustakaan Node.js. Runtime CloudWatch Synthetics menggunakan framework Puppeteer.

Kode

Solusinya tersedia di watch-synthetics-canary-terraform repositori GitHub cloud. Untuk informasi selengkapnya, lihat bagian Informasi tambahan.

Epik

Menerapkan solusi untuk memantau URL pribadi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan persyaratan untuk memantau URL pribadi.	Kumpulkan definisi URL lengkap: domain, parameter, dan header. Untuk berkomunikasi secara pribadi ke Amazon S3 dan CloudWatch Amazon, gunakan titik akhir VPC. Perhatikan bagaimana VPC dan subnet dapat diakses ke titik akhir. Pertimbangkan frekuensi lari kenari.	Arsitek awan, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah solusi yang ada untuk memantau URL pribadi.	Ubah `terraform.tfvars` file: • `name`— Nama kenari Anda. • `runtime_version` — Versi runtime dari kenari. Kami merekomendasikan menggunakan syn-nodejs-puppeteer -3.7. • `take_screenshot` — Apakah tangkapan layar harus diambil. • `api_hostname` — Nama host dari titik akhir yang dipantau. • `api_path`— Jalur titik akhir yang dipantau. • `vpc_id`— ID VPC yang digunakan oleh fungsi Lambda kenari. • `subnet_ids` — Subnet IDs yang digunakan oleh fungsi Lambda kenari. • `frequency` — Frekuensi lari kenari dalam hitungan menit. • `alert_sns_topic` — Topik SNS tempat pemberitahuan CloudWatch alarm dikirim.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan dan mengoperasikan solusi.	Untuk menerapkan solusi, lakukan hal berikut:
	1. Dari `cloudwatch-synthetics-canary-terraform` direktori di lingkungan pengembangan Anda, inisialisasi Terraform.
	`terraform init`
	2. Rencanakan dan tinjau perubahan.
	`terraform plan`
	3. Menyebarkan solusinya.
	`terraform apply`
		Arsitek awan, DevOps insinyur

Pemecahan masalah

Isu	Solusi
Penghapusan sumber daya yang disediakan macet.	Hapus fungsi Lambda kenari secara manual, elastis network interface yang sesuai, dan grup keamanan, dalam urutan itu.

Sumber daya terkait

- Menggunakan pemantauan sintetis
- Pantau titik akhir API Gateway dengan Amazon CloudWatch Synthetics (posting blog)

Informasi tambahan

Artefak repositori

Artefak repositori berada dalam struktur berikut.

```
.
### README.md
### main.tf
### modules
#   ### canary
#   ### canary-infra
### terraform.tfvars
### tf.plan
### variable.tf
```

`main.tf`File berisi modul inti, dan menyebarkan dua submodul:

- `canary-infra`menyebarkan infrastruktur yang dibutuhkan untuk burung kenari.
- `canary`menyebarkan burung kenari.

Parameter input untuk solusi terletak di `terraform.tfvars` file. Anda dapat menggunakan contoh kode berikut untuk membuat satu kenari.

```
module "canary" {
    source = "./modules/canary"
    name   = var.name
    runtime_version = var.runtime_version
    take_screenshot = var.take_screenshot
    api_hostname = var.api_hostname
    api_path = var.api_path
    reports-bucket = module.canary_infra.reports-bucket
    role = module.canary_infra.role
    security_group_id = module.canary_infra.security_group_id
    subnet_ids = var.subnet_ids
    frequency = var.frequency
    alert_sns_topic = var.alert_sns_topic
}
```

File.var yang sesuai mengikuti.

```
name   = "my-canary"
runtime_version = "syn-nodejs-puppeteer-3.7"
take_screenshot = false
api_hostname = "mydomain.internal"
api_path = "/path?param=value"
vpc_id = "vpc_id"
subnet_ids = ["subnet_id1"]
frequency = 5
alert_sns_topic = "arn:aws:sns:eu-central-1:111111111111:yyyyy"
```

Membersihkan solusinya

Jika Anda menguji ini di lingkungan pengembangan, Anda dapat membersihkan solusi untuk menghindari biaya yang masih harus dibayar.

1. Di AWS Management Console, navigasikan ke konsol Amazon S3. Kosongkan bucket Amazon S3 yang dibuat oleh solusi. Pastikan untuk mengambil cadangan data, jika diperlukan.
2. Di lingkungan pengembangan Anda, dari `cloudwatch-synthetics-canary-terraform` direktori, jalankan `destroy` perintah.

   ```
   terraform destroy
   ```

Terapkan ChatOps solusi untuk mengelola hasil pemindaian SAST dengan menggunakan Amazon Q Developer dalam tindakan kustom aplikasi obrolan dan CloudFormation

Anand Bukkapatnam Tirumala, Amazon Web Services

Ringkasan

Pola ini menyajikan solusi komprehensif yang menggunakan Pengembang Amazon Q dalam aplikasi obrolan untuk merampingkan pengelolaan kegagalan pemindaian pengujian keamanan aplikasi statis (SAST) yang dilaporkan melalui. SonarQube Pendekatan inovatif ini mengintegrasikan tindakan dan pemberitahuan khusus ke dalam antarmuka percakapan, memungkinkan kolaborasi yang efisien dan proses pengambilan keputusan dalam tim pengembangan.

Dalam lingkungan pengembangan perangkat lunak yang serba cepat saat ini, mengelola hasil pemindaian SAST secara efisien sangat penting untuk menjaga kualitas dan keamanan kode. Namun, banyak organisasi menghadapi tantangan signifikan berikut:

- Kesadaran yang tertunda akan kerentanan kritis karena sistem notifikasi yang tidak efisien
- Proses pengambilan keputusan yang lambat disebabkan oleh alur kerja persetujuan yang terputus
- Kurangnya respons langsung dan dapat ditindaklanjuti terhadap kegagalan pemindaian SAST
- Komunikasi dan kolaborasi yang terfragmentasi seputar temuan keamanan
- Pengaturan infrastruktur manual yang memakan waktu dan rawan kesalahan untuk perkakas keamanan

Masalah-masalah ini sering menyebabkan peningkatan risiko keamanan, rilis yang tertunda, dan penurunan produktivitas tim. Untuk mengatasi tantangan ini secara efektif diperlukan solusi yang dapat merampingkan manajemen hasil SAST, meningkatkan kolaborasi tim, dan mengotomatiskan penyediaan infrastruktur.

Fitur utama dari solusi ini meliputi:

- Pemberitahuan yang disesuaikan — Peringatan dan notifikasi real-time dikirimkan langsung ke saluran obrolan tim, memastikan kesadaran dan tindakan yang cepat pada kerentanan atau kegagalan pemindaian SAST.

- Persetujuan percakapan - Pemangku kepentingan dapat memulai dan menyelesaikan alur kerja persetujuan untuk hasil pemindaian SAST dengan mulus dalam antarmuka obrolan, mempercepat proses pengambilan keputusan.
- Tindakan kustom — Tim dapat menentukan dan menjalankan tindakan khusus berdasarkan hasil pemindaian SAST, seperti secara otomatis memicu pesan email untuk kegagalan gerbang kualitas, meningkatkan respons terhadap masalah keamanan.
- Kolaborasi terpusat - Semua diskusi, keputusan, dan tindakan terkait pemindaian SAST disimpan dalam lingkungan obrolan terpadu, mendorong peningkatan kolaborasi dan berbagi pengetahuan di antara anggota tim.
- Infrastructure as code (IAc) — Seluruh solusi dibungkus dengan AWS CloudFormation template, memungkinkan penyediaan infrastruktur yang lebih cepat dan lebih andal sekaligus mengurangi kesalahan pengaturan manual.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Peran AWS Identity and Access Management (IAM) dengan izin untuk membuat dan mengelola sumber daya yang terkait dengan yang Layanan AWS tercantum di Alat.
- Ruang kerja Slack.
- Pengembang Amazon Q dalam aplikasi obrolan ditambahkan ke ruang kerja Slack yang diperlukan sebagai plugin. Untuk informasi selengkapnya, lihat Menambahkan aplikasi ke ruang kerja Slack di dokumentasi Slack. Catat ID ruang kerja Slack seperti yang ditunjukkan pada pendaftaran Konsol Manajemen AWS setelah berhasil.
- Pengembang Amazon Q yang dikonfigurasi di klien aplikasi obrolan, dengan ID ruang kerja yang tersedia untuk dimasukkan di CloudFormation konsol. Untuk petunjuk, lihat Mengkonfigurasi klien Slack di Pengembang Amazon Q di Panduan Administrator aplikasi obrolan.
- Akun email sumber yang dibuat dan diverifikasi di Amazon Simple Email Service (Amazon SES) untuk mengirimkan pesan email persetujuan. Untuk petunjuk penyiapan, lihat Membuat dan memverifikasi identitas email di Panduan Pengembang Layanan Email Sederhana Amazon.
- Alamat email tujuan untuk menerima pemberitahuan persetujuan. Alamat ini dapat berupa kotak masuk bersama atau daftar distribusi tim tertentu.
- SonarQube Contoh operasional yang dapat diakses dari Anda Akun AWS. Untuk informasi selengkapnya, lihat petunjuk SonarQube penginstalan.

- Token SonarQube pengguna dengan izin untuk memicu dan membuat proyek melalui pipeline.

Batasan

- Pembuatan tombol tindakan khusus adalah proses manual dalam solusi ini.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Diagram menunjukkan alur kerja jaminan kualitas kode otomatis:

1. Persiapan dan unggah kode:
   - Pengembang mengompres basis kode menjadi file.zip.
   - Pengembang secara manual mengunggah file.zip ke bucket Amazon Simple Storage Service (Amazon S3) yang ditunjuk.
2. Pemicu dan orkestrasi peristiwa Amazon S3: AWS Step Functions
   - Peristiwa upload Amazon S3 memicu alur kerja Step Functions.
   - Step Functions mengatur pemindaian SAST menggunakan. SonarQube
   - Alur kerja memantau status AWS CodeBuild pekerjaan untuk menentukan tindakan selanjutnya. Jika CodeBuild berhasil (quality gate pass), alur kerja berakhir. Jika CodeBuild gagal, AWS Lambda fungsi dipanggil untuk diagnostik. Untuk lebih jelasnya, lihat AWS Step Functions logika nanti di bagian ini.
3. AWS CodeBuild eksekusi:
   - CodeBuild Pekerjaan mengeksekusi SonarQube pemindaian pada basis kode yang diunggah.
   - Artefak pemindaian disimpan dalam bucket Amazon S3 terpisah untuk audit dan analisis.
4. Analisis kegagalan (fungsi Lambda):
   - Pada CodeBuild kegagalan, fungsi `CheckBuildStatus` Lambda dipicu.
   - Setelah CodeBuild sukses, proses dihentikan dan tidak diperlukan tindakan lebih lanjut.

5. Fungsi Lambda menganalisis penyebab kegagalan (kegagalan gerbang kualitas atau masalah lainnya)
   - `CheckBuildStatus`Fungsi ini membuat muatan khusus dengan informasi kegagalan terperinci.
   - `CheckBuildStatus`Fungsi ini menerbitkan payload khusus ke topik Amazon Simple Notification Service (Amazon SNS).
6. Sistem pemberitahuan:
   - Amazon SNS meneruskan muatan ke Pengembang Amazon Q dalam aplikasi obrolan untuk integrasi Slack.
7. Integrasi kendur:
   - Pengembang Amazon Q dalam aplikasi obrolan memposting pemberitahuan di saluran Slack yang ditunjuk.
8. Proses persetujuan:
   - Pemberi persetujuan meninjau detail kegagalan dalam pemberitahuan Slack.
   - Pemberi persetujuan dapat memulai persetujuan menggunakan tombol Approve di Slack.
9. Penangan persetujuan:
   - Fungsi Lambda Persetujuan memproses tindakan persetujuan dari Slack.
   - Fungsi Persetujuan menerbitkan pesan khusus ke Amazon SES.
10. Pesan yang dihasilkan:
   - Fungsi Persetujuan menghasilkan pesan khusus untuk pemberitahuan pengembang.
11. Pemberitahuan pengembang:
   - Amazon SES mengirimkan pesan email ke pengembang dengan langkah berikutnya atau tindakan yang diperlukan.

Alur kerja ini menggabungkan pengunggahan kode manual dengan pemeriksaan kualitas otomatis, memberikan umpan balik langsung melalui Slack, dan memungkinkan intervensi manusia bila diperlukan, memastikan proses peninjauan kode yang kuat dan fleksibel.

AWS Step Functions logika

Seperti yang ditunjukkan pada diagram arsitektur sebelumnya, jika gerbang kualitas diteruskan SonarQube gagal, alur kerja masuk ke fungsi `CheckBuildStatus` Lambda. `CheckBuildStatus`Fungsi ini memicu pemberitahuan di saluran Slack. Setiap pemberitahuan

mencakup informasi dengan langkah-langkah berikutnya yang disarankan. Berikut ini adalah jenis notifikasi:

- Aplikasi telah gagal dalam pemindaian keamanan kode — Pengguna menerima pemberitahuan ini ketika kode yang diunggah tidak lulus pemindaian SonarQube keamanan. Pengguna dapat memilih APPROVE untuk menerima build. Namun, pemberitahuan tersebut menyarankan pengguna untuk berhati-hati terhadap potensi kualitas kode yang buruk dan risiko keamanan. Pemberitahuan tersebut mencakup detail berikut:
  - Langkah selanjutnya: Kesalahan: Status gerbang kualitas: GAGAL — Lihat detail di URL yang disediakan.
  - Triase kerentanan seperti yang disebutkan dalam dokumen di URL yang disediakan.
  - CodeBuild rincian tersedia di lokasi di URL yang disediakan.
- Pipeline pemindaian aplikasi telah gagal karena beberapa alasan lain - Pengguna menerima pemberitahuan ini ketika pipa gagal karena beberapa alasan selain gagal pemindaian keamanan kode. Pemberitahuan tersebut mencakup detail berikut:
  - Untuk langkah selanjutnya, buka tautan yang disediakan untuk pemecahan masalah lebih lanjut.

Untuk melihat tangkapan layar notifikasi saat muncul di saluran Slack, buka folder aset di repositori GitHub chatops-slack.

Diagram berikut menunjukkan contoh status langkah Step Functions setelah quality gate pass gagal.

Alat

Layanan AWS

- Pengembang Amazon Q dalam aplikasi obrolan memungkinkan Anda menggunakan saluran obrolan Amazon Chime, Microsoft Teams, dan Slack untuk memantau dan menanggapi peristiwa operasional di aplikasi Anda. AWS Pemberitahuan akhir dukungan: Pada 20 Februari 2026, AWS akan mengakhiri dukungan untuk layanan Amazon Chime. Setelah 20 Februari 2026, Anda tidak akan lagi dapat mengakses konsol Amazon Chime atau sumber daya aplikasi Amazon Chime. Untuk informasi lebih lanjut, kunjungi posting blog. Ini tidak memengaruhi ketersediaan layanan Amazon Chime SDK.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS

- AWS CodeBuildadalah layanan build yang dikelola sepenuhnya yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon Simple Email Service (Amazon SES) membantu Anda mengirim dan menerima pesan email dengan menggunakan alamat email dan domain Anda sendiri.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.

Alat lainnya

- Slack, penawaran Salesforce, adalah platform percakapan bertenaga AI yang menyediakan kolaborasi obrolan dan video, mengotomatiskan proses tanpa kode, dan mendukung berbagi informasi.
- SonarQubeadalah alat analisis lokal yang dirancang untuk mendeteksi masalah pengkodean di lebih dari 30 bahasa, kerangka kerja, dan platform IAc.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub chatops-slack.

Praktik terbaik

- CloudFormation manajemen tumpukan - Jika Anda mengalami kegagalan selama eksekusi CloudFormation tumpukan, kami sarankan Anda menghapus tumpukan yang gagal. Kemudian, buat ulang dengan nilai parameter yang benar. Pendekatan ini mendukung penerapan yang bersih dan membantu menghindari potensi konflik atau implementasi sebagian.
- Konfigurasi email kotak masuk bersama — Saat Anda mengonfigurasi `SharedInboxEmail` parameter, gunakan daftar distribusi umum yang dapat diakses oleh semua pengembang yang relevan. Pendekatan ini mempromosikan transparansi dan membantu pemberitahuan penting menjangkau anggota tim yang relevan.
- Alur kerja persetujuan produksi — Untuk lingkungan produksi, batasi akses ke saluran Slack yang digunakan untuk persetujuan build. Hanya pemberi persetujuan yang ditunjuk yang harus menjadi anggota saluran ini. Praktik ini mempertahankan rantai tanggung jawab yang jelas dan meningkatkan keamanan dengan membatasi siapa yang dapat menyetujui perubahan kritis.
- Izin IAM - Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Lakukan pengaturan awal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori chatops-slack untuk pola ini, gunakan perintah berikut. `git clone "git@github.com:aws-samples/chatops-slack.git"`	AWS DevOps, Membangun pemimpin, DevOps insinyur, administrator Cloud
Buat file.zip yang berisi kode Lambda.	Buat file.zip untuk kode AWS Lambda fungsi untuk `CheckBuildStatus` dan `ApprovalEmail` fungsionalitas. Untuk membuat	AWS DevOps, Membangun pemimpin, DevOps insinyur, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`notification.zip` `danapproval.zip` , gunakan perintah berikut.
	`cd chatops-slack/src`
	`chmod -R 775 *`
	`zip -r approval.zip approval`
	`zip -r notification.zip notification`	

Terapkan file tumpukan pre-requisite.yml.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan file `pre-requisite.yml` tumpukan.	File `pre-requisite.yml` CloudFormation stack menyebarkan sumber daya awal yang diperlukan sebelum Anda mengeksekusi file `app-security.yml` stack. Untuk menjalankan `pre-requisite.yml` file, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan buka CloudFormationkonsol. Pilih Buat tumpukan, lalu pilih Dengan sumber daya	Administrator AWS, AWS DevOps, Pemimpin build, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	baru (standar) dari daftar dropdown. 2. Pada halaman Buat tumpukan, pilih Pilih templat yang ada dan Unggah file templat. Kemudian, pilih Pilih file dan pilih pre-requi site.yml. Pilih Berikutnya. 3. Pada halaman Tentukan detail tumpukan, masukkan nilai untuk parameter seperti yang dijelaskan dalam Informasi tambahan. Lalu, pilih Selanjutnya. 4. Pada halaman Configure stack options, pilih peran IAM untuk membuat resource seperti yang dijelaskan dalam Prasyarat. Lalu, pilih Selanjutnya. 5. Pada halaman Tinjau dan buat, pilih Kirim. 6. Pada halaman detail untuk tumpukan, pilih tab Sumber Daya dan Output. Catat nilai untuk`S3Lambda`,`CKMSKeyAr n` , dan `CKMSKeyId` parameter yang digunakan dalam langkah-langkah berikut.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file.zip ke bucket Amazon S3.	Unggah `approval.zip` file `notification.zip` dan file yang Anda buat sebelumnya ke bucket Amazon S3 bernama. `S3LambdaBucket` File `app-security.yml` CloudFormation stack digunakan `S3LambdaBucket` untuk menyediakan fungsi Lambda.	AWS DevOps, Build lead, DevOps engineer, administrator sistem AWS

Jalankan file tumpukan app-security.yl

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan file `app-security.yml` tumpukan.	File `app-security.yml` tumpukan menyebarkan infrastruktur yang tersisa untuk notifikasi dan sistem persetujuan. Untuk menjalankan `app-security.yml` file, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan buka CloudFormationkonsol. Pilih Buat tumpukan, lalu pilih Dengan sumber daya baru (standar) dari daftar dropdown. 2. Pada halaman Buat tumpukan, pilih Pilih templat yang ada dan Unggah file templat. Kemudian, pilih	AWS DevOps, administrator sistem AWS, DevOps insinyur, Membangun prospek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pilih file dan pilih app-security.yml. Pilih Berikutnya. 3. Pada halaman Tentukan detail tumpukan, masukkan nilai untuk parameter seperti yang dijelaskan dalam Informasi tambahan. Lalu, pilih Selanjutnya. 4. Pada halaman Configure stack options, pilih peran IAM untuk membuat resource seperti yang dijelaskan dalam Prasyarat. Lalu, pilih Selanjutnya. 5. Pada halaman Tinjau dan buat, pilih Kirim.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji pengaturan notifikasi.	Untuk menguji pengaturan notifikasi, lakukan hal berikut: 1. Buka konsol Amazon SNS. Di panel navigasi kiri, pilih Topics (Topik). 2. Pilih nama topik yang diakhiri dengan LambdaToAWSSlackChatbot. 3. Pada halaman detail topik, pilih Publikasikan pesan. 4. Pada halaman Publikasikan pesan ke topik, agar badan Pesan dikirim ke titik akhir, masukkan yang berikut ini: `{ "version": "1.0", "source": "custom", "content": { "description": ":warning: This is a test notification" } }` 5. Pilih Publish message (Publikasikan Pesan). Setelah pesan pengujian berhasil dikirim, Anda akan melihat pemberitahuan di saluran Slack. Untuk informasi selengkapnya, lihat Menguji	AWS DevOps, administrator sistem AWS, DevOps insinyur, Membangun prospek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	notifikasi dari Layanan AWS ke Slack di Pengemban g Amazon Q di Panduan Administrator aplikasi obrolan.	

Mengatur alur persetujuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan tindakan Lambda khusus.	Untuk mengatur AWS Lambda tindakan kustom, lakukan hal berikut: 1. Pilih tombol elipsis vertikal di bagian bawah notifikasi pengiriman di saluran Slack Anda. 2. Di Kelola tindakan, pilih Buat. 3. Masukkan nama tindakan kustom, misalnya, Menyetujui. Nama ini adalah pengenal unik untuk tindakan kustom Anda. 4. Masukkan nama untuk tombol tindakan kustom Anda, misalnya, Menyetuju i. Nama ini ditampilkan pada tombol pada notifikas i Anda. Nama ini harus 20 karakter atau kurang dan dapat menggabungkan emoji.	Administrator AWS, AWS DevOps, Pemimpin build, DevOps insinyur, Admin Slack

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Untuk jenis tindakan kustom, pilih Tindakan Lambda. 6. Pilih Berikutnya. 7. Pilih Akun AWS dan di Wilayah AWS mana Anda ingin menerapkan tindakan ini. 8. Pilih Muat Lambdas. 9. Dalam Define Lambda Function, pilih fungsi Lambda yang diakhiri dengan. ApprovalE mailLambdaLalu, pilih Selanjutnya. 10. Untuk membuat tombol Setujui, pilih Simpan di halaman Kriteria tampilan.	
Validasi aliran persetujuan.	Untuk memvalidasi bahwa alur persetujuan berfungsi seperti yang diharapkan, pilih tombol Setujui di Slack. Slackbot harus mengirim pemberitahuan di utas pesan dengan string konfirmasi Email Persetujuan berhasil dikirim.	Administrator AWS, AWS DevOps, DevOps insinyur, Admin Slack

Pemecahan masalah

Isu	Solusi
Kesalahan konfigurasi Slack	Untuk informasi tentang masalah pemecahan masalah yang terkait dengan kesalahan konfigurasi Slack, lihat Memecahkan Masalah Pengembang Amazon Q di Panduan Administrator Pengembang Amazon Q di aplikasi obrolan.
Pemindaian gagal karena beberapa alasan lain	Kesalahan ini berarti bahwa tugas pembuatan kode telah gagal. Untuk memecahkan masalah, buka tautan yang ada di pesan. Kegagalan tugas pembuatan kode mungkin memiliki kemungkinan penyebab berikut: • Aplikasi tidak dikemas dengan benar. `sonar-scanner` Perintah tidak dapat menemukan `sonar.project.env.properties` file. • Nilai untuk`SonarFileName` ,`SonarFileDirectory` , atau `SonarToken` parameter tidak benar. Periksa nilainya, lalu jalankan file tumpukan lagi. • Tuan rumah Sonar tidak dapat dijangkau. • Masalah lain yang dapat Anda gunakan log untuk memecahkan masalah.

Sumber daya terkait

AWS dokumentasi

- Konfigurasikan klien Slack
- Membuat tindakan kustom
- Membuat prosedur identitas alamat email

- Tutorial: Memulai dengan Slack

Sumber daya lainnya

- Tambahkan aplikasi ke ruang kerja Slack Anda (dokumentasi Slack)
- Menghasilkan dan menggunakan token (SonarQube dokumentasi)
- Pengantar instalasi server (SonarQube dokumentasi)

Informasi tambahan

Solusi ini menekankan Amazon Q Developer dalam tindakan kustom aplikasi obrolan untuk tujuan manajemen rilis. Namun, Anda dapat menggunakan kembali solusi dengan memodifikasi kode Lambda untuk kasus penggunaan spesifik Anda dan membangun di atasnya.

Parameter file CloudFormation tumpukan

Tabel berikut menunjukkan parameter dan deskripsi mereka untuk file CloudFormation `pre-requisite.yml` stack.

Kunci	Deskripsi
`StackName`	Nama CloudFormation tumpukan.
`S3LambdaBucket`	Nama bucket Amazon S3 tempat Anda mengunggah kode Lambda. Nama harus unik secara global.
`SonarToken`	Token SonarQube pengguna seperti yang dijelaskan dalam Prasyarat.

Tabel berikut menunjukkan parameter dan deskripsi mereka untuk file CloudFormation `app-security.yml` stack.

Kunci	Deskripsi

`CKMSKeyArn`	Nama Sumber Daya AWS KMS key Amazon (ARN) yang digunakan dalam peran IAM dan fungsi Lambda yang dibuat di tumpukan ini.
`CKMSKeyId`	AWS KMS key ID yang digunakan dalam topik Amazon SNS dibuat di tumpukan ini.
`EnvironmentType`	Nama lingkungan klien untuk penyebara n pipeline pemindaian aplikasi. Pilih nama lingkungan dari daftar dropdown nilai yang diizinkan.
`S3LambdaBucket`	Nama bucket Amazon S3 yang berisi file `approval.zip` dan `notification.zip` file.
`SESEmail`	Nama identitas email terdaftar di Amazon SES seperti yang dijelaskan dalam Prasyarat. Identitas ini adalah alamat email sumber.
`SharedInboxMail`	Alamat email tujuan tempat pemberitahuan pemindaian dikirim.
`SlackChannelId`	ID saluran saluran Slack tempat Anda ingin notifikasi dikirim. Untuk menemukan ID saluran, klik kanan nama saluran di Detail Saluran di aplikasi Slack. ID saluran ada di bagian bawah.
`SlackWorkspaceId`	ID ruang kerja Slack seperti yang dijelaska n dalam Prasyarat. Untuk menemukan ID ruang kerja Slack, masuk ke, buka Amazon Q Developer di konsol aplikasi obrolan Konsol Manajemen AWS, dan pilih Klien yang dikonfigurasi, Slack, WorkspaceID.
`StackName`	Nama CloudFormation tumpukan.
`SonarFileDirectory`	Direktori yang berisi `sonar.project.<env >.properties` file.

`SonarFileName`	Nama `sonar.project.<env>properties` file.
`SourceCodeZip`	Nama file.zip yang berisi `sonar.project.<env>properties` file dan kode sumber.

Terapkan sistem agen di Amazon Bedrock dengan framework CreWai dengan menggunakan Terraform

Vanitha Dontireddy, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menerapkan sistem AI multi-agen yang dapat diskalakan dengan menggunakan kerangka kerja CreWai yang terintegrasi dengan Amazon Bedrock dan Terraform. Solusi ini memungkinkan organisasi untuk membuat, menyebarkan, dan mengelola alur kerja agen AI yang canggih melalui infrastruktur sebagai kode (IAc). Dalam pola ini, kemampuan orkestrasi multi-agen CreWai digabungkan dengan model fondasi Amazon Bedrock dan otomatisasi infrastruktur Terraform. Hasilnya, tim dapat membangun sistem AI siap produksi yang menangani tugas-tugas kompleks dengan pengawasan manusia yang minimal. Pola ini menerapkan keamanan tingkat perusahaan, skalabilitas, dan praktik terbaik operasional.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan izin yang sesuai untuk mengakses model dasar Amazon Bedrock
- Terraform versi 1.5 atau yang lebih baru diinstal
- Python versi 3.9 atau yang lebih baru diinstal
- Kerangka kerja CreWai diinstal

Batasan

- Interaksi agen dibatasi oleh jendela konteks model.
- Pertimbangan manajemen status Terraform untuk penerapan skala besar berlaku untuk pola ini.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Dalam pola ini, interaksi berikut terjadi:

- Amazon Bedrock menyediakan dasar untuk intelijen agen melalui rangkaian model fondasinya (FMs). Ini memungkinkan pemrosesan bahasa alami (NLP), penalaran, dan kemampuan pengambilan keputusan untuk agen AI sambil mempertahankan ketersediaan dan skalabilitas yang tinggi.
- Kerangka CreWai berfungsi sebagai lapisan orkestrasi inti untuk membuat dan mengelola agen AI. Ini menangani protokol komunikasi agen, delegasi tugas, dan manajemen alur kerja sambil berintegrasi dengan Amazon Bedrock.
- Terraform mengelola seluruh tumpukan infrastruktur melalui kode, termasuk sumber daya komputasi, jaringan, grup keamanan, dan peran AWS Identity and Access Management (IAM). Ini memastikan penerapan yang konsisten dan dikendalikan versi di seluruh lingkungan. Penerapan Terraform membuat yang berikut:
  - AWS Lambda fungsi untuk menjalankan aplikasi CreWai
  - Bucket Amazon Simple Storage Service (Amazon S3) untuk kode dan laporan
  - Peran IAM dengan izin yang sesuai
  - CloudWatch Pencatatan Amazon
  - Eksekusi terjadwal oleh Amazon EventBridge

Diagram berikut mengilustrasikan arsitektur untuk menerapkan sistem multi-agen CreWai dengan menggunakan Amazon Bedrock dan Terraform.

Diagram menunjukkan alur kerja berikut:

1. Pengguna mengkloning repositori.
2. Pengguna menjalankan perintah `terraform apply` untuk menyebarkan AWS sumber daya.
3. Konfigurasi model Amazon Bedrock mencakup penentuan model pondasi (FM) yang akan digunakan untuk mengonfigurasi agen CrewAI.
4. EventBridge Aturan dibuat untuk memicu fungsi Lambda sesuai dengan jadwal yang ditentukan.
5. Saat dipicu (baik berdasarkan jadwal atau manual), fungsi Lambda menginisialisasi dan mengasumsikan peran IAM dengan izin untuk mengakses dan Amazon Bedrock. Layanan AWS
6. Kerangka kerja CreWai memuat konfigurasi agen dari file YAMAL dan membuat agen AI khusus (kru audit keamanan AWS infrastruktur). Fungsi Lambda secara berurutan mengeksekusi agen ini untuk memindai AWS sumber daya, menganalisis kerentanan keamanan, dan menghasilkan laporan audit yang komprehensif.

7. CloudWatch Log menangkap informasi eksekusi terperinci dari fungsi Lambda dengan periode retensi 365 hari AWS Key Management Service dan enkripsi AWS KMS() untuk persyaratan kepatuhan. Log memberikan visibilitas ke aktivitas agen, pelacakan kesalahan, dan metrik kinerja, memungkinkan pemantauan dan pemecahan masalah yang efektif dari proses audit keamanan.
8. Laporan audit keamanan dibuat dan disimpan secara otomatis di bucket Amazon S3 yang ditentukan. Pengaturan otomatis membantu menjaga pemantauan keamanan yang konsisten dengan overhead operasional minimal.

Setelah penerapan awal, alur kerja menyediakan audit dan pelaporan keamanan berkelanjutan untuk AWS infrastruktur Anda tanpa intervensi manual.

Ikhtisar agen AI

Pola ini menciptakan beberapa agen AI, masing-masing dengan peran, sasaran, dan alat unik:

- Agen analis keamanan mengumpulkan dan menganalisis informasi AWS sumber daya.
- Agen penguji penetrasi mengidentifikasi kerentanan dalam sumber daya. AWS
- Agen pakar kepatuhan memeriksa konfigurasi terhadap standar kepatuhan.
- Agen penulis laporan mengumpulkan temuan ke dalam laporan komprehensif.

Agen-agen ini berkolaborasi dalam serangkaian tugas, memanfaatkan keterampilan kolektif mereka untuk melakukan audit keamanan dan menghasilkan laporan yang komprehensif. (`config/agents.yaml`File tersebut menguraikan kemampuan dan konfigurasi masing-masing agen dalam kru ini.)

Pemrosesan analisis keamanan terdiri dari tindakan berikut:

1. Agen analis keamanan memeriksa data yang dikumpulkan tentang AWS sumber daya seperti berikut:
   - Instans dan grup keamanan Amazon Elastic Compute Cloud (Amazon EC2)
   - Bucket dan konfigurasi Amazon S3
   - Peran, kebijakan, dan izin IAM
   - Konfigurasi virtual private cloud (VPC) dan pengaturan jaringan
   - Basis data Amazon RDS dan pengaturan keamanan
   - Fungsi dan konfigurasi Lambda
   - Lainnya Layanan AWS dalam lingkup audit

2. Agen penguji penetrasi mengidentifikasi potensi kerentanan.
3. Para agen berkolaborasi melalui kerangka CreWai untuk berbagi temuan.

Pembuatan laporan terdiri dari tindakan berikut:

1. Agen penulis laporan mengumpulkan temuan dari semua agen lainnya.
2. Masalah keamanan diatur oleh layanan, tingkat keparahan, dan dampak kepatuhan.
3. Rekomendasi remediasi dihasilkan untuk setiap masalah yang diidentifikasi.
4. Laporan audit keamanan komprehensif dibuat dalam format penurunan harga dan diunggah ke bucket Amazon S3 yang ditentukan. Laporan historis disimpan untuk pelacakan kepatuhan dan peningkatan postur keamanan.

Kegiatan penebangan dan pemantauan meliputi:

- CloudWatch log menangkap detail eksekusi dan kesalahan apa pun.
- Metrik eksekusi Lambda dicatat untuk pemantauan.

> **Note**
>
> Kode untuk `aws-security-auditor-crew` bersumber dari repositori GitHub 3P-Agentic_Frameworks, tersedia dalam koleksi Sampel. AWS

Ketersediaan dan skala

Anda dapat memperluas agen yang tersedia ke lebih dari empat agen inti. Untuk menskalakan dengan agen khusus tambahan, pertimbangkan jenis agen baru berikut:

- Agen spesialis intelijen ancaman dapat melakukan hal berikut:
  - Memantau umpan ancaman eksternal dan berkorelasi dengan temuan internal
  - Memberikan konteks tentang ancaman yang muncul yang relevan dengan infrastruktur Anda
  - Memprioritaskan kerentanan berdasarkan eksploitasi aktif di alam liar
- Agen kerangka kepatuhan dapat fokus pada bidang peraturan tertentu seperti berikut:
  - Agen kepatuhan Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS)

- Agen kepatuhan Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan 1996 (HIPAA)
- Agen kepatuhan Sistem dan Kontrol Organisasi 2 (SOC 2)
- Agen kepatuhan Peraturan Perlindungan Data Umum (GDPR)

Dengan memperluas agen yang tersedia secara serius, solusi ini dapat memberikan wawasan keamanan yang lebih dalam dan lebih khusus sambil mempertahankan skalabilitas di seluruh lingkungan yang luas. AWS Untuk informasi selengkapnya tentang pendekatan implementasi, pengembangan alat, dan pertimbangan penskalaan, lihat Informasi tambahan.

Alat

Layanan AWS

- Amazon Bedrock adalah layanan AI yang dikelola sepenuhnya yang membuat model foundation berkinerja tinggi (FMs) tersedia untuk digunakan melalui API terpadu.
- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS Dalam pola ini, ini digunakan untuk menjadwalkan dan mengatur alur kerja agen.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Dalam pola ini, ia menyediakan penyimpanan objek untuk artefak agen dan manajemen negara.

Alat-alat lainnya

- CrewAI adalah kerangka kerja berbasis Python open source untuk membangun sistem AI multi-agen.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di GitHub deploy-crewai-agents-terraformrepositori.

Praktik terbaik

- Terapkan manajemen status yang tepat untuk Terraform dengan menggunakan backend Amazon S3 dengan penguncian Amazon DynamoDB. Untuk informasi selengkapnya, lihat Praktik terbaik Backend dalam Praktik terbaik untuk menggunakan Penyedia AWS Terraform.
- Gunakan ruang kerja untuk memisahkan lingkungan pengembangan, pementasan, dan produksi.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.
- Aktifkan pencatatan dan pemantauan terperinci melalui CloudWatch Log.
- Menerapkan mekanisme coba lagi dan penanganan kesalahan untuk operasi agen.

Epik

Menyebarkan kerangka CreWai

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori.	Untuk mengkloning repositori pola ini di mesin lokal Anda, jalankan perintah berikut:
	`git clone "git@github.com:aws-samples/deploy-crewai-agents-terraform.git"`
	`cd deploy-crewai-agents-terraform`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Edit variabel lingkungan.	Untuk mengedit variabel lingkungan, lakukan hal berikut: 1. Buat `terraform.tfvars` file dari contoh `terraform.tfvars.example` di direktori terraform. 2. Edit variabel lingkungan untuk menggunakan informasi Anda sendiri.	DevOps insinyur

Buat infrastruktur.	Untuk membuat infrastruktur, jalankan perintah berikut:
	`cd terraform`
	`terraform init`
	`terraform plan`
	Tinjau rencana eksekusi dengan cermat. Jika perubahan yang direncanakan dapat diterima, maka jalankan perintah berikut:
	`terraform apply --auto-approve`
		DevOps insinyur

Akses agen CreWai

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses agen.	Agen di Audit Keamanan AWS Infrastruktur dan kru Pelaporan dikerahkan sebagai fungsi Lambda. Untuk mengakses agen, gunakan langkah-langkah berikut: 1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/. 2. Pada halaman Functions, temukan dan kemudian pilih fungsi bernama`{project_name} \` function (seperti yang didefinisikan dalam variabel Terraform Anda). 3. Dari halaman fungsi, Anda dapat mengambil tindakan berikut: • Lihat detail konfigurasi. • Pantau metrik eksekusi. • Lihat CloudWatch log. • Uji fungsi secara manual.	DevOps insinyur
(Opsional) Konfigurasikan eksekusi manual agen.	Agen dikonfigurasi untuk berjalan secara otomatis pada jadwal harian (tengah malam UTC). Namun, Anda	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dapat memicunya secara manual dengan menggunakan langkah-langkah berikut: 1. Di konsol Lambda, pilih fungsi bernama. `{project_name}` 2. Pada halaman fungsi, pilih tab Uji. 3. Buat acara uji baru dengan objek `{}` JSON kosong. 4. Untuk menjalankan acara, pilih Uji. Untuk detail selengkapnya, lihat Menguji fungsi Lambda di konsol dalam dokumentasi Lambda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses log agen untuk debugging.	Agen CreWai berjalan di lingkungan Lambda dengan izin yang diperlukan untuk melakukan audit keamanan dan menyimpan laporan di Amazon S3. Outputnya adalah laporan penurunan harga yang menyediakan analisis keamanan komprehensif AWS infrastruktur Anda. Untuk membantu proses debug detail perilaku agen, lakukan hal berikut: 1. Di Konsol Manajemen AWS, navigasikan ke CloudWatch Log. 2. Temukan grup log untuk fungsi Lambda Anda. 3. Cari entri log dengan nama agen (misalnya, spesialis pemetaan infrastruktur dan analis keamanan eksploras i). 4. Tinjau log untuk wawasan tentang tindakan masing-masing agen.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat hasil eksekusi agen.	Untuk melihat hasil eksekusi agen, lakukan hal berikut: 1. Di Konsol Manajemen AWS, navigasikan ke Amazon S3. 2. Buka bucket Amazon S3 bernama `{project_name}-reports-{random_suffix}` (seperti yang ditentukan dalam variabel Terraform Anda). Laporan disimpan dengan nama file berbasis stempel waktu sebagai berikut: `security-audit-report-YYYY-MM-DD-HH-MM-SS.md`)	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pantau eksekusi agen.	Untuk memantau eksekusi agen melalui CloudWatch log, lakukan hal berikut: 1. Di Konsol Manajemen AWS, navigasikan ke CloudWatch. 2. Pergi ke Grup log. 3. Pilih grup log bernama `/aws/lambda/{project_name}-function` . 4. Di aliran Log, pilih aliran log terbaru untuk melihat informasi eksekusi terperinci.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Sesuaikan perilaku agen.	Untuk memodifikasi agen atau tugas mereka, lakukan hal berikut:
	1. Perbarui file konfigurasi di repositori lokal Anda:
	• File berikut mendefini sikan peran, kemampuan , dan pengaturan masing-masing agen: `src/aws_infrastructure_security_audit_and_reporting/config/agents.yaml`
	• Dile berikut mendefini sikan tugas dan alur kerja untuk agen:
	`src/aws_infrastructure_security_audit_and_reporting/config/tasks.yaml`
	2. Untuk mengemas ulang dan memperbarui fungsi Lambda, gunakan perintah berikut:
	`cd terraform`
	`terraform apply`
		DevOps insinyur

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hapus sumber daya yang dibuat.	Untuk menghapus semua infrastruktur yang dibuat oleh pola ini, jalankan perintah berikut:
	`terraform plan -destroy`
	Warning
	Perintah berikut akan secara permanen menghapus semua sumber daya yang dibuat oleh pola ini. Perintah akan meminta konfirmasi sebelum menghapus sumber daya apa pun.
	Tinjau rencana penghancuran dengan hati-hati. Jika penghapusan yang direncanakan dapat diterima, maka jalankan perintah berikut:
	`terraform destroy`
		DevOps insinyur

Pemecahan masalah

Isu	Solusi
Perilaku agen	Untuk informasi tentang masalah ini, lihat Menguji dan memecahkan masalah perilaku agen dalam dokumentasi Amazon Bedrock.
Masalah jaringan Lambda	Untuk informasi tentang masalah ini, lihat Memecahkan masalah jaringan di Lambda dalam dokumentasi Lambda.
Izin IAM	Untuk informasi tentang masalah ini, lihat Memecahkan Masalah IAM di dokumentasi IAM.

Sumber daya terkait

AWS Blog

- Bangun sistem agen dengan CreWai dan Amazon Bedrock

AWS dokumentasi

- Dokumentasi Amazon Bedrock
- Bagaimana Agen Amazon Bedrock bekerja
- AWS Kerangka Well-Architected

Sumber daya lainnya

- Dokumentasi CreWai
- Dokumentasi Penyedia Terraform AWS

Informasi tambahan

Bagian ini berisi informasi tentang pendekatan implementasi, pengembangan alat, dan pertimbangan penskalaan yang terkait dengan diskusi sebelumnya dalam Otomasi dan skala.

Pendekatan implementasi

Pertimbangkan pendekatan berikut untuk menambahkan agen:

1. Konfigurasi agen:
   - Tambahkan definisi agen baru ke `config/agents.yaml` file.
   - Tentukan latar belakang, tujuan, dan alat khusus untuk setiap agen.
   - Konfigurasikan kemampuan memori dan analisis berdasarkan spesialisasi agen.
2. Orkestrasi tugas:
   - Perbarui `config/tasks.yaml` file untuk menyertakan tugas khusus agen baru.
   - Buat dependensi antar tugas untuk membantu memastikan arus informasi yang tepat.
   - Menerapkan eksekusi tugas paralel jika sesuai.

Implementasi teknis

Berikut ini adalah tambahan pada `agents.yaml` file untuk agen Spesialis Intelijen Ancaman yang diusulkan:

```
Example new agent configuration in agents.yaml
threat_intelligence_agent:
 name: "Threat Intelligence Specialist"
 role: "Cybersecurity Threat Intelligence Analyst"
 goal: "Correlate AWS security findings with external threat intelligence"
 backstory: "Expert in threat intelligence with experience in identifying emerging
 threats and attack patterns relevant to cloud infrastructure."
verbose: true
allow_delegation: true
tools:
- "ThreatIntelligenceTool"
- "AWSResourceAnalyzer"
```

Pengembangan alat

Dengan kerangka kerja CreWai, Anda dapat mengambil tindakan berikut untuk meningkatkan efektivitas kru audit keamanan Anda:

- Buat alat khusus untuk agen baru.
- Integrasikan dengan eksternal APIs untuk intelijen ancaman.
- Mengembangkan analisa khusus untuk yang berbeda Layanan AWS.

Pertimbangan penskalaan

Saat memperluas sistem Audit dan Pelaporan Keamanan AWS Infrastruktur Anda untuk menangani lingkungan yang lebih besar atau audit yang lebih komprehensif, atasi faktor penskalaan berikut:

- Sumber daya komputasi
  - Meningkatkan alokasi memori Lambda untuk menangani agen tambahan.
  - Pertimbangkan untuk membagi beban kerja agen di beberapa fungsi Lambda.
- Manajemen biaya
  - Pantau penggunaan Amazon Bedrock API saat jumlah agen meningkat.
  - Menerapkan aktivasi agen selektif berdasarkan lingkup audit.
- Efisiensi kolaborasi
  - Optimalkan berbagi informasi antar agen.
  - Menerapkan struktur agen hierarkis untuk lingkungan yang kompleks.
- Peningkatan basis pengetahuan
  - Menyediakan agen dengan basis pengetahuan khusus untuk domain mereka.
  - Perbarui pengetahuan agen secara teratur dengan praktik terbaik keamanan baru.

Menerapkan pekerjaan AWS Glue dengan pipeline AWS CodePipeline CI/CD

Bruno Klein dan Luis Henrique Massao Yamada, Amazon Web Services

Ringkasan

Pola ini menunjukkan bagaimana Anda dapat mengintegrasikan AWS CodeCommit dan AWS CodePipeline dengan AWS Glue, dan menggunakan AWS Lambda untuk meluncurkan pekerjaan segera setelah pengembang mendorong perubahan mereka ke repositori AWS jarak jauh. CodeCommit

Saat pengembang mengirimkan perubahan ke repositori ekstrak, transformasi, dan muat (ETL) dan mendorong perubahan ke AWS CodeCommit, pipeline baru akan dipanggil. Pipeline memulai fungsi Lambda yang meluncurkan pekerjaan AWS Glue dengan perubahan ini. Pekerjaan AWS Glue melakukan tugas ETL.

Solusi ini sangat membantu dalam situasi di mana bisnis, pengembang, dan insinyur data ingin meluncurkan pekerjaan segera setelah perubahan dilakukan dan didorong ke repositori target. Ini membantu mencapai tingkat otomatisasi dan reproduktifitas yang lebih tinggi, sehingga menghindari kesalahan selama peluncuran pekerjaan dan siklus hidup.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Git diinstal pada mesin lokal
- Amazon Cloud Development Kit (Amazon CDK) diinstal pada mesin lokal
- Python diinstal pada mesin lokal
- Kode di bagian Lampiran

Batasan

- Pipeline selesai segera setelah pekerjaan AWS Glue berhasil diluncurkan. Itu tidak menunggu kesimpulan dari pekerjaan.
- Kode yang disediakan dalam lampiran dimaksudkan untuk tujuan demo saja.

Arsitektur

Tumpukan teknologi target

- AWS Glue
- AWS Lambda
- AWS CodePipeline
- AWS CodeCommit

Arsitektur target

Prosesnya terdiri dari langkah-langkah ini:

1. Pengembang atau insinyur data membuat modifikasi dalam kode ETL, melakukan, dan mendorong perubahan ke AWS. CodeCommit
2. Dorongan memulai pipa.
3. Pipeline memulai fungsi Lambda, yang `codecommit:GetFile` memanggil repositori dan mengunggah file ke Amazon Simple Storage Service (Amazon S3).
4. Fungsi Lambda meluncurkan pekerjaan AWS Glue baru dengan kode ETL.
5. Fungsi Lambda menyelesaikan pipeline.

Otomatisasi dan skala

Lampiran sampel menunjukkan bagaimana Anda dapat mengintegrasikan AWS Glue dengan AWS CodePipeline. Ini memberikan contoh dasar yang dapat Anda sesuaikan atau perluas untuk penggunaan Anda sendiri. Untuk detailnya, lihat bagian Epik.

Alat

- AWS CodePipeline — AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dikelola sepenuhnya yang membantu Anda mengotomatiskan saluran pipa rilis untuk pembaruan aplikasi dan infrastruktur yang cepat dan andal.
- AWS CodeCommit — AWS CodeCommit adalah layanan kontrol sumber terkelola penuh yang menampung repositori berbasis Git yang aman.

- AWS Lambda — AWS Lambda adalah layanan komputasi tanpa server yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server.
- AWS Glue — AWS Glue adalah layanan integrasi data tanpa server yang memudahkan untuk menemukan, menyiapkan, dan menggabungkan data untuk analitik, pembelajaran mesin, dan pengembangan aplikasi.
- Klien Git — Git menyediakan alat GUI, atau Anda dapat menggunakan baris perintah atau alat desktop untuk memeriksa artefak yang diperlukan. GitHub
- AWS CDK — AWS CDK adalah kerangka kerja pengembangan perangkat lunak open source yang membantu Anda menentukan sumber daya aplikasi cloud Anda dengan menggunakan bahasa pemrograman yang sudah dikenal.

Epik

Menyebarkan kode sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengonfigurasi AWS CLI.	Konfigurasikan AWS Command Line Interface (AWS CLI) untuk menargetkan dan mengautentikasi dengan akun AWS Anda saat ini. Untuk petunjuk, lihat dokumentasi AWS CLI.	Pengembang, DevOps insinyur
Ekstrak file proyek sampel.	Ekstrak file dari lampiran untuk membuat folder yang berisi file proyek sampel.	Pengembang, DevOps insinyur

Menyebarkan kode sampel.	Setelah Anda mengekstrak file, jalankan perintah berikut dari lokasi ekstrak untuk membuat contoh dasar:
	`cdk bootstrap`
	`cdk deploy`
	`git init`
		Pengembang, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`git remote add origin <code-commit-repository-url>`
	`git stage .`
	`git commit -m "adds sample code"`
	`git push --set-upstream origin main`
	
	Setelah perintah terakhir, Anda dapat memantau status pipeline dan pekerjaan AWS Glue.	

Sesuaikan kode.	Sesuaikan kode untuk file etl.py sesuai dengan kebutuhan bisnis Anda. Anda dapat merevisi kode ETL, memodifikasi tahapan pipeline, atau memperluas solusi.	Insinyur data

Sumber daya terkait

- Memulai dengan AWS CDK
- Menambahkan pekerjaan di AWS Glue
- Integrasi tindakan sumber di CodePipeline
- Memanggil fungsi AWS Lambda dalam pipeline di CodePipeline
- Pemrograman AWS Glue
- AWS CodeCommit GetFile API

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menerapkan kode di beberapa Wilayah AWS menggunakan AWS CodePipeline, AWS CodeCommit, dan AWS CodeBuild

Anand Krishna Varanasi, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara membangun infrastruktur atau arsitektur di beberapa Wilayah Amazon Web Services (AWS) dengan menggunakan AWS CloudFormation. Ini mencakup integrasi berkelanjutan (CI) /penerapan berkelanjutan (CD) di beberapa Wilayah AWS untuk penerapan yang lebih cepat. Langkah-langkah dalam pola ini telah diuji untuk pembuatan CodePipeline pekerjaan AWS untuk diterapkan ke tiga Wilayah AWS sebagai contoh. Anda dapat mengubah jumlah Wilayah berdasarkan kasus penggunaan Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- • CodeBuild Peran dengan AmazonS3 FullAccess dan kebijakan. CloudWatchFullAccess Kebijakan ini memberikan CodeBuild akses untuk menonton peristiwa AWS CodeCommit melalui Amazon CloudWatch dan menggunakan Amazon Simple Storage Service (Amazon S3) sebagai penyimpanan artefak.
  - CloudFormation Peran AWS dengan kebijakan berikut, yang memberikan AWS CloudFormation, pada tahap Build akhir, kemampuan untuk membuat atau memperbarui fungsi AWS Lambda, mendorong atau menonton CloudWatch log Amazon, dan untuk membuat dan memperbarui set perubahan.
    - AWSLambdaFullAccess
    - AWSCodeDeployFullAccess
    - CloudWatchFullAccess
    - AWSCloudFormationFullAccess
    - AWSCodePipelineFullAccess

> **Note**
>
> Dua peran AWS Identity and Access Management (IAM) untuk AWS CodeBuild dan AWS CloudFormation dengan kebijakan yang tepat CodeBuild untuk melakukan tugas

CI yaitu pengujian, bundling, pengemasan artefak, dan penerapan ke beberapa Wilayah AWS secara paralel. Periksa silang kebijakan yang dibuat oleh CodePipeline untuk memverifikasi bahwa CodeBuild dan AWS CloudFormation memiliki izin yang tepat dalam fase CI dan CD.

Arsitektur

Arsitektur dan alur kerja Multiple-region pola ini terdiri dari langkah-langkah berikut.

1. Anda mengirim kode Anda ke CodeCommit repositori.
2. Setelah menerima pembaruan kode atau komit apa pun, CodeCommit memanggil CloudWatch acara, yang pada gilirannya memulai CodePipeline pekerjaan.
3. CodePipeline melibatkan CI yang ditangani oleh. CodeBuild Tugas-tugas berikut dilakukan.
   - Pengujian CloudFormation template AWS (opsional)
   - Pengemasan CloudFormation template AWS untuk setiap Wilayah yang disertakan dalam penerapan. Misalnya, pola ini diterapkan secara paralel ke tiga Wilayah AWS, jadi CodeBuild mengemas CloudFormation template AWS ke dalam tiga bucket S3, satu di setiap Wilayah yang ditentukan. Bucket S3 digunakan oleh CodeBuild sebagai repositori artefak saja.
4. CodeBuild mengemas artefak sebagai input untuk fase Deploy berikutnya, yang berjalan secara paralel di tiga Wilayah AWS. Jika Anda menentukan jumlah Wilayah yang berbeda, CodePipeline akan menyebarkan ke Wilayah tersebut.

Alat

Alat

- AWS CodePipeline — CodePipeline adalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak Anda secara terus menerus.
- AWS CodeBuild — CodeBuild adalah layanan build terkelola penuh yang mengompilasi kode sumber Anda, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit — CodeCommit adalah layanan kontrol versi yang dihosting oleh Amazon Web Services yang dapat Anda gunakan untuk menyimpan dan mengelola aset secara pribadi (seperti kode sumber dan file biner) di cloud.

- AWS CloudFormation — AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan menyiapkan sumber daya Amazon Web Services sehingga Anda dapat menghabiskan lebih sedikit waktu untuk mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi yang berjalan di AWS.
- AWS Identity and Access Management — AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke sumber daya AWS dengan aman.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet. Ini dirancang untuk membuat komputasi skala web lebih mudah bagi pengembang.

Kode

Contoh kode berikut adalah untuk `BuildSpec.yaml` file (Build phase).

```
---
artifacts:
discard-paths: true
files:
- packaged-first-region.yaml
- packaged-second-region.yaml
- packaged-third-region.yaml
phases:
build:
commands:
- echo "********BUILD PHASE - CF PACKAGING**********"
- "aws cloudformation package --template-file sam-template.yaml --s3-bucket
 $S3_FIRST_REGION --output-template-file packaged-first-region.yaml --region
 $FIRST_REGION"
- "aws cloudformation package --template-file sam-template.yaml --s3-bucket
 $S3_SECOND_REGION --output-template-file packaged-second-region.yaml --region
 $SECOND_REGION"
- "aws cloudformation package --template-file sam-template-anand.yaml --s3-bucket
 $S3_THIRD_REGION --output-template-file packaged-third-region.yaml --region
 $THIRD_REGION"
install:
commands:
- echo "********BUILD PHASE - PYTHON SETUP**********"
runtime-versions:
python: 3.8
post_build:
commands:
```

```
- echo "********BUILD PHASE - PACKAGING COMPLETION**********"
pre_build:
commands:
- echo "********BUILD PHASE - DEPENDENCY SETUP**********"
- "npm install --silent --no-progress"
- echo "********BUILD PHASE - DEPENDENCY SETUP DONE**********"
version: 0.2
```

Epik

Siapkan kode dan CodeCommit repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih Wilayah AWS utama untuk penerapan.	Masuk ke akun AWS Anda dan pilih Wilayah utama untuk penerapan. CodeCommit Repositori akan berada di Wilayah utama.	DevOps
Buat CodeCommit repositori.	Buat CodeCommit repositor i, dan tekan kode yang diperlukan ke dalamnya. Kode umumnya mencakup template AWS CloudFormation atau AWS SAM, kode Lambda jika ada, dan CodeBuild `buildspec.yaml` file sebagai input ke AWS. CodePipeline	DevOps
Dorong kode ke dalam CodeCommit repositori.	Di bagian Lampiran, unduh kode untuk contoh ini, lalu dorong kode yang diperlukan ke dalamnya. Secara umum, kode dapat menyertakan templat AWS CloudForm ation atau AWS SAM, kode Lambda, dan CodeBuild	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`buildspec.yaml` file sebagai input ke pipeline.	

Fase sumber: Buat pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodePipeline pekerjaan.	Di CodePipeline konsol, pilih Buat pipeline.	DevOps
Beri nama CodePipeline pekerjaan dan pilih setelan peran layanan.	Masukkan nama untuk pekerjaan, dan pertahank an setelan peran layanan default sehingga CodePipeline menciptakan peran dengan kebijakan yang diperlukan terlampir.	DevOps
Tentukan lokasi untuk toko artefak.	Di bawah Pengaturan lanjutan, pertahankan opsi default sehingga CodePipeline membuat bucket S3 untuk digunakan untuk penyimpan an artefak kode. Jika Anda menggunakan bucket S3 yang ada, bucket harus berada di Region utama yang Anda tentukan di epik pertama.	DevOps
Tentukan kunci enkripsi.	Simpan opsi default, AWS Managed Key Default, atau pilih untuk menggunakan kunci terkelola pelanggan AWS Key Management	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Service (AWS KMS) Anda sendiri.	
Tentukan penyedia sumber.	Di bawah penyedia Sumber, pilih AWS CodeCommit.	DevOps
Tentukan repositori.	Pilih CodeCommit repositor i yang Anda buat di epik pertama. Jika Anda menempatkan kode di cabang, pilih cabang.	DevOps
Tentukan bagaimana perubahan kode terdeteksi.	Pertahankan default, Amazon CloudWatch Events, sebagai pemicu perubahan CodeCommit untuk memulai CodePipeline pekerjaan.	DevOps

Fase build: Konfigurasikan pipeline

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan penyedia build.	Untuk penyedia build, pilih AWS CodeBuild.	DevOps
Tentukan Wilayah AWS.	Pilih Wilayah utama, yang Anda tentukan dalam epik pertama.	DevOps

Fase build: Buat dan konfigurasikan proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat proyek	Pilih Buat proyek, dan masukkan nama untuk proyek tersebut.	DevOps
Tentukan gambar lingkungan.	Untuk demonstrasi pola ini, gunakan gambar CodeBuild terkelola default. Anda juga memiliki opsi untuk menggunakan gambar Docker khusus jika Anda memilikinya.	DevOps
Tentukan sistem operasi.	Pilih Amazon Linux 2 atau Ubuntu. **Note** Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkap nya, lihat Amazon Linux 2 FAQs.	DevOps
Tentukan peran layanan.	Pilih peran yang Anda buat CodeBuild sebelum Anda mulai membuat CodePipel ine pekerjaan. (Lihat bagian Prasyarat.)	DevOps
Tetapkan opsi tambahan.	Untuk Timeout dan Timeout Antrian, pertahankan nilai default. Untuk sertifikat, pertahankan pengaturan default kecuali Anda memiliki	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sertifikat khusus yang ingin Anda gunakan.	
Buat variabel lingkungan.	Untuk setiap Wilayah AWS yang ingin Anda terapkan, buat variabel lingkungan dengan memberikan nama bucket S3 dan nama Region (misalnya, us-east-1).	DevOps
Berikan nama file buildspec, jika bukan buildspec.yml.	Biarkan bidang ini kosong jika nama file adalah default,`buildspec.yaml` . Jika Anda mengganti nama file buildspec, masukkan nama di sini. Pastikan itu cocok dengan nama file yang ada di CodeCommit repositori.	DevOps
Tentukan logging.	Untuk melihat log untuk CloudWatch Acara Amazon, pertahankan pengaturan default. Atau Anda dapat menentukan nama grup atau logger tertentu.	DevOps

Lewati fase Deploy

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lewati fase penyebaran dan selesaikan pembuatan pipa.	Ketika Anda mengatur pipeline, CodePipeline memungkinkan Anda untuk membuat hanya satu tahap dalam fase Deploy. Untuk	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menerapkan ke beberapa Wilayah AWS, lewati fase ini. Setelah pipeline dibuat, Anda dapat menambahkan beberapa tahap fase Deploy.	

Fase penerapan: Konfigurasikan pipeline untuk penerapan ke Wilayah pertama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan tahapan ke fase Deploy.	Edit pipeline dan pilih Add stage di fase Deploy. Tahap pertama ini adalah untuk Wilayah primer.	DevOps
Berikan nama tindakan untuk panggung.	Masukkan nama unik yang mencerminkan tahap pertama (primer) dan Wilayah. <region>Misalnya, masukkan primery_ _deploy.	DevOps
Tentukan penyedia tindakan.	Untuk penyedia Action, pilih AWS CloudFormation.	DevOps
Konfigurasikan Wilayah untuk tahap pertama.	Pilih Wilayah (primer) pertama, Wilayah yang sama di mana CodePipeline dan CodeBuild diatur. Ini adalah Wilayah utama tempat Anda ingin menyebarkan tumpukan.	DevOps
Tentukan artefak input.	Pilih BuildArtifact. Ini adalah output dari fase build.	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan tindakan yang akan diambil.	Untuk mode Tindakan, pilih Buat atau perbarui tumpukan.	DevOps
Masukkan nama untuk CloudFormation tumpukan.		DevOps
Tentukan template untuk Wilayah pertama.	Pilih nama paket khusus Wilayah yang dikemas oleh CodeBuild dan dibuang ke bucket S3 untuk Region (primer) pertama.	DevOps
Tentukan kemampuannya.	Kemampuan diperlukan jika template tumpukan menyertakan sumber daya IAM atau jika Anda membuat tumpukan langsung dari template yang berisi makro. Untuk pola ini, gunakan CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_AUTO_EXPAND.	DevOps

Fase penyebaran: Konfigurasikan pipeline untuk penerapan ke Wilayah kedua

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan tahap kedua ke fase Deploy.	Untuk menambahkan tahap untuk Wilayah kedua, edit pipeline dan pilih Tambah tahap di fase Deploy. Penting: Proses pembuatan Wilayah kedua sama dengan Wilayah	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pertama, kecuali untuk nilai-nil ai berikut.	
Berikan nama tindakan untuk tahap kedua.	Masukkan nama unik yang mencerminkan tahap kedua dan Wilayah kedua.	DevOps
Konfigurasikan Wilayah untuk tahap kedua.	Pilih Wilayah kedua tempat Anda ingin menyebarkan tumpukan.	DevOps
Tentukan template untuk Wilayah kedua.	Pilih nama paket khusus Wilayah yang dikemas oleh CodeBuild dan dibuang ke bucket S3 untuk Wilayah kedua.	DevOps

Fase penyebaran: Konfigurasikan pipeline untuk penerapan ke Wilayah ketiga

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan tahap ketiga ke fase Deploy.	Untuk menambahkan tahap untuk Wilayah ketiga, edit pipeline dan pilih Tambah tahap di fase Deploy. Penting: Proses pembuatan Wilayah kedua sama dengan dua Wilayah sebelumnya, kecuali untuk nilai-nilai berikut.	DevOps
Berikan nama tindakan untuk tahap ketiga.	Masukkan nama unik yang mencerminkan tahap ketiga dan Wilayah ketiga.	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Wilayah untuk tahap ketiga.	Pilih Wilayah ketiga tempat Anda ingin menyebarkan tumpukan.	DevOps
Tentukan template untuk Wilayah ketiga.	Pilih nama paket khusus Wilayah yang dikemas oleh CodeBuild dan dibuang ke bucket S3 untuk Wilayah ketiga.	DevOps

Bersihkan penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya AWS.	Untuk membersihkan penyebaran, hapus CloudFormation tumpukan di setiap Wilayah. Kemudian hapus CodeCommit, CodeBuild, dan CodePipel ine sumber daya dari Wilayah utama.	DevOps

Sumber daya terkait

- Apa itu AWS CodePipeline?
- Model Aplikasi AWS Tanpa Server
- AWS CloudFormation
- Referensi struktur CloudFormation arsitektur AWS untuk AWS CodePipeline

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Terapkan beban kerja dari DevOps pipeline Azure ke kluster Amazon EKS pribadi

Mahendra Revanasiddappa, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menerapkan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) dari DevOps pipeline Azure ke cluster Amazon Elastic Kubernetes Service (Amazon EKS) pribadi. Ini mengatasi tantangan kritis yang dihadapi oleh organisasi yang meningkatkan postur keamanan mereka dengan beralih ke titik akhir server API pribadi untuk kluster Amazon EKS mereka.

Titik akhir publik mengekspos server API Kubernetes langsung ke internet, menciptakan permukaan serangan yang lebih besar yang berpotensi ditargetkan oleh aktor jahat. Dengan beralih ke titik akhir pribadi, akses ke bidang kontrol cluster dibatasi ke dalam virtual private cloud (VPC) pelanggan.

Meskipun transisi cluster Amazon EKS ke titik akhir API pribadi secara signifikan meningkatkan keamanan, ini memperkenalkan tantangan konektivitas untuk platform eksternal seperti Azure. CI/CD DevOps Titik akhir pribadi hanya dapat diakses dari dalam VPC cluster atau jaringan peered. Oleh karena itu, DevOps agen Azure standar yang dihosting Microsoft, yang beroperasi di luar jaringan AWS pribadi, tidak dapat menjangkau server API Kubernetes secara langsung. Ini merusak alur kerja penerapan tipikal yang mengandalkan alat seperti kubectl atau Helm yang berjalan pada agen ini karena mereka gagal membuat koneksi ke cluster.

Untuk mengatasi masalah ini, pola ini menampilkan pendekatan yang efisien dengan menggunakan DevOps agen Azure yang dihosting sendiri dalam kluster Amazon EKS pribadi. Solusi ini menawarkan optimalisasi biaya yang unggul, efisiensi operasional, dan skalabilitas sambil menjaga persyaratan keamanan. Pendekatan ini terutama menguntungkan perusahaan yang ingin merampingkan DevOps proses multi-cloud mereka tanpa mengorbankan kinerja atau keamanan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) versi 2.13.17 atau yang lebih baru, diinstal.
- kubectl versi 1.25.1 atau yang lebih baru, diinstal.
- Cluster Amazon EKS pribadi versi 1.24 atau yang lebih baru dibuat, dengan izin untuk membuat ruang nama, rahasia, dan penerapan.

- Node pekerja di cluster Amazon EKS dengan konektivitas keluar ke internet sehingga DevOps agen Azure yang menjalankannya dapat terhubung ke kumpulan agen Azure DevOps .
- GitHub akun dibuat.
- DevOps Proyek Azure dengan akses untuk mengonfigurasi koneksi layanan, yang merupakan koneksi terautentikasi antara Azure Pipelines dan layanan eksternal atau jarak jauh, dibuat.
- AWS Toolkit for Azure DevOps Versi 1.15 atau yang lebih baru diinstal untuk DevOps proyek Azure yang dijelaskan pada poin sebelumnya. Untuk petunjuk penginstalan, lihat AWS Toolkit for Azure DevOpsdi Visual Studio Marketplace.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Pola ini menciptakan yang berikut:

- Repositori Amazon ECR - Repositori Amazon Elastic Container Registry (Amazon ECR) menyimpan image Docker dengan agen Azure dan aplikasi sampel yang DevOps digunakan.
- Azure DevOps agent pool - Kumpulan agen yang DevOps dihosting sendiri Azure mendaftarkan agen yang berjalan di cluster Amazon EKS pribadi.
- Peran IAM - Peran AWS Identity and Access Management (IAM) untuk koneksi layanan Azure untuk menyediakan akses yang diperlukan ke agen yang berjalan di klaster Amazon EKS pribadi.
- Koneksi DevOps layanan Azure - Koneksi layanan di DevOps akun Azure untuk menggunakan peran IAM yang menyediakan akses yang diperlukan untuk pekerjaan pipeline untuk diakses. Layanan AWS

Diagram berikut menunjukkan arsitektur penerapan DevOps agen Azure yang dihosting sendiri pada kluster Amazon EKS pribadi dan menerapkan aplikasi sampel pada cluster yang sama.

Diagram menunjukkan alur kerja berikut:

1. Terapkan DevOps agen Azure yang dihosting sendiri sebagai penerapan di dalam klaster Amazon EKS.
2. DevOps Agen Azure terhubung ke kumpulan agen pada DevOps akun Azure menggunakan token akses pribadi (PAT) untuk otentikasi.
3. Azure Pipelines mengonfigurasi pipeline untuk diterapkan dengan menggunakan kode dari repositori. GitHub
4. Pipa berjalan pada agen dari kumpulan agen yang dikonfigurasi dalam konfigurasi pipa. DevOps Agen Azure mendapatkan informasi pekerjaan dari pipa dengan terus-menerus melakukan polling ke akun DevOps Azure.
5. DevOps Agen Azure membuat image Docker sebagai bagian dari pekerjaan pipeline dan mendorong gambar ke repositori Amazon ECR.
6. DevOps Agen Azure menyebarkan aplikasi sampel pada cluster Amazon EKS pribadi di namespace yang disebut. `webapp`

Alat

Alat

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- kubectl adalah antarmuka baris perintah yang membantu Anda menjalankan perintah terhadap klaster Kubernetes.

Repositori kode

- Kode untuk pola ini tersedia di repositori GitHub deploy-kubernetes-resources-to- amazon-eks-using-azure -devops.

Praktik terbaik

- Untuk Amazon EKS, lihat Panduan Praktik Terbaik Amazon EKS.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Buat koneksi layanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Temukan GUID DevOps organisasi Azure.	Masuk ke DevOps akun Azure Anda, lalu gunakan URL berikut untuk menemukan GUID organisasi: `https://dev.azure.com/{DevOps_Org_ID}/_apis/projectCollections?api-version=6.0` Di URL, ganti `{DevOps_org_ID}` dengan ID organisasi Azure DevOps Anda.	AWS DevOps
Konfigurasikan iDP di file. Akun AWS	Untuk mengonfigurasi penyedia Identitas (iDP) di Akun AWS sambungan layanan Azure, gunakan langkah-langkah berikut: 1. Masuk ke Konsol Manajemen AWS, dan buka konsol IAM di https://	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	console.aws.amazon.com/iam/. 2. Di panel kiri, pilih Penyedia identitas. 3. Pilih Tambahkan Penyedia. 4. Pilih OpenID Connect sebagai tipe Provider. 5. Untuk URL Penyedia, masukkan URL DevOps penerbit Azure. Setiap penyewa Azure DevOps memiliki keunikan `OrganizationGUID` yang biasanya menggunakan format berikut: `https://vstoken.dev.azure.com/{OrganizationGUID}` Ganti `{OrganizationGUID}` dengan ID organisasi Azure DevOps Anda. 6. Untuk Pemirsa, masukkan api: //Azure Exchange ADToken. Ini adalah nilai tetap untuk Azure DevOps. 7. Pilih Tambahkan Penyedia. 8. Catat ARN dari penyedia yang baru dibuat untuk Anda gunakan di tugas berikutnya. Untuk detail selengkapnya, lihat Cara federasi AWS dari	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Azure menggunakan DevOps OpenID Connect.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM di. Akun AWS	Untuk membuat kebijakan IAM guna memberikan izin yang diperlukan ke peran IAM yang digunakan oleh DevOps pipeline Azure, gunakan langkah-langkah berikut: 1. Di konsol IAM, di panel kiri, pilih Kebijakan. 2. Pilih Buat Kebijakan. 3. Untuk Tentukan Izin, di Editor kebijakan, pilih JSON. Ganti kebijakan JSON default dengan JSON berikut: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "Statement1", "Effect": "Allow", "Action": [ "ecr:*", "eks:DescribeCluster", "eks:ListClusters" ], "Resource": "*"`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre> }] }</pre> 4. Pilih Berikutnya. 5. Untuk nama Kebijakan , masukkan nama untuk kebijakan IAM. Pola ini menggunakan nama ADO-Policy. 6. Pilih Buat kebijakan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM di. Akun AWS	Untuk mengonfigurasi peran IAM dalam Akun AWS koneksi layanan Azure, gunakan langkah-langkah berikut: 1. Di konsol IAM, di panel kiri, pilih Peran. 2. Pilih Buat peran. 3. Untuk jenis entitas Tepercaya, pilih Identitas Web. 4. Pilih iDP yang benar dari daftar dropdown. Nama iDP dimulai dengan. `vstoken.dev.azure.com/{OrganizationGUID}` 5. Dalam daftar tarik-turun Audiens, pilih api: ADToken //Azure Exchange. 6. Untuk membatasi peran ini hanya pada satu koneksi layanan, tambahkan kondisi. Di bawah Kondisi, pilih Tambahkan kondisi, dan untuk Kunci, pilih vstoken.dev.azure.com/{OrganizationGuid} :sub. Dalam Kondisi, pilih StringEquals. Untuk Nilai, gunakan format berikut:`sc://{OrganizationName}/{Pr`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ojectName}/{ServiceConnectionName} . UntukServiceConnectionName , gunakan aws-sc. Anda akan membuat koneksi layanan ini di tugas berikutnya. 7. Pilih Berikutnya. 8. Untuk Menambahkan izin, pilih Kebijakan ADO, yang merupakan kebijakan yang Anda buat di tugas sebelumnya. 9. Pilih Berikutnya, dan untuk nama Peran, masukkan ado-role. Untuk Entitas Tepercaya tertentu, gunakan kebijakan kepercayaan berikut: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Federated": "arn:aws:iam::{account_id}:oidc-provider/vstoken.dev.azure.com/{OrganizationGUID}" },`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"Action": "sts:AssumeRoleWithWebIdentity", "Condition": { "StringEquals": { "vstoken.dev.azure.com/{OrganizationGUID}:aud": "api://AzureADTokenExchange", "vstoken.dev.azure.com/{OrganizationGUID}:sub": "sc://{OrganizationName}/{ProjectName}/{ServiceConnectionName}" } } }] }</pre> Dalam kebijakan, berikan informasi Anda untuk placeholder berikut: • `{account_id}` - Akun AWS ID • `{OrganizationGUID}` - GUID DevOps Organisasi Azure • `{OrganizationName}` - Nama DevOps Organisasi Azure • `{ProjectName}` - Nama DevOps proyek Azure	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `{ServiceConnection Name}` - Nama koneksi DevOps layanan Azure. Gunakan aws-sc. Anda akan membuat koneksi layanan ini di tugas berikutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat koneksi layanan di DevOps akun Azure.	Untuk mengonfigurasi koneksi layanan Azure, gunakan langkah-langkah berikut: 1. Di DevOps proyek Azure Anda, pilih Pengaturan proyek, Koneksi layanan. 2. Pilih Koneksi layanan baru, pilih jenis koneksi layanan sebagai`aws`, lalu pilih Berikutnya. 3. Agar Peran Diasumsikan, masukkan arn untuk peran IAM ado-role. Anda membuat ado-role di tugas sebelumnya Buat peran IAM di. Akun AWS 4. Pilih kotak centang Gunakan OIDC. 5. Untuk nama koneksi Layanan, masukkan aws-sc di properti tugas. 6. Pilih Simpan. Untuk detail selengkapnya, lihat Membuat sambungan layanan di dokumentasi Microsoft.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan peran IAM ke file konfigurasi Amazon EKS.	Peran IAM harus memiliki izin yang diperlukan untuk melakukan operasi yang diperlukan di cluster Amazon EKS. Karena ini adalah peran pipeline, peran IAM harus dapat mengelola hampir semua jenis sumber daya di cluster. Oleh karena itu, izin `system:masters` grup sesuai untuk peran ini.
	
	Untuk menambahkan konfigurasi yang diperlukan ke `aws-auth ConfigMap` dalam Kubernetes, gunakan kode berikut:
	
	`- groups:`
	`  - system:masters`
	`  rolearn: arn:aws:iam::{account_id}:role/ADO-role`
	`  username: ADO-role`
	
	Ganti `{account_id}` dengan Akun AWS ID Anda.
	
	Untuk informasi selengkap nya, lihat Cara Amazon EKS bekerja dengan IAM di dokumentasi Amazon EKS.
		AWS DevOps

Buat kolam agen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kumpulan agen yang dihosting sendiri.	Untuk mengonfigurasi kumpulan agen yang dihosting sendiri di DevOps akun Azure, gunakan langkah-langkah berikut: 1. Masuk ke organisasi DevOps akun Azure Anda. 2. Pilih DevOps Organisasi Azure. 3. Pilih DevOps proyek Azure Anda. 4. Pilih Pengaturan proyek. 5. Pilih kolam Agen. 6. Pilih Tambahkan kolam. 7. Pilih Self-Hosted. 8. Untuk Nama, masukkan eks-agent. 9. Pilih kotak centang Berikan izin akses ke semua saluran pipa. 10. Pilih Buat. Untuk detail selengkapnya, lihat Membuat dan mengelola kumpulan agen di dokumentasi Microsoft.	

Buat gambar DevOps agen Azure dan dorong ke Amazon ECR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR.	Gambar Docker yang digunakan untuk menyebarkan DevOps agen Azure dan aplikasi sampel (`webapp`) pada kluster Amazon EKS pribadi harus disimpan dalam repositori Amazon ECR. Untuk membuat repositori Amazon ECR, gunakan langkah-langkah berikut: 1. Buka konsol Amazon ECR di https://console.aws.amazon.com/ecr/repositori. 2. Dari bilah navigasi, pilih Wilayah AWSuntuk membuat repositori Anda. 3. Pada halaman Repositori, pilih Repositori pribadi, lalu pilih Buat repositori. 4. Untuk nama Repositori, masukkan webapp. Agar aplikasi sampel dalam pola ini berfungsi, nama repositori Amazon ECR harus menggunakan webapp. Jika Anda menggunakan nama lain untuk repositori, lihat Pemecahan masalah.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk detail selengkapnya, lihat Membuat repositori pribadi Amazon ECR untuk menyimpan gambar dalam dokumentasi Amazon ECR.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat Dockerfile untuk membangun agen DevOps Azure.	Buat Dockerfile untuk membangun image Docker yang telah menginstal agen DevOps Azure. Simpan konten berikut dalam file bernama`Dockerfile` :
	
	`FROM ubuntu:22.04`
	`ENV TARGETARCH="linux-x64"`
	`RUN apt update && apt upgrade -y && apt install -y curl git jq libicu70 unzip wget`
	
	`RUN curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"`
	`RUN unzip awscliv2.zip`
	`RUN ./aws/install`
	`RUN rm -rf aws awscliv2.zip`
	
	`RUN curl -sSL https://get.docker.com/ | sh`
	
	`RUN curl -sL https://aka.ms/InstallAzureCLIDeb | bash`
	`RUN mkdir -p azp`
	`WORKDIR /azp/`
	
	`COPY ./start.sh ./`
	`RUN chmod +x ./start.sh`
	
	`RUN useradd -m -d /home/agent agent`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`RUN chown -R agent:agent /azp /home/agent`
	`RUN groupadd -f docker`
	`RUN usermod -aG docker agent`
	`USER agent`
	
	`ENTRYPOINT [ "./start.sh" ]`	

Buat skrip untuk DevOps agen Azure.	Untuk membuat `start.sh` skrip, gunakan langkah-l angkah berikut: 1. Pergi ke prosedur Buat dan bangun Dockerfile di dokumentasi Microsoft, dan gulir ke langkah 5. Simpan konten berikut**`~/azp-agent-in-docker/start.sh`** , pastikan untuk menggunakan akhiran baris Unix-style (LF). 2. Salin konten skrip, dan simpan dalam file bernama `start.sh` di direktori yang sama dengan Dockerfile.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat gambar Docker dengan agen Azure DevOps .	Untuk membuat image Docker untuk menginstal DevOps agen Azure, gunakan Dockerfile yang Anda buat sebelumnya untuk membangun image. Di direktori yang sama tempat Dockerfile disimpan, jalankan perintah berikut:
	`aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com`
	`docker build --platform linux/amd64 -t ado-agent:latest .`
	`docker tag ado-agent:latest aws_account_id.dkr.ecr.region.amazonaws.com/webapp:latest`
	`docker push aws_account_id.dkr.ecr.region.amazonaws.com/webapp:latest`
	Ganti `aws_account_id` dan `region` dengan Akun AWS ID Anda dan Wilayah AWS.
		AWS DevOps

Terapkan DevOps agen Azure ke cluster Amazon EKS pribadi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan token akses pribadi Azure.	Agen yang berjalan di kluster Amazon EKS pribadi memerlukan token akses pribadi (PAT) sehingga dapat mengautentikasi dengan akun Azure DevOps . Untuk menghasilkan PAT, gunakan langkah-langkah berikut: 1. Masuk dengan akun pengguna yang ingin Anda gunakan di DevOps organisasi Azure (`https://dev.azure.com/{Your_Organization}`). 1. Dari halaman beranda, buka pengaturan pengguna Anda, lalu pilih Token akses pribadi. 2. Pilih Token Baru. 3. Masukkan Nama untuk token. 4. Pilih Tampilkan semua cakupan. 5. Untuk Agen Pools, pilih kotak centang Baca & kelola. 6. Pilih Buat.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	7. Untuk membuat rahasia di kluster Amazon EKS pribadi, gunakan konfigurasi berikut:
	
	`apiVersion: v1`
	`kind: Secret`
	`metadata:`
	`  name: azdevops-pat`
	`  namespace: default`
	`type: Opaque`
	`stringData:`
	`  AZP_TOKEN: <PAT Token>`
	
	1. Simpan konfigurasi dalam file bernama `ado-secret.yaml` . Ganti `<PAT Token>` dengan token akses pribadi yang baru saja Anda buat. Untuk membuat rahasia, jalankan perintah berikut:
	
	`kubectl create -f ado-secret.yaml`
	
	Untuk detail selengkapnya, lihat Mendaftarkan agen menggunakan token akses pribadi (PAT) di dokumentasi Microsoft.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Gunakan file manifes Kubernetes untuk penerapan agen.	Untuk menerapkan DevOps agen Azure di kluster Amazon EKS pribadi, salin file manifes berikut dan simpan file sebagai: `agent-deployment.yaml`
	
	`apiVersion: apps/v1`
	`kind: Deployment`
	`metadata:`
	`  name: azure-pipelines-agent-eks`
	`  labels:`
	`  app: azure-pipelines-agent`
	`spec:`
	`  replicas: 1`
	`  selector:`
	`  matchLabels:`
	`  app: azure-pipelines-agent`
	`  template:`
	`  metadata:`
	`  labels:`
	`  app: azure-pipelines-agent`
	`  spec:`
	`  containers:`
	`  - name: docker`
	`  image: docker:dind`
	`  securityContext:`
	`  privileged: true`
	`  volumeMounts:`
	`  - name: shared-workspace`
	`  mountPath: /workspace`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>- name: dind-
	storage
	mountPath: /
	var/lib/docker
	env:
	- name:
	DOCKER_TLS_CERTDIR
	value: ""
	- name: azure-pip
	elines-agent
	image: aws_accou
	nt_id.dkr.ecr.regi
	on.amazonaws.com/w
	ebapp:latest
	env:
	- name: AZP_URL
	value: "<Azure
	account URL>"
	- name: AZP_POOL
	value: "eks-
	agent"
	- name:
	AZP_TOKEN
	valueFrom:
	secretKey
	Ref:
	name:
	azdevops-pat
	key:
	AZP_TOKEN
	- name:
	AZP_AGENT_NAME
	valueFrom:
	fieldRef:
	fieldPath
	: metadata.name
	- name:
	DOCKER_HOST
	value: tcp://
	localhost:2375
	volumeMounts:</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>- mountPath: /workspace
	name: shared-workspace
	volumes:
	- name: dind-storage
	emptyDir: {}
	- name: shared-workspace
	emptyDir: {}</pre>
	Ganti `aws_account_id` dan `<Azure account URL>` dengan Akun AWS ID dan URL DevOps akun Azure Anda.	

Terapkan agen di cluster Amazon EKS pribadi.	Untuk menerapkan agen Azure Devops di klaster Amazon EKS pribadi, gunakan perintah berikut: <pre>kubectl create -f agent-deployment.tf</pre>	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi agen sedang berjalan.	Untuk memverifikasi bahwa DevOps agen Azure sedang berjalan, gunakan perintah berikut:
	`kubectl get deploy azure-pipelines-agent-eks`
	Output yang diharapkan harus serupa dengan yang berikut:
	`NAME READY UP-TO-DATE AVAILABLE AGE azure-pipelines-agent-eks 1/1 1 1 58s`
	Pastikan bahwa `READY` kolom menunjukkan`1/1`.
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi agen terdaftar di kolam DevOps agen Azure.	Untuk memverifikasi bahwa agen dikerahkan di klaster Amazon EKS pribadi dan terdaftar di kumpulan age`neks-agent` , gunakan langkah-langkah berikut: 1. Masuk ke DevOps organisasi Azure Anda (`https://dev.azure.com/{Your_Organization}`). 2. Pilih Pengaturan proyek. 3. Pilih kolam Agen. 4. Pilih kumpulan eks-agent, lalu periksa tab Agen. Anda harus melihat satu agen terdaftar dengan Status Online, dan nama agen harus dimulai dengan azure-pip elines-agent-eks-*.	AWS DevOps

Menyebarkan aplikasi sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Garpu repositori aplikasi sampel ke akun Anda GitHub .	Lakukan fork repositori AWS Sampel berikut ke akun Anda GitHub : https://github.com/aws-samples/deploy-kubernetes-resou	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	rces-to- amazon-eks-using-a zure -devops	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pipa.	Untuk membuat pipeline di DevOps akun Azure Anda, gunakan langkah-langkah berikut: 1. Masuk dengan akun pengguna yang ingin Anda gunakan di DevOps organisasi Azure (`https://dev.azure.com/{Your_Organization}`). 2. Arahkan ke konsol proyek dan pipeline Anda. 3. Pilih Pipeline Baru. 4. Untuk Di mana kode Anda, pilih GitHub. 5. Berikan kredensyal yang diperlukan agar pipeline dapat terhubung ke akun Anda GitHub 6. Pilih repositori deploy-kubernetes-resources-to-amazon-eks-using-azure -devops. 7. Untuk Mengonfigurasi pipeline Anda, pilih file YAML Azure Pipelines yang ada. 8. Untuk Pilih file YAMM yang ada, pilih main untuk Branch dan azure_pipelines.yaml untuk path.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	9. Pilih Lanjutkan.
	10. Untuk Meninjau pipa YAMM Anda, ganti nilai parameter input untuk `awsRegion` dan `awsEKSClusterName` dengan informasi Anda:
	
	`pool:`
	`  name: eks-agent`
	`#pool: self-hosted # If you are running self-hosted Azure DevOps Agents`
	
	`stages:`
	`# Refering the pipeline template, input parameter that are not specified will be added with defaults`
	`- template: ./pipeline_templates/main_template.yaml`
	`  parameters:`
	`  serviceConnectionName: aws-sc`
	`  awsRegion: <your region>`
	`  awsEKSClusterName: <name of your EKS cluster>`
	`  projectName: webapp`
	
	1. Pilih RUN.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa aplikasi sampel digunakan.	Setelah pipeline selesai, verifikasi keberhasilan penerapan aplikasi sampel dengan memeriksa repositori Amazon ECR dan kluster Amazon EKS. Untuk memverifikasi artefak di repositori Amazon ECR, gunakan langkah-langkah berikut: 1. Arahkan ke `webapp` repositori Amazon ECR. 2. Konfirmasikan keberadaan artefak baru berikut: • Gambar Docker — `<date>.<build_number>-image` • Bagan helm — `<date>.<build_number>-helm` Misalnya, `20250501.1-image` dan `20250501.1-helm`. Untuk memverifikasi penerapan pada kluster Amazon EKS pribadi di namespace`webapp`, gunakan perintah berikut:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`kubectl get deploy -n webapp`
	
	Output yang diharapkan adalah sebagai berikut:
	
	`NAME READY UP-TO-DATE AVAILABLE`
	`webapp 1/1 1 1`
	
	Catatan: Jika ini adalah proses pipeline pertama Anda, Anda mungkin perlu mengotorisasi koneksi layanan dan kumpulan agen. Cari permintaan izin di antarmuka DevOps pipeline Azure, dan setujui untuk melanjutkan.	

Pemecahan masalah

Isu	Solusi
Pipeline gagal saat nama repositori Amazon ECR tidak cocok `webapp`	Aplikasi sampel mengharapkan nama repositori Amazon ECR cocok dengan parameter di. `projectName: webapp azure_pipeline.yml` Untuk mengatasi masalah ini, ganti nama repositori Amazon ECR Anda menjadi`webapp`, atau perbarui yang berikut ini:

Isu	Solusi
	• Ganti nama `webapp` direktori di repositori bercabang agar sesuai dengan nama GitHub repositori Amazon ECR Anda. • Perbarui `projectName` parameter `azure_pipeline.yml` agar sesuai dengan nama repositori Amazon ECR Anda.
Kesalahan: Kluster Kubernetes tidak dapat dijangkau: server telah meminta klien untuk memberikan kredensyal	Jika Anda menemukan kesalahan ini di langkah “Tarik dan Terapkan Bagan Helm” di pipeline Azure Anda, akar penyebabnya biasanya berasal dari konfigurasi peran IAM yang salah di cluster Amazon EKS Anda. `aws-auth ConfigMap` Untuk mengatasi masalah ini, periksa hal berikut: • Verifikasi `aws-auth ConfigMap` konfigurasi Anda. • Periksa setelan autentikasi klaster Amazon EKS Anda: Buka konsol Amazon EKS, Detail cluster, konfigurasi Access. Pastikan mode Otentikasi diatur ke EKS API dan ConfigMap (bukan hanya EKS API).

Sumber daya terkait

AWS Blog

- Cara federasi AWS dari Azure menggunakan DevOps OpenID Connect

Layanan AWS dokumentasi

- Amazon ECR
- Amazon EKS

Dokumentasi Microsoft

- Apa itu Azure? DevOps
- Apa itu Azure Pipelines?

Jalankan kueri Amazon Redshift SQL dengan menggunakan Terraform

Sylvia Qi dan Aditya Ambati, Amazon Web Services

Ringkasan

Menggunakan infrastruktur sebagai kode (IAc) untuk penyebaran dan pengelolaan Amazon Redshift adalah praktik umum di dalamnya. DevOps IAc memfasilitasi penyebaran dan konfigurasi berbagai sumber daya Amazon Redshift, seperti cluster, snapshot, dan grup parameter. Namun, IAc tidak meluas ke pengelolaan sumber daya database seperti tabel, skema, tampilan, dan prosedur tersimpan. Elemen-elemen database ini dikelola melalui query SQL dan tidak secara langsung didukung oleh alat IAC. Meskipun ada solusi dan alat untuk mengelola sumber daya ini, Anda mungkin memilih untuk tidak memperkenalkan alat tambahan ke dalam tumpukan teknologi Anda.

Pola ini menguraikan metodologi yang menggunakan Terraform untuk menyebarkan sumber daya database Amazon Redshift, termasuk tabel, skema, tampilan, dan prosedur tersimpan. Pola membedakan antara dua jenis query SQL:

- Kueri yang tidak dapat diulang — Kueri ini dijalankan satu kali selama penerapan Amazon Redshift awal untuk membuat komponen basis data penting.
- Kueri berulang — Kueri ini tidak dapat diubah dan dapat dijalankan kembali tanpa memengaruhi database. Solusinya menggunakan Terraform untuk memantau perubahan dalam kueri yang dapat diulang dan menerapkannya sesuai dengan itu.

Untuk detail selengkapnya, lihat Panduan solusi di Informasi tambahan.

Prasyarat dan batasan

Prasyarat

Anda harus memiliki yang aktif Akun AWS dan menginstal yang berikut ini pada mesin penyebaran Anda:

- AWS Command Line Interface (AWS CLI)
- AWS CLI Profil yang dikonfigurasi dengan izin Amazon Redshift read/write
- Terraform versi 1.6.2 atau yang lebih baru
- Python3
- Boto3

Batasan

- Solusi ini mendukung satu database Amazon Redshift karena Terraform hanya memungkinkan pembuatan satu database selama pembuatan cluster.
- Pola ini tidak menyertakan pengujian untuk memvalidasi perubahan pada kueri berulang sebelum menerapkannya. Kami menyarankan Anda memasukkan tes tersebut untuk meningkatkan keandalan.
- Untuk mengilustrasikan solusinya, pola ini menyediakan `redshift.tf` file contoh yang menggunakan file status Terraform lokal. Namun, untuk lingkungan produksi, kami sangat menyarankan Anda menggunakan file status jarak jauh dengan mekanisme penguncian untuk meningkatkan stabilitas dan kolaborasi.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

Solusi ini dikembangkan dan diuji pada Amazon Redshift patch 179.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub amazon-redshift-sql-deploy-terraform.

Arsitektur

Diagram berikut mengilustrasikan cara Terraform mengelola sumber daya database Amazon Redshift dengan menangani kueri SQL yang tidak dapat diulang dan yang dapat diulang.

Diagram menunjukkan langkah-langkah berikut:

1. Terraform menerapkan kueri SQL yang tidak dapat diulang selama penerapan klaster Amazon Redshift awal.
2. Pengembang melakukan perubahan pada kueri SQL berulang.
3. Terraform memantau perubahan dalam kueri SQL yang dapat diulang.
4. Terraform menerapkan kueri SQL yang dapat diulang ke database Amazon Redshift.

Solusi yang disediakan oleh pola ini dibuat berdasarkan modul Terraform untuk Amazon Redshift. Modul Terraform menyediakan cluster dan database Amazon Redshift. Untuk menyempurnakan modul, kami menggunakan `terraform_data` sumber daya, yang memanggil skrip Python khusus untuk menjalankan kueri SQL menggunakan operasi Amazon Redshift API. ExecuteStatement Akibatnya, modul dapat melakukan hal berikut:

- Menyebarkan sejumlah sumber daya database dengan menggunakan kueri SQL setelah database disediakan.
- Pantau terus menerus untuk perubahan dalam kueri SQL yang dapat diulang dan terapkan perubahan tersebut menggunakan Terraform.

Untuk detail selengkapnya, lihat Panduan solusi di Informasi tambahan.

Alat

Layanan AWS

- Amazon Redshift adalah layanan gudang data skala petabyte yang dikelola sepenuhnya di. AWS Cloud

Alat-alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.
- Python adalah bahasa pemrograman tujuan umum yang digunakan dalam pola ini untuk mengeksekusi query SQL.

Praktik terbaik

- Praktik terbaik Amazon Redshift
- Menggunakan Amazon Redshift Data API untuk berinteraksi dengan klaster Amazon Redshift

Epik

Terapkan solusi menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositori Git yang berisi kode Terraform untuk menyediakan cluster Amazon Redshift, gunakan perintah berikut. `git clone https://github.com/aws-samples/amazon-redshift-sql-deploy-terraform.git`	DevOps insinyur

Perbarui variabel Terraform.	Untuk menyesuaikan penerapan klaster Amazon Redshift sesuai dengan kebutuhan spesifik Anda, perbarui parameter berikut dalam file. `terraform.tfvars`
	`region = "<AWS_REGION>"`
	`cluster_identifier = "<REDSHIFT_CLUSTER_IDENTIFIER>"`
	`node_type = "<REDSHIFT_NODE_TYPE>"`
	`number_of_nodes = "<REDSHIFT_NODE_COUNT>"`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`database_name = "<REDSHIFT_DB_NAME>"`
	`subnet_ids = "<REDSHIFT_SUBNET_IDS>"`
	`vpc_security_group_ids = "<REDSHIFT_SECURITY_GROUP_IDS>"`
	`run_nonrepeatable_queries = true`
	`run_repeatable_queries = true`
	`sql_path_bootstrap = "<BOOTSTRAP_SQLS_PATH>"`
	`sql_path_nonrepeatable = "<NON-REPEATABLE_SQLS_PATH>"`
	`sql_path_repeatable = "<REPEATABLE_SQLS_PATH>"`
	`sql_path_finalize = "<FINALIZE_SQLS_PATH>"`
	`create_random_password = false`
	`master_username = "<REDSHIFT_MASTER_USERNAME>"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan sumber daya menggunakan Terraform.	1. Untuk mempersiapkan proses penerapan, gunakan perintah berikut untuk menginisialisasi Terraform dalam repositori kloning.
	`terraform init`
	2. Untuk melihat pratinjau perubahan yang akan diterapkan Terraform ke infrastruktur, gunakan perintah berikut untuk membuat rencana eksekusi.
	`terraform plan -var-file terraform.tfvars`
	3. Untuk menyediakan cluster Amazon Redshift dan sumber daya terkait, gunakan perintah berikut untuk menerapkan rencana eksekusi Terraform.
	`terraform apply -var-file terraform.tfvars`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

(Opsional) Jalankan kueri SQL tambahan.	Repositori sampel menyediakan beberapa kueri SQL untuk tujuan demo. Untuk menjalankan kueri SQL Anda sendiri, tambahkan ke folder berikut:
	`/bootstrap`
	`/nonrepeatable`
	`/repeatable`
	`/finalize`	

Memantau eksekusi pernyataan SQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memantau penyebaran pernyataan SQL.	Anda dapat memantau hasil eksekusi SQL ke cluster Amazon Redshift. Untuk contoh output yang menunjukkan eksekusi SQL yang gagal dan berhasil, lihat Contoh pernyataan SQL dalam Informasi tambahan.	DBA, DevOps insinyur
Pembersihan sumber daya	Untuk menghapus semua sumber daya yang digunakan oleh Terraform, jalankan perintah berikut. `terraform destroy`	DevOps insinyur

Validasi hasilnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi data di cluster Amazon Redshift.	1. Masuk ke Konsol Manajemen AWS, dan buka konsol Amazon Redshift. 2. Pada menu navigasi, pilih Cluster. Pilih nama cluster yang relevan dalam daftar. 3. Ikuti petunjuk dalam Menanyakan database menggunakan editor kueri Amazon Redshift v2 di dokumentasi Amazon Redshift.	DBA, AWS DevOps

Sumber daya terkait

AWS dokumentasi

- Cluster yang disediakan Amazon Redshift
- Memecahkan masalah untuk Amazon Redshift Data API

Sumber daya lainnya

- Perintah: terapkan (dokumentasi Terraform)

Informasi tambahan

Panduan solusi

Untuk menggunakan solusinya, Anda harus mengatur kueri Amazon Redshift SQL Anda dengan cara tertentu. Semua kueri SQL harus disimpan dalam file dengan ekstensi. `.sql`

Dalam contoh kode yang disediakan dengan pola ini, query SQL diatur dalam struktur folder berikut. Anda dapat memodifikasi kode (`sql-queries.tfdansql-queries.py`) untuk bekerja dengan struktur apa pun yang sesuai dengan kasus penggunaan unik Anda.

```
/bootstrap
     |- Any # of files
     |- Any # of sub-folders
/nonrepeatable
     |- Any # of files
     |- Any # of sub-folders
/repeatable
     /udf
          |- Any # of files
          |- Any # of sub-folders
     /table
          |- Any # of files
          |- Any # of sub-folders
     /view
          |- Any # of files
          |- Any # of sub-folders
     /stored-procedure
          |- Any # of files
          |- Any # of sub-folders
/finalize
     |- Any # of files
     |- Any # of sub-folders
```

Mengingat struktur folder sebelumnya, selama penerapan klaster Amazon Redshift, Terraform mengeksekusi kueri dengan urutan berikut:

1. `/bootstrap`
2. `/nonrepeatable`
3. `/repeatable`
4. `/finalize`

`/repeatable`Folder berisi empat subfolder:`/udf`,, `/table`/`view`, dan/`stored-procedure`. Subfolder ini menunjukkan urutan di mana Terraform mengeksekusi kueri SQL.

Script Python yang mengeksekusi query SQL adalah. `sql-queries.py` Pertama, skrip membaca semua file dan subfolder dari direktori sumber tertentu, misalnya, `sql_path_bootstrap` parameter.

Kemudian skrip mengeksekusi kueri dengan memanggil operasi Amazon ExecuteStatementRedshift API. Anda mungkin memiliki satu atau lebih kueri SQL dalam sebuah file. Cuplikan kode berikut menunjukkan fungsi Python yang mengeksekusi pernyataan SQL yang disimpan dalam file terhadap cluster Amazon Redshift.

```
def execute_sql_statement(filename, cluster_id, db_name, secret_arn, aws_region):
    """Execute SQL statements in a file"""
    redshift_client = boto3.client(
        'redshift-data', region_name=aws_region)
    contents = get_contents_from_file(filename),
    response = redshift_client.execute_statement(
        Sql=contents[0],
        ClusterIdentifier=cluster_id,
        Database=db_name,
        WithEvent=True,
        StatementName=filename,
        SecretArn=secret_arn
    )
    ...
```

Skrip Terraform `sql-queries.tf` membuat sumber daya terraform_data yang memanggil skrip. `sql-queries.py` Ada `terraform_data` sumber daya untuk masing-masing dari empat folder:`/bootstrap`,`/nonrepeatable`,`/repeatable`, dan`/finalize`. Cuplikan kode berikut menunjukkan `terraform_data` sumber daya yang mengeksekusi query SQL dalam folder. `/bootstrap`

```
locals {
  program               = "${path.module}/sql-queries.py"
  redshift_cluster_name = try(aws_redshift_cluster.this[0].id, null)
}

resource "terraform_data" "run_bootstrap_queries" {
  count      = var.create && var.run_nonrepeatable_queries && (var.sql_path_bootstrap !
= "") && (var.snapshot_identifier == null) ? 1 : 0
  depends_on = [aws_redshift_cluster.this[0]]

  provisioner "local-exec" {
    command = "python3 ${local.program} ${var.sql_path_bootstrap}
 ${local.redshift_cluster_name} ${var.database_name} ${var.redshift_secret_arn}
 ${local.aws_region}"
  }
}
```

Anda dapat mengontrol apakah akan menjalankan query ini dengan menggunakan variabel berikut. Jika Anda tidak ingin menjalankan kueri di`sql_path_bootstrap`,,`sql_path_nonrepeatable`, atau `sql_path_repeatablesql_path_finalize`, setel nilainya. `""`

```
  run_nonrepeatable_queries = true
  run_repeatable_queries    = true
  sql_path_bootstrap        = "src/redshift/bootstrap"
  sql_path_nonrepeatable    = "src/redshift/nonrepeatable"
  sql_path_repeatable       = "src/redshift/repeatable"
  sql_path_finalize         = "src/redshift/finalize"
```

Saat Anda menjalankan`terraform apply`, Terraform mempertimbangkan `terraform_data` sumber daya yang ditambahkan setelah skrip selesai, terlepas dari hasil skripnya. Jika beberapa kueri SQL gagal, dan Anda ingin menjalankannya kembali, Anda dapat menghapus sumber daya secara manual dari status Terraform, dan menjalankannya lagi. `terraform apply` Misalnya, perintah berikut menghapus `run_bootstrap_queries` sumber daya dari status Terraform.

```
terraform state rm module.redshift.terraform_data.run_bootstrap_queries[0]
```

Contoh kode berikut menunjukkan bagaimana `run_repeatable_queries` sumber daya memantau perubahan dalam `repeatable` folder dengan menggunakan hash sha256. Jika ada file di dalam folder yang diperbarui, Terraform menandai seluruh direktori untuk pembaruan. Kemudian, Terraform menjalankan kueri di direktori lagi selama berikutnya. `terraform apply`

```
resource "terraform_data" "run_repeatable_queries" {
  count      = var.create_redshift && var.run_repeatable_queries &&
 (var.sql_path_repeatable != "") ? 1 : 0
  depends_on = [terraform_data.run_nonrepeatable_queries]

  # Continuously monitor and apply changes in the repeatable folder
  triggers_replace = {
    dir_sha256 = sha256(join("", [for f in fileset("${var.sql_path_repeatable}",
 "**") : filesha256("${var.sql_path_repeatable}/${f}")]))
  }

  provisioner "local-exec" {
    command = "python3 ${local.sql_queries} ${var.sql_path_repeatable}
 ${local.redshift_cluster_name} ${var.database_name} ${var.redshift_secret_arn}"
  }
}
```

Untuk memperbaiki kode, Anda dapat menerapkan mekanisme untuk mendeteksi dan menerapkan perubahan hanya pada file yang telah diperbarui dalam `repeatable` folder, daripada menerapkan perubahan ke semua file tanpa pandang bulu.

Contoh pernyataan SQL

Output berikut menunjukkan eksekusi SQL gagal, bersama dengan pesan kesalahan.

```
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec): Executing:
 ["/bin/sh" "-c" "python3 modules/redshift/sql-queries.py src/redshift/nonrepeatable
 testcluster-1 db1 arn:aws:secretsmanager:us-east-1:XXXXXXXXXXXX:secret:/redshift/
master_user/password-8RapGH us-east-1"]
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec):
 ----------------------------------------------------------------
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec): src/redshift/
nonrepeatable/table/admin/admin.application_family.sql
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec):
 ----------------------------------------------------------------
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec): Status:
 FAILED
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec): SQL execution
 failed.
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec): Error
 message: ERROR: syntax error at or near ")"
module.redshift.terraform_data.run_nonrepeatable_queries[0] (local-exec):   Position:
 244
module.redshift.terraform_data.run_nonrepeatable_queries[0]: Creation complete after 3s
 [id=ee50ba6c-11ae-5b64-7e2f-86fd8caa8b76]
```

Output berikut menunjukkan eksekusi SQL yang sukses.

```
module.redshift.terraform_data.run_bootstrap_queries[0]: Provisioning with 'local-
exec'...
module.redshift.terraform_data.run_bootstrap_queries[0] (local-exec): Executing:
 ["/bin/sh" "-c" "python3 modules/redshift/sql-queries.py src/redshift/bootstrap
 testcluster-1 db1 arn:aws:secretsmanager:us-east-1:XXXXXXXXXXXX:secret:/redshift/
master_user/password-8RapGH us-east-1"]
module.redshift.terraform_data.run_bootstrap_queries[0] (local-exec):
 ----------------------------------------------------------------
module.redshift.terraform_data.run_bootstrap_queries[0] (local-exec): src/redshift/
bootstrap/db.sql
module.redshift.terraform_data.run_bootstrap_queries[0] (local-exec):
 ----------------------------------------------------------------
```

```
module.redshift.terraform_data.run_bootstrap_queries[0] (local-exec): Status: FINISHED
module.redshift.terraform_data.run_bootstrap_queries[0] (local-exec): SQL execution
 successful.
module.redshift.terraform_data.run_bootstrap_queries[0]: Creation complete after 2s
 [id=d565ef6d-be86-8afd-8e90-111e5ea4a1be]
```

Ekspor laporan AWS Backup dari seluruh organisasi di AWS Organizations sebagai file CSV

Aromal Raj Jayarajan dan Purushotham G K, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara mengekspor laporan pekerjaan AWS Backup dari seluruh organisasi di AWS Organizations sebagai file CSV. Solusinya menggunakan AWS Lambda dan Amazon EventBridge untuk mengkategorikan laporan pekerjaan AWS Backup berdasarkan statusnya, yang dapat membantu saat mengonfigurasi otomatisasi berbasis status.

AWS Backup membantu organisasi mengelola dan mengotomatiskan perlindungan data secara terpusat di seluruh layanan AWS, di cloud, dan di tempat. Namun, untuk pekerjaan AWS Backup yang dikonfigurasi dalam AWS Organizations, pelaporan konsolidasi hanya tersedia di AWS Management Console dari setiap akun manajemen organisasi. Membawa pelaporan ini ke luar akun manajemen dapat mengurangi upaya yang diperlukan untuk mengaudit dan meningkatkan ruang lingkup untuk otomatisasi, pemberitahuan, dan peringatan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Organisasi aktif di AWS Organizations yang mencakup setidaknya akun manajemen dan akun anggota
- AWS Backup dikonfigurasi pada tingkat organisasi di AWS Organizations (untuk informasi selengkapnya, lihat Mengotomatiskan pencadangan terpusat pada skala di seluruh layanan AWS menggunakan AWS Backup di AWS Blog)
- Git, diinstal dan dikonfigurasi pada mesin lokal Anda

Keterbatasan

Solusi yang disediakan dalam pola ini mengidentifikasi sumber daya AWS yang dikonfigurasi hanya untuk pekerjaan AWS Backup. Laporan tidak dapat mengidentifikasi sumber daya AWS yang tidak dikonfigurasi untuk pencadangan melalui AWS Backup.

Arsitektur

Tumpukan teknologi target

- AWS Backup
- AWS CloudFormation
- Amazon EventBridge
- AWS Lambda
- AWS Security Token Service (AWS STS)
- Amazon Simple Storage Service (Amazon S3)
- AWS Identity and Access Management (IAM)

Arsitektur target

Diagram berikut menunjukkan contoh alur kerja untuk mengekspor laporan pekerjaan AWS Backup dari seluruh organisasi di AWS Organizations sebagai file CSV.

Diagram menunjukkan alur kerja berikut:

1. Aturan EventBridge acara terjadwal memanggil fungsi Lambda di akun AWS anggota (pelaporan).
2. Fungsi Lambda kemudian menggunakan AWS STS untuk mengambil peran IAM yang memiliki izin yang diperlukan untuk terhubung ke akun manajemen.
3. Fungsi Lambda kemudian melakukan hal berikut:
   - Meminta laporan pekerjaan AWS Backup yang dikonsolidasikan dari layanan AWS Backup
   - Mengkategorikan hasil berdasarkan status pekerjaan AWS Backup
   - Mengonversi respons ke file CSV
   - Mengunggah hasil ke bucket Amazon S3 di akun pelaporan dalam folder yang diberi label berdasarkan tanggal pembuatannya

Alat

Alat

- AWS Backup adalah layanan yang dikelola sepenuhnya yang membantu Anda memusatkan dan mengotomatiskan perlindungan data di seluruh layanan AWS, di cloud, dan di tempat.

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, fungsi AWS Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun AWS lainnya.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Kode

Kode untuk pola ini tersedia di GitHub aws-backup-report-generatorrepositori.

Praktik terbaik

- Praktik terbaik keamanan untuk Amazon S3 (Panduan Pengguna Amazon S3)
- Praktik terbaik untuk bekerja dengan fungsi AWS Lambda (Panduan Pengembang AWS Lambda)
- Praktik terbaik untuk akun manajemen (Panduan Pengguna AWS Organizations)

Epik

Menyebarkan komponen solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Kloning GitHub aws-backup-report-generatorrepositori dengan menjalankan perintah berikut di jendela terminal:	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/aws-samples/aws-backup-report-generator.git` Untuk informasi selengkapnya, lihat Mengkloning repositori di Dokumen. GitHub	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan komponen solusi di akun AWS anggota (pelaporan).	1. Di akun anggota (pelaporan), masuk ke AWS Management Console lalu buka CloudFormation konsol. 2. Pilih Buat tumpukan, lalu pilih Dengan sumber daya baru (standar). 3. Pada halaman Buat tumpukan, di bagian Tentukan templat, pilih Unggah file templat. 4. Pilih Pilih file. Kemudian, navigasikan ke folder root dari GitHub repositori kloning di workstation lokal Anda dan pilih template-reporting.yaml. 5. Pilih Buka, lalu pilih Berikutnya. 6. Pada halaman Tentukan detail tumpukan, untuk nama Stack, masukkan nama untuk CloudFormation tumpukan Anda. 7. Untuk ManagementAccountID, masukkan ID akun AWS untuk akun manajemen organisasi Anda di AWS Organizations. 8. Pilih Berikutnya.	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	9. Pada halaman Configure Stack Options, pilih Next. 10. Pada halaman Tinjauan, pilih kotak centang untuk mengetahui bahwa Anda meninjau konfigurasi. 11. Pilih Buat tumpukan. Tumpukan menunjukkan status CREATE_COMPLETE saat komponen solusi diterapkan di akun anggota (pelaporan).	

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pastikan EventBridge aturan berjalan sebelum pengujian.	Pastikan EventBridge aturan berjalan dengan menunggu setidaknya 24 jam, atau dengan meningkatkan frekuensi laporan dalam file template-reporting.yml. CloudFormation template. Untuk meningkatkan frekuensi laporan 1. Buka file template-reporting.ymldi repositori yang dikloning. 2. Dalam aturan acara dengan ID logis 'LambdaSc	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	hedule', temukan 'Schedule Expression'. 3. Edit tombol ScheduleE xpression" sehingga menyertakan ekspresi cron yang valid. Misalnya, ekspresi cron berikut menjadwalkan aturan acara untuk dijalankan setiap lima menit: `"cron (* /5 * * * *)"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa bucket Amazon S3 untuk laporan yang dihasilkan.	1. Di akun anggota (pelaporan), masuk ke AWS Management Console lalu buka CloudFormation konsol. 2. Di panel Stacks, pilih nama tumpukan yang Anda buat. Kemudian, pilih tab Sumber Daya. 3. Di panel Resources, di kolom Logical ID, temukan BackupReportS3Bucket. Kemudian, buka bucket Amazon S3 terkait di tab baru dengan memilih tautan di kolom ID Fisik di sebelah ID logis tersebut. 4. Pastikan bucket berisi laporan yang dihasilkan dalam format berikut: BackupReports////BackupReport- - - - .csv <yyyy><mm><dd><BACKUP JOB STATUS><dd><Mon><yyyy>	AWS DevOps, DevOps insinyur

Bersihkan sumber daya Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus komponen solusi dari akun anggota (pelaporan).	1. Di akun anggota (pelaporan), buka bucket Amazon S3 solusi. Untuk petunjuk,	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lihat langkah 2-4 di bucket Periksa S3 untuk cerita laporan yang dihasilkan dari bagian Uji solusi dari pola ini. 2. Hapus isi ember dan kosongkan ember. Untuk petunjuk, lihat Mengosongkan bucket di Panduan Pengguna Amazon S3. 3. Di akun anggota (pelaporan), masuk ke AWS Management Console lalu buka CloudFormation konsol. 4. Di panel Stacks, pilih kotak centang di sebelah nama tumpukan yang Anda buat. Lalu, pilih Hapus.	
Hapus komponen solusi dari akun manajemen.	1. Di akun manajemen, masuk ke AWS Management Console, lalu buka CloudFormation konsol. 2. Di panel Stacks, pilih kotak centang di sebelah nama tumpukan yang Anda buat. Lalu, pilih Hapus.	AWS DevOps, DevOps insinyur

Sumber daya terkait

- Tutorial: Menggunakan AWS Lambda dengan acara terjadwal (dokumentasi AWS Lambda)
- Membuat acara terjadwal untuk menjalankan fungsi AWS Lambda (AWS SDK untuk dokumentasi) JavaScript

- Tutorial IAM: Delegasikan akses di seluruh akun AWS menggunakan peran IAM (dokumentasi IAM)
- Terminologi dan konsep AWS Organizations (dokumentasi AWS Organizations)
- Membuat rencana laporan menggunakan konsol AWS Backup (dokumentasi AWS Backup)
- Membuat laporan audit (dokumentasi AWS Backup)
- Membuat laporan sesuai permintaan (dokumentasi AWS Backup)
- Apa itu AWS Backup? (Dokumentasi AWS Backup)
- Mengotomatiskan pencadangan terpusat pada skala di seluruh layanan AWS menggunakan AWS Backup (postingan blog AWS)

Ekspor tag untuk daftar EC2 instans Amazon ke file CSV

Sida Ju dan Pac Joonhyun, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara mengekspor tag secara terprogram untuk daftar instans Amazon Elastic Compute Cloud (Amazon EC2) ke file CSV.

Dengan menggunakan contoh skrip Python yang disediakan, Anda dapat mengurangi berapa lama waktu yang dibutuhkan untuk meninjau dan mengkategorikan instance EC2 Amazon Anda berdasarkan tag tertentu. Misalnya, Anda dapat menggunakan skrip untuk mengidentifikasi dan mengkategorikan daftar contoh yang ditandai oleh tim keamanan Anda untuk pembaruan perangkat lunak dengan cepat.

Prasyarat dan batasan

Prasyarat

- Python 3 diinstal dan dikonfigurasi
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi

Batasan

Contoh skrip Python yang disediakan dalam pola ini dapat mencari EC2 instance Amazon hanya berdasarkan atribut berikut:

- Contoh IDs
- IPv4 Alamat pribadi
- IPv4 Alamat publik

Alat

- Python adalah bahasa pemrograman komputer tujuan umum.
- virtualenv membantu Anda membuat lingkungan Python yang terisolasi.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.

Repositori kode

Contoh skrip Python untuk pola ini tersedia di repositori GitHub search-ec2 -. instances-export-tags

Epik

Instal dan konfigurasikan prasyarat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	**Note** Jika Anda menerima kesalahan saat menjalankan perintah AWS CLI, pastikan Anda menggunakan versi AWS CLI terbaru. Kloning instances-export-tags repositori GitHub search-ec2- dengan menjalankan perintah Git berikut di jendela terminal: `git clone https://github.com/aws-samples/search-ec2-instances-export-tags.git`	DevOps insinyur
Instal dan aktifkan virtualenv.	1. Instal virtualenv dengan menjalankan perintah berikut: `python3 -m pip install virtualenv`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Buat lingkungan virtual baru dengan menjalankan perintah berikut:
	`python3 -m venv env`
	3. Aktifkan lingkungan virtual baru dengan menjalankan perintah berikut:
	`source env/bin/activate`
	Untuk informasi selengkapnya, lihat Panduan Pengguna virtualenv.	

Instal dependensi.	1. Buka direktori kode dengan menjalankan perintah berikut di terminal:
	`cd search-ec2-instances-export-tags`
	2. Instal `requirements.txt` file dengan menjalankan perintah pip berikut:
	`pip3 install -r requirements.txt`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan profil bernama AWS.	Jika Anda belum melakukan nya, konfigurasikan profil bernama AWS yang menyertakan kredensil yang diperlukan untuk menjalank an skrip. Untuk membuat profil bernama, jalankan perintah aws configure. Untuk informasi selengkap nya, lihat Menggunakan profil bernama dalam dokumentasi AWS CLI.	DevOps insinyur

Konfigurasikan dan jalankan skrip Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file input.	Buat file input yang berisi daftar EC2 instance Amazon yang Anda inginkan skrip untuk mencari dan mengekspo r tag. Anda dapat mencantum kan contoh IDs, IPv4 alamat pribadi, atau IPv4 alamat publik. **Important** Pastikan bahwa setiap EC2 instance Amazon terdaftar pada barisnya sendiri di file input.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Contoh file masukan
	
	`1 i-0547c351bdfe85b9f`
	`2 54.157.194.156`
	`3 172.31.85.33`
	`4 54.165.198.144`
	`5 i-0b6223b5914111a4b`
	`6 172.31.85.44`
	`7 54.165.198.145`
	`8 172.31.80.219`
	`9 172.31.94.199`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip Python.	Jalankan skrip dengan menjalankan perintah berikut di terminal: `python search_instances.py -i INPUTFILE -o OUTPUTFILE -r REGION [-p PROFILE]` **Note** Ganti `INPUTFILE` dengan nama file input Anda. Ganti `OUTPUTFILE` dengan nama yang ingin Anda berikan pada file keluaran CSV. Ganti `REGION` dengan Wilayah AWS tempat EC2 sumber daya Amazon Anda berada. Jika Anda menggunakan profil bernama AWS, ganti `PROFILE` dengan profil bernama yang Anda gunakan. Untuk mendapatkan daftar parameter yang didukung dan deskripsinya, jalankan perintah berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`python search_instances.py -h` Untuk informasi selengkapnya dan untuk melihat contoh file keluaran, lihat `README.md` file di repositori GitHub search-ec2 -. instances-export-tags	

Sumber daya terkait

- Mengonfigurasi AWS CLI (Panduan Pengguna AWS CLI)

Buat CloudFormation template AWS yang berisi aturan terkelola AWS Config menggunakan Troposfer

Lucas Nation dan Freddie Wilson, Amazon Web Services

Ringkasan

Banyak organisasi menggunakan aturan terkelola AWS Config untuk mengevaluasi kepatuhan sumber daya Amazon Web Services (AWS) mereka terhadap praktik terbaik umum. Namun, aturan ini dapat memakan waktu lama untuk dipelihara dan pola ini membantu Anda memanfaatkan Troposfer, pustaka Python, untuk membuat dan mengelola aturan terkelola AWS Config.

Pola ini membantu Anda mengelola aturan terkelola AWS Config dengan menggunakan skrip Python untuk mengonversi spreadsheet Microsoft Excel yang berisi aturan terkelola AWS menjadi templat AWS. CloudFormation Troposfer bertindak sebagai infrastruktur sebagai kode (IAc) dan ini berarti Anda dapat memperbarui spreadsheet Excel dengan aturan terkelola, alih-alih menggunakan file berformat JSON atau YAML. Anda kemudian menggunakan template untuk meluncurkan CloudFormation tumpukan AWS yang membuat dan memperbarui aturan terkelola di akun AWS Anda.

CloudFormation Template AWS mendefinisikan setiap aturan yang dikelola AWS Config dengan menggunakan spreadsheet Excel dan membantu Anda menghindari pembuatan aturan individual secara manual di AWS Management Console. Skrip default setiap parameter aturan terkelola ke kamus kosong dan default cakupan dari. *`ComplianceResourceTypes`* *`THE_RULE_IDENTIFIER.template file`* Untuk informasi selengkapnya tentang pengenal aturan, lihat Membuat aturan terkelola AWS Config dengan templat AWS dalam CloudFormation dokumentasi AWS Config.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Keakraban dengan menggunakan CloudFormation templat AWS untuk membuat aturan terkelola AWS Config. Untuk informasi selengkapnya tentang hal ini, lihat Membuat aturan terkelola AWS Config dengan CloudFormation templat AWS dalam dokumentasi AWS Config.
- Python 3, diinstal dan dikonfigurasi. Untuk informasi lebih lanjut tentang ini, lihat dokumentasi Python.

- Lingkungan pengembangan terintegrasi (IDE) yang ada.
- Identifikasi unit organisasi Anda (OUs) dalam kolom dalam contoh spreadsheet `excel_config_rules.xlsx` Excel (terlampir).

Epik

Kustomisasi dan konfigurasikan aturan terkelola AWS Config

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui contoh spreadsheet Excel.	Unduh contoh spreadsheet `excel_config_rules .xlsx` Excel (terlampir) dan beri label sebagai aturan `Implemented` terkelola AWS Config yang ingin Anda gunakan. Aturan yang ditandai sebagai `Implemented` akan ditambahkan ke CloudForm ation template AWS.	Developer
(Opsional) Perbarui file config_rules_params.json dengan parameter aturan AWS Config.	Beberapa aturan yang dikelola AWS Config memerlukan parameter dan harus diteruska n ke skrip Python sebagai file JSON dengan menggunak an opsi. `--param-file` Misalnya, aturan `access-keys-rotated` terkelola menggunakan `maxAccess KeyAge` parameter berikut: `{ "access-keys-rotated": {`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"InputPar
	ameters": {
	
	"maxAccessKeyAge": 90
	}
	}
	}</pre>
	Dalam parameter sampel ini, `maxAccessKeyAge` diatur ke 90 hari. Skrip membaca file parameter dan menambahkan apa pun `InputParameters` yang ditemukannya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Perbarui file config_rules_params.json dengan AWS Config. ComplianceResourceTypes	Secara default, skrip Python mengambil template yang ditentukan `ComplianceResourceTypes` AWS. Jika Anda ingin mengganti cakupan aturan terkelola AWS Config tertentu, maka Anda harus meneruskannya ke skrip Python sebagai file JSON menggunakan opsi. `--param-file` Misalnya, kode contoh berikut menunjukkan bagaimana `ComplianceResourceTypes` for `ec2-volume-inuse-check` diatur ke `["AWS::EC2::Volume"]` daftar: `{ "ec2-volume-inuse-check": { "Scope": { "ComplianceResourceTypes": [ "AWS::EC2::Volume" ] } } }`	Developer

Jalankan skrip Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal paket pip dari file requirements.txt.	Download `requirements.txt` file (terlampir) dan jalankan perintah berikut di IDE Anda untuk menginstal paket Python: `pip3 install -r requirements.txt`	Developer
Jalankan skrip Python.	1. Unduh `aws_config_rules.py` file (terlampir) ke mesin lokal Anda. 2. **Note** Jalankan `python3 aws_config_rules.py --ou <OU_NAME>` perintah.: `--ou` mendefinisikan kolom OU mana yang akan dipilih di spreadsheet Excel. Anda juga dapat menambahkan parameter opsional berikut: • `--config-rule-option` — Mendefinisikan aturan untuk	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memilih dari spreadsheet Excel. Defaultnya adalah `Implemented` parameter nya. • `--excel-file` — Jalur untuk spreadsheet Excel. Nilai default-nya `aws_config_rules.xlsx` . • `--param-file` — Jalur file parameter JSON. Nilai default-nya `config_rules_params.json` . • `--max-execution-frequency` — Menentukan seberapa sering aturan terkelola AWS Config dievaluasi. Pilihannya adalah`One_Hour`,`Three_Hours` ,`Six_Hours` ,`Twelve_Hours` , atau`TwentyFour_Hours` . Nilai default-nya `TwentyFour_Hours` .	

Menerapkan aturan terkelola AWS Config

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan CloudFormation tumpukan AWS.	1. Masuk ke AWS Management Console, buka CloudFormation konsol AWS, lalu pilih Buat tumpukan.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pada halaman Tentukan templat, pilih Unggah file templat lalu unggah CloudFormation templat AWS Anda. 3. Tentukan nama tumpukan lalu pilih Berikutnya. 4. Tentukan tag dan kemudian pilih Berikutnya. 5. Pilih Buat tumpukan.	

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Berikan instans SageMaker notebook akses sementara ke CodeCommit repositori di akun AWS lain

Healge Aufderheide, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memberikan instans SageMaker notebook Amazon dan pengguna akses sementara ke CodeCommit repositori AWS yang ada di akun AWS lain. Pola ini juga menunjukkan bagaimana Anda dapat memberikan izin granular untuk tindakan tertentu yang dapat dilakukan setiap entitas pada setiap repositori.

Organizations sering menyimpan CodeCommit repositori di akun AWS yang berbeda dari akun yang menampung lingkungan pengembangannya. Pengaturan multi-akun ini membantu mengontrol akses ke repositori dan mengurangi risiko penghapusan yang tidak disengaja. Untuk memberikan izin lintas akun ini, merupakan praktik terbaik untuk menggunakan peran AWS Identity and Access Management (IAM). Kemudian, identitas IAM yang telah ditentukan sebelumnya di setiap akun AWS dapat mengambil peran sementara untuk membuat rantai kepercayaan yang terkontrol di seluruh akun.

> **Note**
>
> Anda dapat menerapkan prosedur serupa untuk memberikan identitas IAM lainnya akses lintas akun ke repositori. CodeCommit Untuk informasi selengkapnya, lihat Mengonfigurasi akses lintas akun ke CodeCommit repositori AWS menggunakan peran dalam Panduan Pengguna CodeCommit AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan CodeCommit repositori (akun A)
- Akun AWS aktif kedua dengan instance SageMaker notebook (akun B)
- Pengguna AWS dengan izin yang memadai untuk membuat dan memodifikasi peran IAM di akun A
- Pengguna AWS kedua dengan izin yang cukup untuk membuat dan memodifikasi peran IAM di akun B

Arsitektur

Diagram berikut menunjukkan contoh alur kerja untuk memberikan instance SageMaker notebook dan pengguna dalam satu akun AWS akses lintas akun ke repositori: CodeCommit

Diagram menunjukkan alur kerja berikut:

1. Peran pengguna AWS dan peran instans SageMaker notebook di akun B mengasumsikan profil bernama.
2. Kebijakan izin profil bernama menentukan peran CodeCommit akses di akun A yang kemudian diasumsikan oleh profil.
3. Kebijakan kepercayaan peran CodeCommit akses di akun A memungkinkan profil bernama di akun B untuk mengambil peran CodeCommit akses.
4. Kebijakan izin IAM CodeCommit repositori di akun A memungkinkan peran CodeCommit akses untuk mengakses repositori. CodeCommit

Tumpukan teknologi

- CodeCommit
- Git
- IAM
- pip
- SageMaker

Alat

- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Git adalah sistem kontrol versi terdistribusi untuk melacak perubahan kode sumber selama pengembangan perangkat lunak.

- git-remote-codecommitadalah utilitas yang membantu Anda mendorong dan menarik kode dari CodeCommit repositori dengan memperluas Git.
- pip adalah penginstal paket untuk Python. Anda dapat menggunakan pip untuk menginstal paket dari Python Package Index dan indeks lainnya.

Praktik terbaik

Saat Anda menetapkan izin dengan kebijakan IAM, pastikan Anda hanya memberikan izin yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa paling sedikit dalam dokumentasi IAM.

Saat menerapkan pola ini, pastikan Anda melakukan hal berikut:

- Konfirmasikan bahwa prinsip IAM hanya memiliki izin yang diperlukan untuk melakukan tindakan spesifik yang diperlukan dalam setiap repositori. Misalnya, disarankan untuk mengizinkan prinsip IAM yang disetujui untuk mendorong dan menggabungkan perubahan ke cabang repositori tertentu, tetapi hanya meminta penggabungan ke cabang yang dilindungi.
- Konfirmasikan bahwa prinsip-prinsip IAM diberi peran IAM yang berbeda berdasarkan peran dan tanggung jawab masing-masing untuk setiap proyek. Misalnya, pengembang akan memiliki izin akses yang berbeda dari manajer rilis atau AWS Administrator.

Epik

Konfigurasikan peran IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kebijakan peran CodeCommit akses dan izin.	**Note** Untuk mengotoma tiskan proses penyiapan manual yang didokumen tasikan dalam epik ini, Anda dapat menggunakan	AWS Umum, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CloudFormation template AWS. Di akun yang berisi CodeCommit repositori (akun A), lakukan hal berikut: 1. Buat peran IAM yang dapat diasumsikan oleh peran instance SageMaker notebook di akun B. 2. Buat kebijakan IAM yang memberikan akses ke repositori dan lampirkan kebijakan ke peran tersebut. Hanya untuk tujuan pengujian, pilih kebijakan terkelola AWSCodeCommitPowerUserAWS. Kebijakan ini memberikan semua CodeCommit izin kecuali kemampuan untuk menghapus sumber daya. 3. Ubah kebijakan kepercayaan peran sehingga akun B terdaftar sebagai entitas tepercaya. ⚠ Important Sebelum memindahkan penyiapan ini ke	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lingkungan produksi Anda, sebaiknya tulis kebijakan IAM Anda sendiri yang menerapkan izin hak istimewa paling sedikit. Untuk informasi selengkapnya, lihat bagian Informasi tambahan dari pola ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan peran instans SageMaker notebook dalam izin akun B untuk mengambil peran CodeCommit akses di akun A.	Di akun yang berisi peran IAM instance SageMaker notebook (akun B), lakukan hal berikut: 1. Buat kebijakan IAM yang memungkinkan peran IAM atau pengguna untuk mengambil peran CodeCommit akses di akun A. Contoh kebijakan izin IAM yang memungkinkan peran IAM atau pengguna untuk mengambil peran lintas akun `{ "Version": "2012-10-17", "Statement": [ { "Sid": "VisualEditor0", "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam:::accountA_ID:role/accountArole_ID" } }` 2. Lampirkan kebijakan ke peran instans SageMaker buku catatan Anda di akun B.	AWS Umum, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Mintalah peran instance SageMaker notebook di akun B mengasumsikan peran CodeCommit akses di akun A. **Note** Untuk melihat Nama Sumber Daya Amazon (ARN) repositori Anda, lihat Melihat detail CodeCommit repositori di Panduan Pengguna AWS. CodeCommit	

Siapkan instance SageMaker buku catatan Anda di akun B

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan profil pengguna pada instans SageMaker notebook AWS untuk mengambil peran dalam akun A.	**Important** Pastikan Anda telah menginstal AWS Command Line Interface (AWS CLI) versi terbaru. Di akun yang berisi instance SageMaker notebook (akun B), lakukan hal berikut:	AWS Umum, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Masuk ke AWS Management Console dan buka SageMaker konsol. 2. Akses instance SageMaker notebook Anda. Antarmuka Jupyter terbuka. 3. Pilih Baru, lalu pilih Terminal. Jendela terminal baru terbuka di lingkungan Jupyter Anda. 4. Arahkan ke file ~/.aws/config instance SageMaker notebook. Kemudian, tambahkan profil pengguna ke file dengan memasukkan pernyataan berikut: `------.aws/config-------------[profile remoteprepouser]role_arn = arn:aws:iam::<ID of Account A>:role/<rolename>role_session_name = remoteaccesssessionregion = eu-west-1credential_source = Ec2InstanceMetadata----------------------------------`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal git-remote-codecommit utilitas.	Ikuti petunjuk di Langkah 2: Instal git-remote-codecommit di Panduan CodeCommit Pengguna AWS.	Ilmuwan data

Akses repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses CodeCommit repositori dengan menggunakan perintah Git atau. SageMaker	Untuk menggunakan Git Prinsipal IAM yang mengasumsikan peran instance SageMaker notebook di akun B sekarang dapat menjalankan perintah Git untuk mengakses CodeCommit repositori di akun A. Misalnya, pengguna dapat menjalankan perintah seperti,, dan. `git clone` `git pull` `git push` Untuk petunjuk, lihat Connect ke CodeCommit repositori AWS di CodeCommit Panduan Pengguna AWS. Untuk informasi tentang cara menggunakan Git CodeCommit, lihat Memulai AWS CodeCommit di Panduan CodeCommit Pengguna AWS.	Git, konsol bash

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk menggunakan SageMaker Untuk menggunakan Git dari SageMaker konsol, Anda harus mengizinkan Git untuk mengambil kredensi dari repositori Anda. CodeCommi t Untuk petunjuknya, lihat Mengaitkan CodeCommit repositori di akun AWS yang berbeda dengan instans notebook dalam dokumentasi. SageMaker	

Sumber daya terkait

- Konfigurasikan akses lintas akun ke CodeCommit repositori AWS menggunakan peran (dokumentasi AWS) CodeCommit
- Tutorial IAM: Delegasikan akses di seluruh akun AWS menggunakan peran IAM (dokumentasi IAM)

Informasi tambahan

Membatasi CodeCommit izin untuk tindakan tertentu

Untuk membatasi tindakan yang dapat dilakukan oleh kepala sekolah IAM di CodeCommit repositori, ubah tindakan yang diizinkan dalam kebijakan akses. CodeCommit

Untuk informasi selengkapnya tentang operasi CodeCommit API, lihat referensi CodeCommit izin di Panduan CodeCommit Pengguna AWS.

> **Note**
>
> Anda juga dapat mengedit kebijakan terkelola AWSCodeCommitPowerUserAWS agar sesuai dengan kasus penggunaan Anda.

Membatasi CodeCommit izin ke repositori tertentu

Untuk membuat lingkungan multitenant di mana lebih dari satu repositori kode hanya dapat diakses oleh pengguna tertentu, lakukan hal berikut:

1. Buat beberapa peran CodeCommit akses di akun A. Kemudian, konfigurasikan kebijakan kepercayaan setiap peran akses agar pengguna tertentu di akun B dapat mengambil peran tersebut.
2. Batasi repositori kode apa yang dapat diasumsikan oleh setiap peran dengan menambahkan kondisi “Sumber Daya” ke kebijakan setiap peran CodeCommit akses.

Contoh kondisi “Sumber Daya” yang membatasi akses prinsipal IAM ke repositori tertentu CodeCommit

```
"Resource" : [<REPOSITORY_ARN>,<REPOSITORY_ARN> ]
```

> **Note**
>
> Untuk membantu mengidentifikasi dan membedakan beberapa repositori kode di akun AWS yang sama, Anda dapat menetapkan awalan yang berbeda ke nama repositori. Misalnya, Anda dapat memberi nama repositori kode dengan awalan yang sejajar dengan grup pengembang yang berbeda, seperti myproject-subproject1-repo1 dan myproject-subproject2-repo1. Kemudian, Anda dapat membuat peran IAM untuk setiap grup pengembang berdasarkan awalan yang ditetapkan. Misalnya, Anda dapat membuat peran bernama myproject-subproject1-repoaccess dan memberikannya akses ke semua repositori kode yang menyertakan awalan myproject-subproject1.

Contoh kondisi “Resource” yang mengacu pada ARN repositori kode yang menyertakan awalan tertentu

```
"Resource" : arn:aws:codecommit:<region>:<account-id>:myproject-subproject1-*
```

Menerapkan strategi percabangan GitHub Flow untuk lingkungan multi-akun DevOps

Mike Stephens dan Abhilash Vinod, Amazon Web Services

Ringkasan

Saat mengelola repositori kode sumber, strategi percabangan yang berbeda memengaruhi pengembangan perangkat lunak dan proses rilis yang digunakan tim pengembangan. Contoh strategi percabangan umum termasuk Trunk, GitHub Flow, dan Gitflow. Strategi-strategi ini menggunakan cabang yang berbeda, dan kegiatan yang dilakukan di setiap lingkungan berbeda. Organizations yang menerapkan DevOps proses akan mendapat manfaat dari panduan visual untuk membantu mereka memahami perbedaan antara strategi percabangan ini. Menggunakan visual ini di organisasi Anda membantu tim pengembangan menyelaraskan pekerjaan mereka dan mengikuti standar organisasi. Pola ini memberikan visual ini dan menjelaskan proses penerapan strategi percabangan GitHub Flow di organisasi Anda.

Pola ini merupakan bagian dari rangkaian dokumentasi tentang memilih dan menerapkan strategi DevOps percabangan untuk organisasi dengan banyak Akun AWS. Seri ini dirancang untuk membantu Anda menerapkan strategi dan praktik terbaik yang benar sejak awal, untuk merampingkan pengalaman Anda di cloud. GitHub Flow hanyalah salah satu strategi percabangan yang mungkin dapat digunakan organisasi Anda. Seri dokumentasi ini juga mencakup model percabangan Trunk dan Gitflow. Jika Anda belum melakukannya, kami sarankan Anda meninjau Memilih strategi percabangan Git untuk DevOps lingkungan multi-akun sebelum menerapkan panduan dalam pola ini. Harap gunakan uji tuntas untuk memilih strategi percabangan yang tepat untuk organisasi Anda.

Panduan ini menyediakan diagram yang menunjukkan bagaimana organisasi dapat menerapkan strategi GitHub Flow. Disarankan agar Anda meninjau Panduan AWS DevOps Well-Architected untuk meninjau praktik terbaik. Pola ini mencakup tugas, langkah, dan batasan yang direkomendasikan untuk setiap langkah dalam DevOps proses.

Prasyarat dan batasan

Prasyarat

- Git, diinstal. Ini digunakan sebagai alat repositori kode sumber.
- Draw.io, diinstal. Aplikasi ini digunakan untuk melihat dan mengedit diagram.

Arsitektur

Arsitektur target

Diagram berikut dapat digunakan seperti kotak Punnett (Wikipedia). Anda menyusun cabang pada sumbu vertikal dengan AWS lingkungan pada sumbu horizontal untuk menentukan tindakan apa yang harus dilakukan dalam setiap skenario. Angka-angka menunjukkan urutan tindakan dalam alur kerja. Contoh ini membawa Anda dari `feature` cabang melalui penerapan dalam produksi.

Untuk informasi selengkapnya tentang Akun AWS, lingkungan, dan cabang dalam pendekatan GitHub Flow, lihat Memilih strategi percabangan Git untuk lingkungan multi-akun DevOps .

Otomatisasi dan skala

Integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) is the process of automating the software release lifecycle. It automates much or all of the manual processes traditionally required to get new code from an initial commit into production. A CI/CD pipeline encompasses the sandbox, development, testing, staging, and production environments. In each environment, the CI/CD pipeline provisions any infrastructure that is needed to deploy or test the code. By using CI/CD, development teams can make changes to code that are then automatically tested and deployed. CI/CDjaringan pipa juga menyediakan tata kelola dan pagar pembatas untuk tim pengembangan dengan menegakkan konsistensi, standar, praktik terbaik, dan tingkat penerimaan minimal untuk penerimaan dan penerapan fitur. Untuk informasi selengkapnya, lihat Mempraktikkan Integrasi Berkelanjutan dan Pengiriman Berkelanjutan di AWS.

AWS menawarkan serangkaian layanan pengembang yang dirancang untuk membantu Anda membangun CI/CD jaringan pipa. Misalnya, AWS CodePipelineadalah layanan pengiriman berkelanjutan yang dikelola sepenuhnya yang membantu Anda mengotomatiskan saluran pipa rilis untuk pembaruan aplikasi dan infrastruktur yang cepat dan andal. AWS CodeBuildmengkompilasi kode sumber, menjalankan tes, dan menghasilkan paket ready-to-deploy perangkat lunak. Untuk informasi selengkapnya, lihat Alat Pengembang di AWS.

Alat

AWS layanan dan alat

AWS menyediakan serangkaian layanan pengembang yang dapat Anda gunakan untuk menerapkan pola ini:

- AWS CodeArtifactadalah layanan repositori artefak terkelola yang sangat skalabel yang membantu Anda menyimpan dan berbagi paket perangkat lunak untuk pengembangan aplikasi.
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeDeploymengotomatiskan penerapan ke Amazon Elastic Compute Cloud (Amazon EC2) atau instans, fungsi AWS Lambda , atau layanan Amazon Elastic Container Service (Amazon ECS) atau lokal.
- AWS CodePipelinemembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.

Alat-alat lainnya

- Draw.io Desktop adalah aplikasi untuk membuat diagram alur dan diagram. Repositori kode berisi template dalam format.drawio untuk Draw.io.
- Figma adalah alat desain online yang dirancang untuk kolaborasi. Repositori kode berisi template dalam format.fig untuk Figma.

Repositori kode

File sumber untuk diagram dalam pola ini tersedia di GitHub Git Branching Strategy for GitHub Flow repository. Ini termasuk file dalam format PNG, draw.io, dan Figma. Anda dapat memodifikasi diagram ini untuk mendukung proses organisasi Anda.

Praktik terbaik

Ikuti praktik dan rekomendasi terbaik dalam Panduan AWS DevOps Well-Architected dan Memilih strategi percabangan Git untuk lingkungan multi-akun. DevOps Ini membantu Anda menerapkan pengembangan GitHub berbasis Flow secara efektif, mendorong kolaborasi, meningkatkan kualitas kode, dan merampingkan proses pengembangan.

Epik

Meninjau alur kerja GitHub Alur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau proses GitHub Flow standar.	1. Di lingkungan kotak pasir, pengembang membuat `feature` cabang dari `main` cabang dan menggunakan pola `feature/<ticket>_<initials>_<short description>` penamaan. 2. Pengembang menambahkan satu atau lebih komit ke `feature` cabang, masing-masing mewakili perubahan atau peningkatan diskrit. 3. Pengembang membuka permintaan gabungan (MR) untuk menggabungkan perubahan ke cabang. `main` Ini memulai proses peninjauan. 4. Selama proses peninjauan, pengembang mendiskusikan perubahan kode dan memberikan umpan balik. Tujuannya adalah untuk memastikan bahwa perubahannya berkualitas tinggi dan memenuhi standar proyek. 5. Setelah pengembang membuat permintaa	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n gabungan, proses pembuatan otomatis dimulai dan menerapka n perubahan di `feature` cabang ke lingkungan pengembangan. 6. Pengujian otomatis memverifikasi integritas dan kualitas perubahan yang dienkapsulasi dalam permintaan gabungan. Pembuatan yang berhasil, penerapan yang berhasil, dan pengujian yang berhasil diperlukan untuk menyelesa ikan permintaan penggabun gan. 7. Ketika proses peninjaua n selesai, perubahan digabungkan ke dalam `main` cabang. 8. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan pengujian. 9. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan pementasan. 10. Pemberi persetujuan secara manual menyetujui penyebaran artefak rilis ke lingkungan produksi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau proses GitHub Alur perbaikan bug.	1. Pengembang membuat `bugfix` cabang dari `main` cabang dan menggunakan pola penamaan`bugfix/<ticket number>_<developer initials>_<descriptor>` . 2. Pengembang memperbaiki masalah, melakukan perbaikan, dan membangun cabang. `bugfix` 3. Pengembang membuka permintaan gabungan untuk menggabungkan `bugfix` cabang ke cabang. `main` Ini memulai proses peninjauan. 4. Selama proses peninjauan, pengembang mendiskusikan perubahan kode dan memberikan umpan balik. 5. Setelah peninjauan selesai dan disetujui, pengembang menyelesaikan permintaan penggabungan `bugfix` cabang ke cabang. `main` 6. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan yang lebih tinggi.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau proses GitHub Flow hotfix.	GitHub Flow dirancang untuk memungkinkan pengirima n berkelanjutan, di mana perubahan kode sering dan andal digunakan ke lingkunga n yang lebih tinggi. Kuncinya adalah bahwa setiap `feature` cabang dapat diterapkan kapan saja. `Hotfix`cabang, yang mirip dengan `feature` atau `bugfix` cabang, dapat mengikuti proses yang sama seperti salah satu cabang lainnya. Namun, mengingat urgensinya, hotfix biasanya memiliki prioritas yang lebih tinggi. Bergantung pada kebijakan tim dan kedekatan situasi, langkah-langkah tertentu dalam proses dapat dipercepat. Misalnya, ulasan kode untuk perbaikan terbaru mungkin dilacak dengan cepat. Oleh karena itu, sementara proses perbaikan terbaru sejajar dengan fitur atau proses perbaikan bug, urgensi seputar perbaikan terbaru dapat memerlukan modifikasi dalam kepatuhan prosedural. Sangat penting untuk menetapkan pedoman tentang mengelola perbaikan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	terbaru untuk memastikan bahwa mereka ditangani secara efisien dan aman.	

Pemecahan masalah

Isu	Solusi
Konflik cabang	Masalah umum yang dapat terjadi dengan model GitHub Flow adalah di mana perbaikan terbaru perlu terjadi dalam produksi tetapi perubahan yang sesuai perlu terjadi di`feature`,`bugfix`, atau `hotfix` cabang di mana sumber daya yang sama sedang dimodifikasi. Kami menyarankan agar Anda sering menggabungkan perubahan dari `main` cabang yang lebih rendah untuk menghindari konflik yang signifikan saat Anda bergabung. `main`
Kedewasaan tim	GitHub Flow mendorong penyebaran harian ke lingkungan yang lebih tinggi, merangkul integrasi berkelanjutan sejati dan pengiriman berkelanjutan (CI/CD). Sangat penting bahwa tim memiliki kematangan teknik untuk membangun fitur dan membuat tes otomatisasi untuk mereka. Tim harus melakukan peninjauan permintaan gabungan lengkap sebelum perubahan disetujui. Ini menumbuhkan budaya teknik yang kuat yang mempromosikan kualitas, akuntabilitas, dan efisiensi dalam proses pengembangan.

Sumber daya terkait

Panduan ini tidak termasuk pelatihan untuk Git; namun, ada banyak sumber daya berkualitas tinggi yang tersedia di internet jika Anda memerlukan pelatihan ini. Kami menyarankan Anda memulai dengan situs dokumentasi Git.

Sumber daya berikut dapat membantu Anda dengan perjalanan percabangan GitHub Flow Anda di. AWS Cloud

AWS DevOps bimbingan

- AWS DevOps Bimbingan
- AWS Arsitektur Referensi Pipeline Deployment
- Apa itu DevOps?
- DevOps sumber daya

GitHub Panduan aliran

- GitHub Tutorial Mulai Cepat Aliran () GitHub
- Mengapa GitHub Mengalir?

Sumber daya lainnya

- Metodologi aplikasi dua belas faktor (12factor.net)

Menerapkan strategi percabangan Gitflow untuk lingkungan multi-akun DevOps

Mike Stephens, Stephen, Abhilash Vinod DiCato, dan Tim Wondergem, Amazon Web Services

Ringkasan

Saat mengelola repositori kode sumber, strategi percabangan yang berbeda memengaruhi pengembangan perangkat lunak dan proses rilis yang digunakan tim pengembangan. Contoh strategi percabangan umum termasuk Trunk, Gitflow, dan GitHub Flow. Strategi-strategi ini menggunakan cabang yang berbeda, dan kegiatan yang dilakukan di setiap lingkungan berbeda. Organizations yang menerapkan DevOps proses akan mendapat manfaat dari panduan visual untuk membantu mereka memahami perbedaan antara strategi percabangan ini. Menggunakan visual ini di organisasi Anda membantu tim pengembangan menyelaraskan pekerjaan mereka dan mengikuti standar organisasi. Pola ini memberikan visual ini dan menjelaskan proses penerapan strategi percabangan Gitflow di organisasi Anda.

Pola ini merupakan bagian dari rangkaian dokumentasi tentang memilih dan menerapkan strategi DevOps percabangan untuk organisasi dengan banyak Akun AWS. Seri ini dirancang untuk membantu Anda menerapkan strategi dan praktik terbaik yang benar sejak awal, untuk merampingkan pengalaman Anda di cloud. Gitflow hanyalah salah satu kemungkinan strategi percabangan yang dapat digunakan organisasi Anda. Seri dokumentasi ini juga mencakup model percabangan Trunk dan GitHub Flow. Jika Anda belum melakukannya, kami sarankan Anda meninjau Memilih strategi percabangan Git untuk DevOps lingkungan multi-akun sebelum menerapkan panduan dalam pola ini. Harap gunakan uji tuntas untuk memilih strategi percabangan yang tepat untuk organisasi Anda.

Panduan ini menyediakan diagram yang menunjukkan bagaimana organisasi dapat menerapkan strategi Gitflow. Disarankan agar Anda meninjau Panduan AWS DevOps Well-Architected untuk meninjau praktik terbaik. Pola ini mencakup tugas, langkah, dan batasan yang direkomendasikan untuk setiap langkah dalam DevOps proses.

Prasyarat dan batasan

Prasyarat

- Git, diinstal. Ini digunakan sebagai alat repositori kode sumber.
- Draw.io, diinstal. Aplikasi ini digunakan untuk melihat dan mengedit diagram.
- (Opsional) Plugin Gitflow, diinstal.

Arsitektur

Arsitektur target

Diagram berikut dapat digunakan seperti kotak Punnett (Wikipedia). Anda menyusun cabang pada sumbu vertikal dengan AWS lingkungan pada sumbu horizontal untuk menentukan tindakan apa yang harus dilakukan dalam setiap skenario. Angka-angka menunjukkan urutan tindakan dalam alur kerja. Contoh ini membawa Anda dari cabang fitur melalui penerapan dalam produksi.

Untuk informasi selengkapnya tentang Akun AWS, lingkungan, dan cabang dalam pendekatan Gitflow, lihat Memilih strategi percabangan Git untuk lingkungan DevOps multi-akun.

Otomatisasi dan skala

Integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) is the process of automating the software release lifecycle. It automates much or all of the manual processes traditionally required to get new code from an initial commit into production. A CI/CD pipeline encompasses the sandbox, development, testing, staging, and production environments. In each environment, the CI/CD pipeline provisions any infrastructure that is needed to deploy or test the code. By using CI/CD, development teams can make changes to code that are then automatically tested and deployed. CI/CDjaringan pipa juga menyediakan tata kelola dan pagar pembatas untuk tim pengembangan dengan menegakkan konsistensi, standar, praktik terbaik, dan tingkat penerimaan minimal untuk penerimaan dan penerapan fitur. Untuk informasi selengkapnya, lihat Mempraktikkan Integrasi Berkelanjutan dan Pengiriman Berkelanjutan di AWS.

AWS menawarkan serangkaian layanan pengembang yang dirancang untuk membantu Anda membangun CI/CD jaringan pipa. Misalnya, AWS CodePipelineadalah layanan pengiriman berkelanjutan yang dikelola sepenuhnya yang membantu Anda mengotomatiskan saluran pipa rilis untuk pembaruan aplikasi dan infrastruktur yang cepat dan andal. AWS CodeBuildmengkompilasi kode sumber, menjalankan tes, dan menghasilkan paket ready-to-deploy perangkat lunak. Untuk informasi selengkapnya, lihat Alat Pengembang di AWS.

Alat

AWS layanan dan alat

AWS menyediakan serangkaian layanan pengembang yang dapat Anda gunakan untuk menerapkan pola ini:

- AWS CodeArtifactadalah layanan repositori artefak terkelola yang sangat skalabel yang membantu Anda menyimpan dan berbagi paket perangkat lunak untuk pengembangan aplikasi.
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeDeploymengotomatiskan penerapan ke Amazon Elastic Compute Cloud (Amazon EC2) atau instans, fungsi AWS Lambda , atau layanan Amazon Elastic Container Service (Amazon ECS) atau lokal.
- AWS CodePipelinemembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.

Alat lainnya

- Draw.io Desktop adalah aplikasi untuk membuat diagram alur dan diagram. Repositori kode berisi template dalam format.drawio untuk Draw.io.
- Figma adalah alat desain online yang dirancang untuk kolaborasi. Repositori kode berisi template dalam format.fig untuk Figma.
- (Opsional) Plugin Gitflow adalah kumpulan ekstensi Git yang menyediakan operasi repositori tingkat tinggi untuk model percabangan Gitflow.

Repositori kode

File sumber untuk diagram dalam pola ini tersedia dalam Strategi Percabangan GitHub Git untuk GitFlow repositori. Ini termasuk file dalam format PNG, draw.io, dan Figma. Anda dapat memodifikasi diagram ini untuk mendukung proses organisasi Anda.

Praktik terbaik

Ikuti praktik dan rekomendasi terbaik dalam Panduan AWS DevOps Well-Architected dan Memilih strategi percabangan Git untuk lingkungan multi-akun. DevOps Ini membantu Anda menerapkan pengembangan berbasis GitFlow secara efektif, mendorong kolaborasi, meningkatkan kualitas kode, dan merampingkan proses pengembangan.

Epik

Meninjau alur kerja Gitflow

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau proses Gitflow standar.	1. Di lingkungan kotak pasir, pengembang membuat `feature` cabang dari `develop` cabang dan menggunakan pola `feature/<ticket>_<initials>_<short description>` penamaan. 2. Pengembang mengembangkan kode dan menyebarkan kode ke lingkungan kotak pasir secara iteratif untuk menyelesaikan tiket. **Note** Pengembang secara opsional dapat membuat `sandbox` cabang untuk menjalankan build otomatis atau deploy pipeline di lingkungan kotak pasir. 3. Pengembang membuat permintaan gabungan dari `feature` cabang ke `develop` cabang dengan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggunakan gabungan squash. 4. Pipa integrasi berkelanjutan dan pengiriman berkelanj utan (CI/CD) secara otomatis membangun dan menyebarkan `develop` cabang ke lingkungan pengembangan. 5. (Opsional) Pengembang mengintegrasikan `feature` cabang tambahan ke dalam cabang pengembangan sebelum melanjutkan aktivitas rilis. 6. Ketika Anda siap untuk merilis fitur di `develop` cabang, pengembang membuat `release` cabang bernama `release/v <number>` dari `develop` cabang. 7. Pengembang membangun cabang rilis, yang menerbitk an artefak untuk digunakan kembali di lingkungan lain. 8. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan pengujian. 9. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan pementasan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	10. Penyetuju secara manual menyetujui penyebaran artefak rilis ke lingkungan produksi. 11. Pengembang menggabungkan `release` cabang ke `main` cabang. Idealnya, pengembang menggunakan skrip otomatis untuk melakukan penggabungan maju cepat. Jangan gunakan gabungan squash. 12. Pengembang menggabungkan `release` cabang ke `develop` cabang. Idealnya, pengembang menggunakan skrip otomatis untuk melakukan penggabungan maju cepat. Jangan gunakan gabungan squash.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau proses Gitflow hotfix.	1. Pengembang membuat `hotfix` cabang dari `main` cabang dan menggunakan pola penamaan`hotfix/<ticket>_<initials>_<short description>` . 2. Pengembang membuat `release` cabang dari `main` cabang dan menamainy `arelease/v<number>` . 3. Pengembang memperbaiki masalah, melakukan perbaikan, dan membangun cabang. `hotfix` 4. Pengembang membuat permintaan gabungan dari `hotfix` cabang ke `release/v<number>` cabang dengan menggunakan gabungan squash. 5. Pengembang membangun `release` cabang, yang menerbitkan artefak untuk digunakan kembali di lingkungan lain. 6. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan pengujian. 7. Penyetuju secara manual menyetujui penerapan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	artefak rilis ke lingkungan pementasan. 8. Penyetuju secara manual menyetujui penyebaran artefak rilis ke lingkungan produksi. 9. Pengembang menggabungkan `release` cabang ke `main` cabang. Idealnya, pengembang menggunakan skrip otomatis untuk melakukan penggabungan maju cepat. Jangan gunakan gabungan squash. 10. Pengembang menggabungkan `release` cabang ke `develop` cabang. Idealnya, pengembang menggunakan skrip otomatis untuk melakukan penggabungan maju cepat. Jangan gunakan gabungan squash. 11. Jika konflik terdeteksi, pengembang menerima peringatan dan menyelesaikan konflik dengan permintaan gabungan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau proses Gitflow perbaikan bug.	1. Pengembang membuat `bugfix` cabang dari `release/v<number>` cabang saat ini dan menggunakan pola penamaan`bugfix/<ticket number>_<developer initials>_<descriptor>` . 2. Pengembang memperbaiki masalah, melakukan perbaikan, dan membangun cabang. `bugfix` 3. Pengembang membuat permintaan gabungan dari `bugfix` cabang ke `release/v<number>` cabang dengan menggunakan gabungan squash. 4. Pengembang membangun `release` cabang, yang menerbitkan artefak untuk digunakan kembali di lingkungan lain. 5. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan Pengujian. 6. Pemberi persetujuan secara manual menyetujui penerapan artefak rilis ke lingkungan Stage.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Pemberi persetujuan secara manual menyetuju i penerapan artefak rilis ke lingkungan Produksi. 8. Pengembang menggabun gkan `release` cabang ke `main` cabang. Idealnya, pengembang menggunak an skrip otomatis untuk melakukan penggabun gan maju cepat. Jangan gunakan gabungan squash. 9. Pengembang menggabun gkan `release` cabang ke `develop` cabang. Idealnya, pengembang menggunak an skrip otomatis untuk melakukan penggabun gan maju cepat. Jangan gunakan gabungan squash. 10. Jika konflik terdeteksi, pengembang menerima peringatan dan menyelesa ikan konflik dengan permintaan gabungan.	

Pemecahan masalah

Isu	Solusi
Konflik cabang	Masalah umum yang dapat terjadi dengan model Gitflow adalah di mana perbaikan terbaru perlu terjadi dalam produksi tetapi

Isu	Solusi
	perubahan yang sesuai perlu terjadi di lingkungan yang lebih rendah, di mana cabang lain memodifikasi sumber daya yang sama. Kami menyarankan Anda hanya memiliki satu cabang rilis yang aktif pada satu waktu. Jika Anda memiliki lebih dari satu aktif pada satu waktu, perubahan lingkungan mungkin bertabrakan, dan Anda mungkin tidak dapat memindahkan cabang ke produksi.
Penggabungan	Rilis harus digabungkan kembali ke main dan dikembangkan sesegera mungkin untuk mengkonsolidasikan pekerjaan kembali ke cabang utama.
Penggabungan squash	Hanya gunakan gabungan squash saat Anda menggabungkan dari `feature` cabang ke cabang. `develop` Menggunakan penggabun gan squash di cabang yang lebih tinggi menyebabkan kesulitan saat menggabun gkan perubahan kembali ke cabang yang lebih rendah.

Sumber daya terkait

Panduan ini tidak termasuk pelatihan untuk Git; namun, ada banyak sumber daya berkualitas tinggi yang tersedia di internet jika Anda memerlukan pelatihan ini. Kami menyarankan Anda memulai dengan situs dokumentasi Git.

Sumber daya berikut dapat membantu Anda dengan perjalanan percabangan Gitflow Anda di. AWS Cloud

AWS DevOps bimbingan

- AWS DevOps Bimbingan
- AWS Arsitektur Referensi Pipeline Deployment

- Apa itu DevOps?
- DevOps sumber daya

Panduan Gitflow

- Blog Gitflow asli (posting blog Vincent Driessen)
- Alur kerja Gitflow (Atlassian)
- Gitflow on GitHub: Cara menggunakan alur kerja Git Flow dengan Repos GitHub Berbasis (video) YouTube
- Contoh Init Aliran Git (YouTube video)
- Cabang Rilis Gitflow dari Awal hingga Selesai (YouTube video)

Sumber daya lainnya

Metodologi aplikasi dua belas faktor (12factor.net)

Menerapkan strategi percabangan Trunk untuk lingkungan multi-akun DevOps

Mike Stephens dan Rayjan Wilson, Amazon Web Services

Ringkasan

Saat mengelola repositori kode sumber, strategi percabangan yang berbeda memengaruhi pengembangan perangkat lunak dan proses rilis yang digunakan tim pengembangan. Contoh strategi percabangan umum termasuk Trunk, GitHub Flow, dan Gitflow. Strategi-strategi ini menggunakan cabang yang berbeda, dan kegiatan yang dilakukan di setiap lingkungan berbeda. Organizations yang menerapkan DevOps proses akan mendapat manfaat dari panduan visual untuk membantu mereka memahami perbedaan antara strategi percabangan ini. Menggunakan visual ini di organisasi Anda membantu tim pengembangan menyelaraskan pekerjaan mereka dan mengikuti standar organisasi. Pola ini memberikan visual ini dan menjelaskan proses penerapan strategi percabangan Trunk di organisasi Anda.

Pola ini merupakan bagian dari rangkaian dokumentasi tentang memilih dan menerapkan strategi DevOps percabangan untuk organisasi dengan banyak Akun AWS. Seri ini dirancang untuk membantu Anda menerapkan strategi dan praktik terbaik yang benar sejak awal, untuk merampingkan pengalaman Anda di cloud. Trunk hanyalah salah satu strategi percabangan yang mungkin dapat digunakan organisasi Anda. Seri dokumentasi ini juga mencakup model percabangan GitHub Flow dan Gitflow. Jika Anda belum melakukannya, kami sarankan Anda meninjau Memilih strategi percabangan Git untuk DevOps lingkungan multi-akun sebelum menerapkan panduan dalam pola ini. Harap gunakan uji tuntas untuk memilih strategi percabangan yang tepat untuk organisasi Anda.

Panduan ini menyediakan diagram yang menunjukkan bagaimana organisasi dapat menerapkan strategi Trunk. Disarankan agar Anda meninjau Panduan AWS DevOps Well-Architected resmi untuk meninjau praktik terbaik. Pola ini mencakup tugas, langkah, dan batasan yang direkomendasikan untuk setiap langkah dalam DevOps proses.

Prasyarat dan batasan

Prasyarat

- Git, diinstal. Ini digunakan sebagai alat repositori kode sumber.
- Draw.io, diinstal. Aplikasi ini digunakan untuk melihat dan mengedit diagram.

Arsitektur

Arsitektur target

Diagram berikut dapat digunakan seperti kotak Punnett (Wikipedia). Anda menyusun cabang pada sumbu vertikal dengan AWS lingkungan pada sumbu horizontal untuk menentukan tindakan apa yang harus dilakukan dalam setiap skenario. Angka-angka menunjukkan urutan tindakan dalam alur kerja. Contoh ini membawa Anda dari `feature` cabang melalui penerapan dalam produksi.

Untuk informasi selengkapnya tentang Akun AWS, lingkungan, dan cabang dalam pendekatan Trunk, lihat Memilih strategi percabangan Git untuk lingkungan multi-akun DevOps .

Otomatisasi dan skala

Integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) is the process of automating the software release lifecycle. It automates much or all of the manual processes traditionally required to get new code from an initial commit into production. A CI/CD pipeline encompasses the sandbox, development, testing, staging, and production environments. In each environment, the CI/CD pipeline provisions any infrastructure that is needed to deploy or test the code. By using CI/CD, development teams can make changes to code that are then automatically tested and deployed. CI/CDjaringan pipa juga menyediakan tata kelola dan pagar pembatas untuk tim pengembangan dengan menegakkan konsistensi, standar, praktik terbaik, dan tingkat penerimaan minimal untuk penerimaan dan penerapan fitur. Untuk informasi selengkapnya, lihat Mempraktikkan Integrasi Berkelanjutan dan Pengiriman Berkelanjutan di AWS.

AWS menawarkan serangkaian layanan pengembang yang dirancang untuk membantu Anda membangun CI/CD jaringan pipa. Misalnya, AWS CodePipelineadalah layanan pengiriman berkelanjutan yang dikelola sepenuhnya yang membantu Anda mengotomatiskan saluran pipa rilis untuk pembaruan aplikasi dan infrastruktur yang cepat dan andal. AWS CodeBuildmengkompilasi kode sumber, menjalankan tes, dan menghasilkan paket ready-to-deploy perangkat lunak. Untuk informasi selengkapnya, lihat Alat Pengembang di AWS.

Alat

AWS layanan dan alat

AWS menyediakan serangkaian layanan pengembang yang dapat Anda gunakan untuk menerapkan pola ini:

- AWS CodeArtifactadalah layanan repositori artefak terkelola yang sangat skalabel yang membantu Anda menyimpan dan berbagi paket perangkat lunak untuk pengembangan aplikasi.
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeDeploymengotomatiskan penerapan ke Amazon Elastic Compute Cloud (Amazon EC2) atau instans, fungsi AWS Lambda , atau layanan Amazon Elastic Container Service (Amazon ECS) atau lokal.
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.

Alat-alat lainnya

- Draw.io Desktop — Aplikasi untuk membuat diagram alur dan diagram.
- Figma adalah alat desain online yang dirancang untuk kolaborasi. Repositori kode berisi template dalam format.fig untuk Figma.

Repositori kode

File sumber untuk diagram dalam pola ini tersedia di GitHub Git Branching Strategy untuk repositori Trunk. Ini termasuk file dalam format PNG, draw.io, dan Figma. Anda dapat memodifikasi diagram ini untuk mendukung proses organisasi Anda.

Praktik terbaik

Ikuti praktik dan rekomendasi terbaik dalam Panduan AWS DevOps Well-Architected dan Memilih strategi percabangan Git untuk lingkungan multi-akun. DevOps Ini membantu Anda menerapkan pengembangan berbasis Trunk secara efektif, mendorong kolaborasi, meningkatkan kualitas kode, dan merampingkan proses pengembangan.

Epik

Meninjau alur kerja Trunk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau proses Trunk standar.	1. Di lingkungan kotak pasir, pengembang membuat `feature` cabang dari `main` cabang dan menggunakan pola `feature/<ticket>_<initials>_<short description>` penamaan. 2. Pengembang mengembangkan kode dan menyebarkan kode ke lingkungan kotak pasir secara iteratif untuk menyelesaikan tiket. **Note** Pengembang secara opsional dapat membuat `sandbox` cabang untuk menjalankan build otomatis atau deploy pipeline di lingkungan kotak pasir. 3. Pengembang membuat permintaan gabungan dari `feature` cabang ke `main` cabang dengan menggunakan gabungan squash.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pipa integrasi berkelanjutan dan pengiriman berkelanj utan (CI/CD) secara otomatis membangun dan menerbitkan artefak dari `main` cabang ke lingkungan pengembangan. 5. Pemberi persetujuan secara manual menyetujui penyebaran artefak rilis ke lingkungan pengembangan. 6. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan pengujian. 7. Penyetuju secara manual menyetujui penerapan artefak rilis ke lingkungan pementasan. 8. Pemberi persetujuan secara manual menyetujui penyebaran artefak rilis ke lingkungan produksi.	

Pemecahan masalah

Isu	Solusi
Konflik cabang	Masalah umum yang dapat terjadi dengan model Trunk adalah di mana perbaikan terbaru perlu terjadi dalam produksi tetapi perubahan yang sesuai perlu terjadi di `feature` cabang, di mana sumber daya yang sama sedang

Isu	Solusi
	dimodifikasi. Kami menyarankan agar Anda sering menggabungkan perubahan dari cabang yang lebih rendah untuk menghindari konflik signifikan pada penggabungan `main` ke. `main`

Sumber daya terkait

Panduan ini tidak termasuk pelatihan untuk Git; namun, ada banyak sumber daya berkualitas tinggi yang tersedia di internet jika Anda memerlukan pelatihan ini. Kami menyarankan Anda memulai dengan situs dokumentasi Git.

Sumber daya berikut dapat membantu Anda dengan perjalanan percabangan Trunk Anda di. AWS Cloud

AWS DevOps bimbingan

- AWS DevOps Bimbingan
- Arsitektur Referensi Pipeline Deployment AWS
- Apa itu DevOps?
- DevOps sumber daya

Bimbingan batang

- Pengembangan Berbasis Batang

Sumber daya lainnya

- Metodologi aplikasi dua belas faktor (12factor.net)

Menerapkan pemindaian Checkov kustom terpusat untuk menegakkan kebijakan sebelum menerapkan infrastruktur AWS

Benjamin Morris, Amazon Web Services

Ringkasan

Pola ini menyediakan kerangka kerja GitHub Tindakan untuk menulis kebijakan Checkov kustom dalam satu repositori yang dapat digunakan kembali di seluruh organisasi. GitHub Dengan mengikuti pola ini, tim keamanan informasi dapat menulis, menambah, dan memelihara kebijakan khusus berdasarkan persyaratan perusahaan. Kebijakan kustom dapat ditarik ke semua pipeline dalam GitHub organisasi secara otomatis. Pendekatan ini dapat digunakan untuk menegakkan standar perusahaan untuk sumber daya sebelum sumber daya digunakan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- GitHub Organisasi yang menggunakan GitHub Tindakan
- AWS infrastruktur yang digunakan dengan HashiCorp Terraform atau AWS CloudFormation

Keterbatasan

- Pola ini ditulis untuk GitHub Tindakan. Namun, dapat disesuaikan dengan kerangka kerja integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang serupa seperti. GitLab Tidak diperlukan versi GitHub berbayar khusus.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Titik akhir layanan dan kuota dalam AWS dokumentasi, dan pilih tautan untuk layanan.

Arsitektur

Pola ini dirancang untuk digunakan sebagai GitHub repositori yang berisi alur kerja yang GitHub dapat digunakan kembali dan kebijakan Checkov kustom. Alur kerja yang dapat digunakan kembali dapat memindai Terraform dan CloudFormation infrastruktur sebagai repositori kode (IAc).

Diagram berikut menunjukkan repositori GitHub alur kerja Reusable dan repositori kebijakan Custom Checkov sebagai ikon terpisah. Namun, Anda dapat mengimplementasikan repositori ini baik sebagai

repositori terpisah atau repositori tunggal. Kode contoh menggunakan satu repositori, dengan file untuk alur kerja (`.github/workflows`) dan file untuk kebijakan khusus (`custom_policies`folder dan file `.checkov.yml` konfigurasi) di repositori yang sama.

Diagram menunjukkan alur kerja berikut:

1. Seorang pengguna membuat permintaan tarik dalam GitHub repositori.
2. Alur kerja pipeline dimulai di GitHub Actions, termasuk referensi ke alur kerja Checkov yang dapat digunakan kembali.
3. Alur kerja pipeline mengunduh alur kerja Checkov yang dapat digunakan kembali yang direferensikan dari repositori eksternal dan menjalankan alur kerja Checkov tersebut dengan menggunakan Tindakan. GitHub
4. Alur kerja Checkov yang dapat digunakan kembali mengunduh kebijakan kustom dari repositori eksternal.
5. Alur kerja Checkov yang dapat digunakan kembali mengevaluasi IAc di GitHub repositori terhadap kebijakan Checkov bawaan dan kustom. Alur kerja Checkov yang dapat digunakan kembali lolos atau gagal berdasarkan apakah masalah keamanan ditemukan.

Otomatisasi dan skala

Pola ini memungkinkan manajemen pusat konfigurasi Checkov, sehingga pembaruan kebijakan dapat diterapkan di satu lokasi. Namun, pola ini memang mengharuskan setiap repositori menggunakan alur kerja yang berisi referensi ke alur kerja pusat yang dapat digunakan kembali. Anda dapat menambahkan referensi ini secara manual atau menggunakan skrip untuk mendorong file ke `.github/workflows` folder untuk setiap repositori.

Alat

Layanan AWS

- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah. Checkov dapat memindai CloudFormation.

Alat lainnya

- Checkov adalah alat analisis kode statis yang memeriksa IAc untuk kesalahan konfigurasi keamanan dan kepatuhan.
- GitHub Tindakan diintegrasikan ke dalam GitHub platform untuk membantu Anda membuat, berbagi, dan menjalankan alur kerja dalam GitHub repositori Anda. Anda dapat menggunakan GitHub Tindakan untuk mengotomatiskan tugas seperti membangun, menguji, dan menerapkan kode Anda.
- Terraform adalah alat IAc HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal. Checkov dapat memindai Terraform.

Repositori kode

Kode untuk pola ini tersedia di GitHub centralized-custom-checkov-sastrepositori.

Praktik terbaik

- Untuk mempertahankan postur keamanan yang konsisten, selaraskan kebijakan keamanan perusahaan Anda dengan kebijakan Checkov.
- Pada fase awal penerapan kebijakan kustom Checkov, Anda dapat menggunakan opsi soft-fail dalam pemindaian Checkov Anda untuk memungkinkan IAc dengan masalah keamanan digabungkan. Saat proses matang, beralih dari opsi soft-fail ke opsi hard-fail.

Epik

Buat repositori Checkov pusat untuk kebijakan kustom

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Checkov pusat.	Buat repositori untuk menyimpan kebijakan Checkov kustom yang akan digunakan dalam organisasi. Untuk memulai dengan cepat, Anda dapat menyalin isi repositori pola ini ke GitHub centralized-custom-checkov-	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sast repositori Checkov pusat Anda.	
Buat repositori untuk alur kerja yang dapat digunakan kembali.	Jika repositori untuk alur kerja yang dapat digunakan kembali sudah ada, atau Anda berencana untuk menyertak an file alur kerja yang dapat digunakan kembali dalam repositori yang sama dengan kebijakan Checkov kustom, Anda dapat melewati langkah ini. Buat GitHub repositori untuk menyimpan alur kerja yang dapat digunakan kembali. Pipeline repositori lain akan mereferensikan repositori ini.	DevOps insinyur

Buat alur kerja Checkov yang dapat digunakan kembali dan contoh

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan alur kerja Checkov yang dapat digunakan kembali.	Buat alur kerja GitHub Tindakan Checkov (file YAMAL) yang dapat digunakan kembali di repositor i alur kerja yang dapat digunakan kembali. Anda dapat menyesuaikan alur kerja yang dapat digunakan kembali ini dari file alur kerja yang disediakan dalam pola ini.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Contoh perubahan yang mungkin ingin Anda lakukan adalah mengubah alur kerja yang dapat digunakan kembali untuk menggunakan opsi soft-fail. Pengaturan `soft-fail` untuk `true` memungkinkan pekerjaan selesai dengan sukses bahkan jika ada pemindaian Checkov yang gagal. Untuk instruksi, lihat Gagal keras dan lunak dalam dokumentasi Checkov.	
Tambahkan alur kerja contoh.	Tambahkan contoh alur kerja Checkov yang mereferensikan alur kerja. `reusable` Ini akan memberikan template untuk cara menggunakan kembali `reusable` alur kerja. Dalam contoh repositori, `checkov-source.yaml` adalah alur kerja yang dapat digunakan kembali dan `checkov-scan.yaml` merupakan contoh yang dikonsumsi. `checkov-source` Untuk detail selengkapnya tentang menulis contoh alur kerja Checkov, lihat Informasi tambahan.	DevOps insinyur

Mengaitkan kebijakan perusahaan dengan kebijakan kustom Checkov

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan kebijakan yang dapat ditegakkan dengan Checkov.	1. Tinjau kebijakan perusahaan yang terkait dengan keamanan infrastruktur dan persyaratan mana yang harus diberlakukan. 2. Tentukan persyaratan mana yang dapat diterapkan dengan menggunakan kebijakan kustom Checkov. 3. Buat konvensi penamaan yang memetakan kontrol kebijakan ke kebijakan kustom Checkov. Biasanya, kebijakan kustom Checkov memiliki pengenal dengan nama Checkov, sumber kebijakan (kustom), dan nomor kebijakan (misalnya ,). `CKV2_CUSTOM_123` Untuk detail selengkapnya tentang membuat kebijakan kustom Checkov, lihat Ikhtisar Kebijakan Kustom di dokumentasi Checkov.	Keamanan dan Kepatuhan
Tambahkan kebijakan kustom Checkov.	Ubah kebijakan perusahaan yang diidentifikasi menjadi kebijakan Checkov khusus di repositori pusat. Anda dapat menulis kebijakan Checkov	Keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sederhana baik dalam Python atau YAMAL.	

Menerapkan kebijakan kustom Checkov terpusat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan alur kerja Checkov yang dapat digunakan kembali ke semua repositori.	Pada titik ini, Anda harus memiliki contoh alur kerja Checkov yang mereferen sikan alur kerja yang dapat digunakan kembali. Salin contoh alur kerja Checkov yang mereferensikan alur kerja yang dapat digunakan kembali ke setiap repositori yang memerlukannya.	DevOps insinyur
Buat mekanisme untuk memastikan bahwa Checkov berjalan sebelum bergabung.	Untuk memastikan bahwa alur kerja Checkov dijalanka n untuk setiap permintaa n tarik, buat pemeriksaan status yang memerlukan alur kerja Checkov yang berhasil sebelum permintaa n tarik dapat digabungk an. GitHub memungkinkan Anda mengharuskan alur kerja tertentu dijalankan sebelum permintaan tarik dapat digabungkan.	DevOps insinyur
Buat PAT di seluruh organisas i, dan bagikan sebagai rahasia.	Jika GitHub organisasi Anda terlihat publik, Anda dapat melewati langkah ini.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pola ini mengharuskan alur kerja Checkov dapat mengunduh kebijakan khusus dari repositori kebijakan khusus di organisasi Anda. GitHub Anda harus memberikan izin sedemikian rupa sehingga alur kerja Checkov dapat mengakses repositori tersebut. Untuk melakukan ini, buat token akses pribadi (PAT) dengan izin untuk membaca repositori organisasi. Bagikan PAT ini dengan repositori, baik sebagai rahasia seluruh organisasi (jika pada paket berbayar) atau rahasia di setiap repositori (versi gratis). Dalam kode sampel, nama default untuk rahasia adalah `ORG_PAT`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Lindungi file alur kerja Checkov dari modifikasi.	Untuk melindungi file alur kerja Checkov dari perubahan yang tidak diinginkan, Anda dapat menggunakan file. `CODEOWNERS` `CODEOWNERS` File ini biasanya digunakan di root direktori. Misalnya, untuk meminta persetujuan dari `secEng` grup GitHub organisasi Anda saat `checkov-scan.yaml` file diubah, tambahkan yang berikut ini ke file repositori: `CODEOWNERS` `[Checkov] .github/workflows /checkov-scan.yaml @myOrg/secEng` `CODEOWNERS` File khusus untuk repositori tempat tinggalnya. Untuk melindungi alur kerja Checkov yang digunakan oleh repositori, Anda harus menambahkan (atau memperbarui) `CODEOWNERS` file di setiap repositori. Untuk informasi selengkapnya tentang melindungi file alur kerja Checkov, lihat Informasi tambahan. Untuk informasi selengkapnya	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tentang `CODEOWNERS` file, lihat dokumentasi resmi untuk CI/CD penyedia Anda (seperti GitHub).	

Sumber daya terkait

- Ikhtisar Kebijakan Kustom Checkov
- CloudFormation Pemindaian Konfigurasi
- GitHub Tindakan Alur Kerja yang Dapat Digunakan Kembali

Informasi tambahan

Menulis file alur kerja Checkov

Saat menulis`checkov-scan.yaml`, pertimbangkan kapan Anda ingin menjalankannya. on Kunci tingkat atas menentukan kapan alur kerja berjalan. Dalam repositori contoh, alur kerja akan berjalan ketika ada permintaan tarik yang menargetkan `main` cabang (dan kapan saja cabang sumber permintaan tarik diubah). Alur kerja juga dapat dijalankan sesuai kebutuhan karena `workflow_dispatch` kuncinya.

Anda dapat mengubah kondisi pemicu alur kerja berdasarkan seberapa sering Anda ingin alur kerja dijalankan. Misalnya, Anda dapat mengubah alur kerja untuk dijalankan setiap kali kode didorong ke cabang mana pun dengan mengganti `pull_request` dengan `push` dan menghapus `branches` kunci.

Anda dapat memodifikasi contoh file alur kerja yang Anda buat dalam repositori individual. Misalnya, Anda dapat menyesuaikan nama cabang target dari `main` `production` jika repositori terstruktur di sekitar cabang. `production`

Melindungi file alur kerja Checkov

Pemindaian Checkov memberikan informasi yang berguna tentang kemungkinan kesalahan konfigurasi keamanan. Namun, beberapa pengembang mungkin menganggapnya sebagai penghalang produktivitas mereka dan mencoba untuk menghapus atau menonaktifkan alur kerja pemindaian.

Ada beberapa cara untuk mengatasi masalah ini, termasuk pesan yang lebih baik tentang nilai jangka panjang dari pemindaian keamanan dan dokumentasi yang lebih jelas tentang cara menerapkan infrastruktur yang aman. Ini adalah pendekatan “lunak” penting untuk DevSecOps kolaborasi yang dapat dilihat sebagai solusi untuk akar masalah ini. Namun, Anda juga dapat menggunakan kontrol teknis seperti `CODEOWNERS` file sebagai pagar pembatas untuk membantu pengembang tetap berada di jalur yang benar.

Pola pengujian di kotak pasir

Untuk menguji pola ini di lingkungan kotak pasir, ikuti langkah-langkah berikut:

1. Buat GitHub organisasi baru. Buat token dengan akses hanya-baca ke semua repositori di organisasi. Karena token ini untuk lingkungan kotak pasir, bukan lingkungan berbayar, Anda tidak akan dapat menyimpan token ini dalam rahasia seluruh organisasi.
2. Buat `checkov` repositori untuk menyimpan konfigurasi Checkov dan `github-workflows` repositori untuk menyimpan konfigurasi alur kerja yang dapat digunakan kembali. Isi repositori dengan isi repositori contoh.
3. Buat repositori aplikasi, dan salin dan tempel `checkov-scan.yaml` alur kerja ke foldernya. `.github/workflows` Tambahkan rahasia ke repositori yang berisi PAT yang Anda buat untuk akses hanya-baca organisasi. Rahasia default adalah`ORG_PAT`.
4. Buat permintaan tarik yang menambahkan beberapa Terraform atau CloudFormation kode ke repositori aplikasi. Checkov harus memindai dan mengembalikan hasilnya.

Menerapkan diagnostik Kubernetes yang didukung AI dan pemecahan masalah dengan integrasi K8SGPT dan Amazon Bedrock

Ishwar Chauthaiwale, Muskan., dan Prafful Gupta, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menerapkan diagnostik dan pemecahan masalah Kubernetes yang didukung AI dengan mengintegrasikan K8SGPT dengan model Anthropic Claude v2 yang tersedia di Amazon Bedrock. Solusi ini menyediakan analisis bahasa alami dan langkah-langkah remediasi untuk masalah klaster Kubernetes melalui arsitektur host bastion yang aman. Dengan menggabungkan keahlian K8SGPT Kubernetes dengan kemampuan bahasa canggih Amazon Bedrock, tim dapat dengan cepat mengidentifikasi dan menyelesaikan masalah DevOps klaster. Dengan kemampuan ini, dimungkinkan untuk mengurangi mean time to resolution (MTTR) hingga 50 persen.

Pola cloud-native ini memanfaatkan Amazon Elastic Kubernetes Service (Amazon EKS) untuk manajemen Kubernetes. Pola ini menerapkan praktik terbaik keamanan melalui peran yang tepat AWS Identity and Access Management (IAM) dan isolasi jaringan. Solusi ini sangat berharga bagi organisasi yang ingin merampingkan operasi Kubernetes mereka dan meningkatkan kemampuan pemecahan masalah mereka dengan bantuan AI.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan izin yang sesuai
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi
- Klaster Amazon EKS
- Akses ke model Anthropic Claude 2 di Amazon Bedrock
- Host bastion dengan pengaturan grup keamanan yang diperlukan
- K8sgpt diinstal

Batasan

- Analisis K8sgpt dibatasi oleh ukuran jendela konteks model Claude v2.
- Batas tarif Amazon Bedrock API berlaku berdasarkan kuota akun Anda.

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Amazon EKS versi 1.31 atau yang lebih baru
- Model Claude 2 di Amazon Bedrock
- K8sgpt v0.4.2 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan arsitektur untuk diagnostik Kubernetes bertenaga AI menggunakan K8SGPT yang terintegrasi dengan Amazon Bedrock di. AWS Cloud

Arsitektur menunjukkan alur kerja berikut:

1. Pengembang mengakses lingkungan melalui koneksi aman ke host bastion. EC2 Instans Amazon ini berfungsi sebagai titik masuk aman dan berisi instalasi antarmuka baris perintah (CLI) K8sgpt dan konfigurasi yang diperlukan.
2. Host bastion, yang dikonfigurasi dengan peran IAM tertentu, membuat koneksi aman ke cluster Amazon EKS dan titik akhir Amazon Bedrock. K8sgpt diinstal dan dikonfigurasi pada host bastion untuk melakukan analisis klaster Kubernetes.
3. Amazon EKS mengelola bidang kontrol Kubernetes dan node pekerja, menyediakan lingkungan target untuk analisis K8sgpt. Layanan ini berjalan di beberapa Availability Zone dalam virtual private cloud (VPC), yang membantu menyediakan ketersediaan dan ketahanan yang tinggi. Amazon EKS memasok data operasional melalui Kubernetes API, memungkinkan analisis klaster yang komprehensif.
4. K8sgpt mengirimkan data analisis ke Amazon Bedrock, yang menyediakan model dasar Claude v2 (FM) untuk pemrosesan bahasa alami. Layanan ini memproses analisis K8sgpt untuk menghasilkan penjelasan yang dapat dibaca manusia dan menawarkan saran remediasi terperinci berdasarkan masalah yang diidentifikasi. Amazon Bedrock beroperasi sebagai layanan AI tanpa server dengan ketersediaan dan skalabilitas tinggi.

> **Note**
>
> Sepanjang alur kerja ini, IAM mengontrol akses antar komponen melalui peran dan kebijakan, mengelola otentikasi untuk host bastion, Amazon EKS, dan interaksi Amazon Bedrock. IAM menerapkan prinsip hak istimewa paling sedikit dan memungkinkan komunikasi lintas layanan yang aman di seluruh arsitektur.

Otomatisasi dan skala

Operasi K8SGPT dapat diotomatisasi dan diskalakan di beberapa kluster Amazon EKS melalui berbagai alat dan. Layanan AWS Solusi ini mendukung integrasi berkelanjutan dan integrasi penerapan berkelanjutan (CI/CD) menggunakan Jenkins, GitHub Actions, atau AWS CodeBuilduntuk analisis terjadwal. Operator K8sgpt memungkinkan pemantauan in-cluster berkelanjutan dengan kemampuan deteksi dan pelaporan masalah otomatis. Untuk penerapan skala perusahaan, Anda dapat menggunakan Amazon EventBridge untuk menjadwalkan pemindaian dan memicu respons otomatis dengan skrip khusus. AWS Integrasi SDK memungkinkan kontrol terprogram di seluruh armada cluster yang besar.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Alat lainnya

- K8SGPT adalah alat bertenaga AI open source yang mengubah manajemen Kubernetes. Ini bertindak sebagai ahli rekayasa keandalan situs virtual (SRE), secara otomatis memindai, mendiagnosis, dan memecahkan masalah klaster Kubernetes. Administrator dapat berinteraksi dengan K8sgpt menggunakan bahasa alami dan mendapatkan wawasan yang jelas dan dapat

ditindaklanjuti tentang status cluster, pod crash, dan kegagalan layanan. Alat analisis bawaan alat mendeteksi berbagai masalah, mulai dari komponen yang salah konfigurasi hingga kendala sumber daya, dan memberikan penjelasan dan solusi. easy-to-understand

Praktik terbaik

- Menerapkan kontrol akses aman dengan menggunakan AWS Systems Manager Session Manager untuk akses host bastion.
- Pastikan autentikasi K8sgpt menggunakan peran IAM khusus dengan izin hak istimewa paling sedikit untuk interaksi Amazon Bedrock dan Amazon EKS. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.
- Konfigurasikan penandaan sumber daya, aktifkan CloudWatch pencatatan Amazon untuk jejak audit, dan terapkan anonimisasi data untuk informasi sensitif.
- Pertahankan pencadangan reguler konfigurasi K8SGPT sambil menyiapkan jadwal pemindaian otomatis selama jam sibuk untuk meminimalkan dampak operasional.

Epik

Tambahkan Amazon Bedrock ke daftar penyedia backend AI.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan Amazon Bedrock sebagai penyedia backend AI untuk K8sgpt.	Untuk menyetel Amazon Bedrock sebagai backend AI menyediakan r untuk K8sgpt, gunakan perintah berikut: AWS CLI `k8sgpt auth add -b amazonbedrock \ -r us-west-2 \ -m anthropic.claude-v2 \ -n endpoint-name` Contoh perintah digunakan `us-west-2` untuk Wilayah	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	AWS. Namun, Anda dapat memilih Wilayah lain, asalkan kluster Amazon EKS dan model Amazon Bedrock yang sesuai tersedia dan diaktifkan di Wilayah yang dipilih tersebut.
	
	Untuk memeriksa `amazonbedrock` apakah ditambahkan ke daftar penyedia backend AI dan dalam `Active` status, jalankan perintah berikut:
	
	`k8sgpt auth list`
	
	Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	
	`Default:`
	`> openai`
	`Active:`
	`> amazonbedrock`
	`Unused:`
	`> openai`
	`> localai`
	`> ollama`
	`> azureopenai`
	`> cohere`
	`> amazonsagemaker`
	`> google`
	`> noopai`
	`> huggingface`
	`> googlevertexai`
	`> oci`
	`> customrest`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`> ibmwatsonxai`	

Pindai sumber daya menggunakan filter

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lihat daftar filter yang tersedia.	Untuk melihat daftar semua filter yang tersedia, gunakan AWS CLI perintah berikut:
	`k8sgpt filters list`
	Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	`Active:`
	`> Deployment`
	`> ReplicaSet`
	`> PersistentVolumeCl aim`
	`> Service`
	`> CronJob`
	`> Node`
	`> MutatingWebhookCon figuration`
	`> Pod`
	`> Ingress`
	`> StatefulSet`
	`> ValidatingWebhookC onfiguration`
		AWS DevOps

Pindai pod di namespace tertentu dengan menggunakan filter.	Perintah ini berguna untuk debugging tertarget dari masalah pod tertentu dalam klaster Kubernetes,	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	menggunakan kemampuan Amazon Bedrock AI untuk menganalisis dan menjelaskan masalah yang ditemukannya.
	
	Untuk memindai pod di namespace tertentu dengan menggunakan filter, gunakan perintah berikut: AWS CLI
	
	`k8sgpt analyze --backend amazonbedrock --explain --filter Pod -n default`
	
	Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	
	`100% |################################################| (1/1, 645 it/s)`
	
	`AI Provider: amazonbedrock`
	
	`0: Pod default/crashme()`
	`- Error: the last termination reason is Error container=crashme pod=crashme`
	`Error: The pod named crashme terminated because the container named crashme crashed.`
	`Solution: Check logs for crashme pod to`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`identify reason for crash. Restart pod or redeploy application to resolve crash.`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Pindai penyebaran di namespace tertentu dengan menggunakan filter.	Perintah ini berguna untuk mengidentifikasi dan memecahkan masalah khusus penerapan, terutama ketika status sebenarnya tidak cocok dengan status yang diinginkan.
	
	Untuk memindai penyebaran di namespace tertentu dengan menggunakan filter, gunakan perintah berikut: AWS CLI
	
	`k8sgpt analyze --backend amazonbedrock --explain --filter Deployment -n default`
	
	Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	
	`100% |########## ########## ########## #####| (1/1, 10 it/min)`
	
	`AI Provider: amazonbedrock`
	
	`0: Deployment default/nginx()`
	`- Error: Deployment default/nginx has 1 replicas but 2 are available`
	`Error: The Deployment named nginx in the`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`default namespace has 1 replica specified but 2 pod replicas are running.` `Solution: Check if any other controlle rs like ReplicaSet or StatefulSet have created extra pods. Delete extra pods or adjust replica count to match available pods.`	

Pindai node di namespace tertentu dengan menggunakan filter.	Untuk memindai node di namespace tertentu dengan menggunakan filter, gunakan perintah berikut: AWS CLI
	`k8sgpt analyze --backend amazonbedrock --explain --filter Node -n default`
	Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	`AI Provider: amazonbedrock`
	
	`No problems detected`
		AWS DevOps

Menganalisis output terperinci

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Dapatkan output terperinci.	Untuk mendapatkan output terperinci, gunakan AWS CLI perintah berikut:
	`k8sgpt analyze --backend amazonbedrock --explain --ouput json`
	Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	`{ "provider": "amazonbedrock", "errors": null, "status": "ProblemDetected", "problems": 1, "results": [ { "kind": "Pod", "name": "default/crashme", "error": [ { "Text": "the last termination reason is Error container=crashme pod=crashme", "KubernetesDoc": "", "Sensitive": [] } ],`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"details": " Error: The pod named crashme terminated because the container named crashme crashed. \nSolution: Check logs for crashme pod to identify reason for crash. Restart pod or redeploy application to resolve crash.", "parentObject": "" }] }</pre>	

Periksa pod yang bermasalah.	Untuk memeriksa pod bermasalah tertentu, gunakan AWS CLI perintah berikut:
	<pre>kubectl get pods --all-namespaces | grep -v Running</pre>Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	<pre>NAMESPACE NAME READY STATUS RESTARTS AGE
	
	default crashme 0/1 CrashLoopBackOff 260(91s ago) 21h</pre>
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Dapatkan wawasan khusus aplikasi.	Perintah ini sangat berguna ketika:
	• Anda ingin fokus pada aplikasi tertentu di cluster Anda.
	• Anda menggunakan label secara efektif untuk mengatur sumber daya Kubernetes Anda.
	• Anda perlu dengan cepat memeriksa kesehatan komponen aplikasi tertentu.
	
	Untuk mendapatkan wawasan khusus aplikasi, gunakan perintah berikut:
	`k8sgpt analyze --backend amazonbedrock --explain -L app=nginx -n default`
	
	Berikut ini adalah contoh output yang diharapkan dari perintah ini:
	`AI Provider: amazonbedrock`
	
	`No problems detected`	

Sumber daya terkait

Blog AWS

- Mengotomatiskan pemecahan masalah Amazon EKS menggunakan alur kerja agen Amazon Bedrock
- Gunakan K8SGPT dan Amazon Bedrock untuk pemeliharaan klaster Kubernetes yang disederhanakan

AWS dokumentasi

- AWS CLI perintah: buat-cluster dan deskripsikan-cluster
- Memulai Amazon EKS (dokumentasi Amazon EKS)
- Praktik terbaik keamanan dalam IAM (dokumentasi IAM)

Sumber daya lainnya

- K8sgpt

Secara otomatis mendeteksi perubahan dan memulai CodePipeline jaringan pipa yang berbeda untuk monorepo di CodeCommit

Helton Ribeiro, Petrus Batalha, dan Ricardo Morais, Amazon Web Services

Ringkasan

Pemberitahuan: AWS Cloud9 tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS Cloud9 dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

Pola ini membantu Anda secara otomatis mendeteksi perubahan pada kode sumber aplikasi berbasis monorepo AWS CodeCommit dan kemudian memulai pipeline yang menjalankan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) automation for each microservice. This approach means that each microservice in your monorepo-based application can have a dedicated CI/CDpipeline, AWS CodePipeline yang memastikan visibilitas yang lebih baik, berbagi kode yang lebih mudah, dan peningkatan kolaborasi, standardisasi, dan kemampuan ditemukan.

Solusi yang dijelaskan dalam pola ini tidak melakukan analisis ketergantungan apa pun di antara layanan mikro di dalam monorepo. Ini hanya mendeteksi perubahan dalam kode sumber dan memulai pipa yang cocok CI/CD .

Pola ini digunakan AWS Cloud9 sebagai lingkungan pengembangan terintegrasi (IDE) dan AWS Cloud Development Kit (AWS CDK) untuk mendefinisikan infrastruktur dengan menggunakan dua CloudFormation tumpukan: `MonoRepoStack` dan`PipelinesStack`. `MonoRepoStack`Tumpukan menciptakan monorepo AWS CodeCommit dan AWS Lambda fungsi yang memulai pipeline. CI/CD `PipelinesStack`Tumpukan mendefinisikan infrastruktur pipeline Anda.

> **Important**
>
> Alur kerja pola ini adalah bukti konsep (PoC). Kami menyarankan Anda menggunakannya hanya di lingkungan pengujian. Jika Anda ingin menggunakan pendekatan pola ini di lingkungan produksi, lihat Praktik terbaik keamanan di IAM dalam dokumentasi AWS Identity and Access Management (IAM) dan buat perubahan yang diperlukan pada peran IAM Anda dan. Layanan AWS

Prasyarat dan batasan

Prasyarat

- AWS Akun aktif.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal, memperbarui, dan menghapus instalan AWS CLI dalam AWS CLI dokumentasi.
- Python 3 dan`pip`, diinstal pada mesin lokal Anda. Untuk informasi selengkapnya, lihat dokumentasi Python.
- AWS CDK, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat AWS CDK Memulai dengan AWS CDK dokumentasi.
- AWS Cloud9 IDE, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menyiapkan AWS Cloud9 dalam AWS Cloud9 dokumentasi.
- GitHub AWS CodeCommit Monorepo multi-pipeline memicu repositori, dikloning pada mesin lokal Anda.
- Direktori yang ada berisi kode aplikasi yang ingin Anda buat dan terapkan CodePipeline.
- Keakraban dan pengalaman dengan praktik DevOps terbaik di. AWS Cloud Untuk meningkatkan keakraban Anda DevOps, Anda dapat menggunakan pola Bangun arsitektur yang digabungkan secara longgar dengan layanan mikro menggunakan DevOps praktik dan AWS Cloud9 di situs web Prescriptive Guidance. AWS

Arsitektur

Diagram berikut menunjukkan bagaimana menggunakan AWS CDK untuk mendefinisikan infrastruktur dengan dua AWS CloudFormation tumpukan: `MonoRepoStack` dan`PipelinesStack`.

Diagram menunjukkan alur kerja berikut:

1. Proses bootstrap menggunakan AWS CDK untuk membuat AWS CloudFormation tumpukan `MonoRepoStack` dan`PipelinesStack`.
2. `MonoRepoStack`Tumpukan membuat CodeCommit repositori untuk aplikasi Anda dan fungsi `monorepo-event-handler` Lambda yang dimulai setelah setiap komit.
3. `PipelinesStack`Tumpukan membuat pipeline yang diprakarsai oleh fungsi Lambda. CodePipeline Setiap layanan mikro harus memiliki pipa infrastruktur yang ditentukan.
4. Pipeline untuk `microservice-n` diprakarsai oleh fungsi Lambda dan memulai tahapan CI/CD terisolasi yang didasarkan pada kode sumber di. CodeCommit

5. Pipeline untuk `microservice-1` diprakarsai oleh fungsi Lambda dan memulai tahapan CI/CD terisolasi yang didasarkan pada kode sumber di. CodeCommit

Diagram berikut menunjukkan penyebaran AWS CloudFormation tumpukan `MonoRepoStack` dan `PipelinesStack` dalam akun.

1. Seorang pengguna mengubah kode di salah satu layanan mikro aplikasi.
2. Pengguna mendorong perubahan dari repositori lokal ke repositori. CodeCommit
3. Aktivitas push memulai fungsi Lambda yang menerima semua dorongan ke repositori. CodeCommit
4. Fungsi Lambda membaca parameter di Parameter Store, kemampuan AWS Systems Manager, untuk mengambil ID komit terbaru. Parameter memiliki format penamaan:`/MonoRepoTrigger/{repository}/{branch_name}/LastCommit`. Jika parameter tidak ditemukan, fungsi Lambda membaca ID komit terakhir dari CodeCommit repositori dan menyimpan nilai yang dikembalikan di Parameter Store.
5. Setelah mengidentifikasi ID komit dan file yang diubah, fungsi Lambda mengidentifikasi pipeline untuk setiap direktori microservice dan memulai pipeline yang diperlukan. CodePipeline

Alat

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak untuk mendefinisikan infrastruktur cloud dalam kode dan menyediakannya. CloudFormation
- Python adalah bahasa pemrograman yang memungkinkan Anda bekerja dengan cepat dan mengintegrasikan sistem dengan lebih efektif.

Kode

Kode sumber dan template untuk pola ini tersedia di repositori pemicu multi-pipeline GitHub AWS CodeCommit monorepo.

Praktik terbaik

- Contoh arsitektur ini tidak menyertakan solusi pemantauan untuk infrastruktur yang dikerahkan. Jika Anda ingin menerapkan solusi ini di lingkungan produksi, kami sarankan Anda mengaktifkan pemantauan. Untuk informasi selengkapnya, lihat Memantau aplikasi tanpa server Anda dengan

CloudWatch Application Insights dalam dokumentasi AWS Serverless Application Model ()AWS SAM.

- Saat Anda mengedit kode sampel yang disediakan oleh pola ini, ikuti praktik terbaik untuk mengembangkan dan menerapkan infrastruktur cloud dalam AWS CDK dokumentasi.
- Saat Anda menentukan saluran pipa layanan mikro, tinjau praktik terbaik keamanan dalam dokumentasi. AWS CodePipeline
- Anda juga dapat memeriksa AWS CDK kode Anda untuk praktik terbaik dengan menggunakan utilitas cdk-nag. Alat ini menggunakan seperangkat aturan, dikelompokkan berdasarkan paket, untuk mengevaluasi kode Anda. Paket yang tersedia adalah:
  - AWS Perpustakaan Solusi
  - Keamanan Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA)
  - Institut Nasional Standar dan Teknologi (NIST) 800-53 rev 4
  - NIST 800-53 rev 5
  - Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS) 3.2.1

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat lingkungan Python virtual.	Di AWS Cloud9 IDE Anda, buat lingkungan Python virtual dan instal dependensi yang diperlukan dengan menjalankan perintah berikut: `make install`	Developer
Bootstrap Akun AWS dan Wilayah AWS untuk AWS CDK.	Bootstrap required Akun AWS dan Region dengan menjalankan perintah berikut: `make bootstrap account-id=<your-AWS-account-ID>`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`region=<required-region>`	

Tambahkan pipeline baru untuk microservice

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan kode sampel Anda ke direktori aplikasi Anda.	Tambahkan direktori yang berisi kode aplikasi sampel Anda ke `monorepo-sample` direktori di repositori pemicu multi-pipeline GitHub AWS CodeCommit monorepo kloning.	Developer
Edit file `monorepo-main.json`	Tambahkan nama direktori kode aplikasi Anda dan nama pipeline ke `monorepo-main.json` file di repositori kloning.	Developer
Buat pipa.	Dalam `Pipelines` direktori untuk repositori, tambahkan pipeline `class` untuk aplikasi Anda. Direktori berisi dua file sampel, `pipeline_hotsite.py` dan`pipeline_demo.py` . Setiap file memiliki tiga tahap: source, build, dan deploy. Anda dapat menyalin salah satu file dan membuat perubahan sesuai dengan kebutuhan aplikasi Anda.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Edit file `monorepo_config.py`	Di `service_map`, tambahkan nama direktori untuk aplikasi Anda dan kelas yang Anda buat untuk pipeline.
	
	Misalnya, kode berikut menunjukkan definisi pipeline dalam `Pipelines` direktori yang menggunakan file bernama `pipeline_mysample.py` dengan `MySamplePipeline` kelas:
	
	`...`
	`# Pipeline definition imports`
	`from pipelines.pipeline_demo import DemoPipeline`
	`from pipelines.pipeline_hotsite import HotsitePipeline`
	`from pipelines.pipeline_mysample import MySamplePipeline`
	
	`### Add your pipeline configuration here`
	`service_map: Dict[str, ServicePipeline] = {`
	`# folder-name -> pipeline-class`
	`'demo': DemoPipeline(),`
	`'hotsite': HotsitePipeline(),`
	`'mysample': MySamplePipeline()`
		Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`}`	

Menyebarkan tumpukan MonoRepoStack

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan AWS CloudFormation tumpukan.	Terapkan AWS CloudFormation `MonoRepoStack` tumpukan dengan nilai parameter default di direktori root dari repositori kloning dengan menjalankan perintah. `make deploy-core` Anda dapat mengubah nama repositori dengan menjalankan perintah. `make deploy-core monorepo-name=<repo_name>` **Note** Anda dapat secara bersamaan menyebarkan kedua pipeline dengan menggunakan perintah. `make deploy monorepo-name=<repo_name>`	Developer
Validasi CodeCommit repositori.	Validasi bahwa sumber daya Anda dibuat	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan menjalankan `aws codecommit get-repository --repository-name <repo_name>` perintah. **Important** Karena CloudFormation tumpukan membuat CodeCommit repositori tempat monorepo disimpan, jangan jalankan `cdk destroy MonoRepoStack` perintah jika Anda sudah mulai mendorong modifikasi ke dalamnya.	
Validasi hasil CloudFormation tumpukan.	Validasi bahwa CloudFormation `MonoRepoStack` tumpukan dibuat dan dikonfigurasi dengan benar dengan menjalankan perintah berikut: `aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --query 'StackSummaries[?StackName == 'MonoRepoStack']'`	Developer

Menyebarkan tumpukan PipelinesStack

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation tumpukan.	AWS CloudFormation `PipelinesStack` Tumpukan harus digunakan setelah Anda menyebarkan tumpukan. `MonoRepoStack` Tumpukan bertambah besar ketika layanan mikro baru ditambahkan ke basis kode monorepo dan digunakan kembali saat layanan mikro baru di-onboard. Menyebarkan Pipelines Stack tumpukan dengan menjalankan `make deploy-pipelines` perintah. **Note** Anda juga dapat menyebarkan secara bersamaan menyebarkan kedua pipeline dengan menjalankan perintah. `make deploy monorepo-name=<repo_name>`	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Contoh keluaran berikut menunjukkan bagaimana `PipelinesStacks` penerapan mencetak URLs untuk layanan mikro di akhir implementasi: `Outputs: PipelinesStack.dem ourl = .cloudfront.net PipelinesStack.hotsi teurl = .cloudfro nt.net`	
Validasi hasil AWS CloudFormation tumpukan.	Validasi bahwa AWS CloudFormation `Pipelines Stacks` tumpukan dibuat dan dikonfigurasi dengan benar dengan menjalankan perintah berikut: `aws cloudformation list-stacks --stack-s tatus-filter CREATE_CO MPLETE UPDATE_COMPLETE --query 'StackSum maries[?StackName == 'PipelinesStack']'`	Developer

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus AWS CloudFormation tumpukan Anda.	Jalankan perintah `make destroy`.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus bucket S3 untuk pipeline Anda.	1. Masuk ke Konsol Manajemen AWSdan buka konsol Amazon Simple Storage Service (Amazon S3). 2. Hapus bucket S3 yang terkait dengan pipeline Anda dan gunakan nama berikut: `pipelines stack-codepipeline *`	Developer

Pemecahan masalah

Isu	Solusi
Saya mengalami AWS CDK masalah.	Lihat Memecahkan AWS CDK masalah umum dalam dokumentasi AWS CDK.
Saya mendorong kode layanan mikro saya, tetapi pipa layanan mikro tidak berjalan.	Validasi pengaturan Verifikasi konfigurasi cabang: • Pastikan Anda mendorong kode Anda ke cabang yang benar. Pipeline ini dikonfigu rasi untuk berjalan hanya ketika perubahan dilakukan ke `main` cabang. Dorongan ke cabang lain tidak memulai pipeline kecuali jika dikonfigurasi secara khusus. • Setelah Anda mendorong kode Anda, periksa apakah komit terlihat AWS CodeCommi t untuk memastikan bahwa push berhasil dan bahwa koneksi antara lingkungan lokal Anda dan repositori masih utuh. Segarkan

Isu	Solusi
	kredensyal Anda jika ada masalah dalam mendorong kode. Validasi file konfigurasi: • Konfirmasikan bahwa `service_map` variabel secara `monorepo_config.py` akurat mencerminkan struktur direktori layanan mikro Anda saat ini. Variabel ini memainkan peran penting dalam memetakan push kode Anda ke pipeline masing-masing. • Pastikan `monorepo-main.json` itu diperbarui untuk menyertakan pemetaan baru untuk layanan mikro Anda. File ini penting agar pipeline mengenali dan menangani perubahan pada layanan mikro Anda dengan benar. Pemecahan masalah di konsol AWS CodePipeline cek: • Pada Konsol Manajemen AWS, konfirmas ikan bahwa Anda berada di Wilayah AWS tempat pipeline Anda di-host. Buka CodePipeline konsol dan periksa apakah pipeline yang sesuai dengan layanan mikro Anda telah dimulai. Analisis kesalahan: Jika pipeline dimulai tetapi gagal, tinjau pesan kesalahan atau log apa pun yang disediakan oleh CodePipeline untuk memahami apa yang salah. AWS Lambda pemecahan masalah:

Isu	Solusi
	• Di AWS Lambda konsol, buka fungsi `monorepo-event-handler` Lambda. Verifikasi bahwa fungsi dimulai sebagai respons terhadap push kode. Analisis log: Periksa log fungsi Lambda untuk masalah apa pun. Log dapat memberikan wawasan terperinci tentang apa yang terjadi saat fungsi dijalankan dan membantu mengidentifikasi apakah fungsi memproses peristiwa seperti yang diharapkan.

Isu	Solusi
Saya perlu memindahkan semua layanan mikro saya.	Ada dua pendekatan untuk memaksa pemindahan semua layanan mikro. Pilih opsi yang sesuai dengan kebutuhan Anda. Pendekatan 1: Hapus parameter di Parameter Store Metode ini melibatkan penghapusan parameter tertentu dalam Systems Manager Parameter Store yang melacak ID komit terakhir yang digunakan untuk penerapan. Saat Anda menghapus parameter ini, sistem dipaksa untuk menerapkan kembali semua layanan mikro pada pemicu berikutnya, karena ia menganggapnya sebagai keadaan baru. Langkah: 1. Temukan entri Parameter Store tertentu yang menyimpan ID komit atau penanda penerapan terkait untuk monorepo Anda. Nama parameter mengikuti format: `"/MonoRepoTrigger/{repository}/{branch_name}/LastCommit"` 2. Pertimbangkan untuk mencadangkan nilai parameter jika penting atau jika Anda ingin mempertahankan catatan status penerapan sebelum mengatur ulang. 3. Gunakan Konsol Manajemen AWS, AWS CLI, atau SDKs untuk menghapus parameter yang diidentifikasi. Tindakan ini mengatur ulang penanda penerapan. 4. Setelah penghapusan, dorongan berikutny a ke repositori akan menyebabkan sistem menerapkan semua layanan mikro, karena

Isu	Solusi
	mencari komit terbaru untuk dipertimb angkan untuk penerapan. Kelebihan: • Sederhana dan cepat diimplementasikan dengan langkah-langkah minimal. • Tidak perlu membuat perubahan kode arbitrer untuk memulai penerapan. Kontra: • Kurang kontrol granular atas proses penyebaran. • Berpotensi berisiko jika Parameter Store digunakan untuk mengelola konfigurasi penting lainnya. Pendekatan 2: Dorong komit di setiap subfolder monorepo Metode ini melibatkan membuat perubahan kecil dan mendorongnya di setiap subfolder layanan mikro di dalam monorepo untuk memulai jaringan pipa masing-masing. Langkah: 1. Buat daftar semua layanan mikro dalam monorepo yang perlu dipindahkan. 2. Untuk setiap layanan mikro, buat perubahan minimal dan tidak berdampak pada subfolder nya. Ini mungkin memperbarui `README` file, menambahkan komentar dalam file

Isu	Solusi
	konfigurasi, atau perubahan apa pun yang tidak memengaruhi fungsionalitas layanan. 3. Komit perubahan ini dengan pesan yang jelas (seperti “Inisiate redeployment of microservices”) dan dorong mereka ke repositori. Pastikan Anda mendorong perubahan ke cabang yang memulai penerapan. 4. Pantau saluran pipa untuk setiap layanan mikro untuk memastikan bahwa mereka dimulai dan diselesaikan dengan sukses. Kelebihan: • Memberikan kontrol granular atas layanan mikro mana yang dipindahkan. • Lebih aman karena tidak melibatkan penghapusan parameter konfigurasi yang mungkin digunakan untuk tujuan lain. Kontra: • Lebih memakan waktu, terutama dengan sejumlah besar layanan mikro. • Membutuhkan membuat perubahan kode yang tidak perlu yang dapat mengacaukan riwayat komit.

Sumber daya terkait

- Integrasi dan pengiriman berkelanjutan (CI/CD) menggunakan CDK Pipelines (dokumentasi)AWS CDK
- modul aws-cdk/pipelines (referensi API)AWS CDK

Integrasikan repositori Bitbucket dengan AWS Amplify menggunakan AWS CloudFormation

Alwin Abraham, Amazon Web Services

Ringkasan

AWS Amplify membantu Anda menerapkan dan menguji situs web statis dengan cepat tanpa harus menyiapkan infrastruktur yang biasanya diperlukan. Anda dapat menerapkan pendekatan pola ini jika organisasi Anda ingin menggunakan Bitbucket untuk kontrol sumber, baik untuk memigrasikan kode aplikasi yang ada atau membuat aplikasi baru. Dengan menggunakan AWS CloudFormation untuk mengatur Amplify secara otomatis, Anda memberikan visibilitas ke dalam konfigurasi yang Anda gunakan.

Pola ini menjelaskan cara membuat pipeline dan penerapan pipeline dan penerapan berkelanjutan front-end dan lingkungan penerapan berkelanjutan (CI/CD) dengan menggunakan AWS untuk mengintegrasikan repositori Bitbucket dengan CloudFormation AWS Amplify. Pendekatan pola berarti Anda dapat membuat pipeline front-end Amplify untuk penerapan berulang.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif
- Akun Bitbucket aktif dengan akses administrator
- Akses ke terminal yang menggunakan cURL atau aplikasi Postman
- Keakraban dengan Amplify
- Keakraban dengan AWS CloudFormation
- Keakraban dengan file berformat YAML

Arsitektur

Tumpukan teknologi

- Amplify
- AWS CloudFormation
- Bitbucket

Alat

- AWS Amplify — Amplify membantu pengembang mengembangkan dan menerapkan aplikasi seluler dan web bertenaga cloud.
- AWS CloudFormation — AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan menyiapkan sumber daya AWS sehingga Anda dapat menghabiskan lebih sedikit waktu untuk mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi yang berjalan di AWS.
- Bitbucket - Bitbucket adalah solusi manajemen repositori Git yang dirancang untuk tim profesional. Ini memberi Anda tempat sentral untuk mengelola repositori Git, berkolaborasi pada kode sumber Anda, dan memandu Anda melalui alur pengembangan.

Kode

`bitbucket-amplify.yml`File (terlampir) berisi CloudFormation template AWS untuk pola ini.

Epik

Konfigurasikan repositori Bitbucket

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Buat repositori Bitbucket.	1. Masuk ke akun Bitbucket Anda dan buat repositor i baru. Untuk informasi selengkapnya tentang hal ini, lihat Membuat repositor i Git dalam dokumentasi Bitbucket. 2. Catat nama ruang kerja. **Note** Anda juga dapat menggunakan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	repositori Bitbucket yang ada.	
Buka pengaturan ruang kerja.	1. Buka ruang kerja dan pilih tab Repositori. 2. Pilih repositori yang ingin Anda integrasikan dengan Amplify. 3. Pilih nama ruang kerja yang berada di atas nama repositori. 4. Di bilah sisi, pilih Pengaturan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat OAuth konsumen.	1. Di bagian Aplikasi dan Fitur, pilih OAuth konsumen, lalu pilih Tambah konsumen. 2. Masukkan nama untuk konsumen Anda, misalnya, `Amplify Integration .` 3. Masukkan URL panggilan balik. Meskipun bidang ini adalah input yang diperlukan, itu tidak digunakan untuk menyelesaikan integrasi sehingga nilainya bisa `http://localhost:3000` 4. Centang kotak untuk Ini adalah konsumen pribadi. 5. Pilih izin berikut: • Proyek - `Read` • Repositori — `Admin` • Permintaan tarik - `Read` • Webhooks — dan `Read` `Write` 6. Tinggalkan pilihan default untuk semua bidang lainnya dan pilih Kirim. 7. Catat kunci dan rahasia yang dihasilkan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan token OAuth akses.	1. Buka jendela terminal dan jalankan perintah berikut: `curl -X POST -u "KEY:SECRET" https://bitbucket.org/site/oauth2/access_token -d grant_type=client_credentials` **Important** Ganti `KEY` dan `SECRET` dengan kunci dan rahasia yang Anda rekam sebelumnya. 2. Rekam token akses tanpa menggunakan tanda kutip. Token hanya berlaku untuk waktu terbatas dan waktu default adalah dua jam. Anda harus menjalankan CloudFormation template AWS dalam jangka waktu ini.	DevOps insinyur

Buat dan terapkan tumpukan AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh CloudFormation templat AWS.	Unduh CloudFormation templat `bitbucket-`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`amplify.yml` AWS (terlampir). Template ini membuat CI/CD pipeline di Amplify, selain proyek dan cabang Amplify.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan terapkan CloudFormation tumpukan AWS.	1. Masuk ke AWS Management Console di Wilayah AWS yang ingin Anda terapkan dan buka CloudFormation konsol AWS. 2. Pilih Buat Tumpukan (dengan sumber daya baru) dan kemudian pilih Unggah File Template. 3. Unggah `bitbucket-amplify.yml` file. 4. Pilih Berikutnya, masukkan nama tumpukan, lalu masukkan parameter berikut: • Access token: Rekatkan token OAuth akses yang Anda buat sebelumnya. • URL Repositori: Tambahkan URL repositori proyek Bitbucket. URL biasanya dalam format berikut: `https://bitbucket.org/<WORKSPACE_NAME>/<REPO_NAME>` • Nama cabang: Ini harus cocok dengan nama cabang di repositori Bitbucket Anda. Cabang ini tidak perlu ada	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	saat Anda menjalank an CloudFormation tumpukan AWS tetapi diperlukan untuk menerapkan kode ke lingkungan. • Nama proyek: Ini adalah nama untuk dikaitkan dengan proyek Amplify. 5. Pilih Berikutnya dan kemudian pilih Create Stack.	

Uji CI/CD pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan kode ke cabang di repositori Anda.	1. Kloning repositori Bitbucket Anda dengan menjalank an perintah berikut: `git clone https://b itbucket.org/<WORK SPACE_NAME>/<REPO_ NAME>` 2. Lihat nama cabang yang digunakan saat menjalank an CloudFormation skrip AWS. Untuk membuat dan memeriksa cabang baru, jalankan `git checkout -b <BRANCH_NAME>` perintah. Untuk memeriksa cabang yang ada,	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	jalankan `git checkout <BRANCH_NAME>` perintah 3. Komit kode ke cabang dan dorong ke cabang jarak jauh dengan menjalankan `git push` perintah `git commit` and. 4. Amplify kemudian membangun dan menyebarkan aplikasi. Untuk informasi selengkap nya tentang ini, lihat Perintah Git Dasar dalam dokumentasi Bitbucket.	

Sumber daya terkait

Metode otentikasi (dokumentasi Atlassian)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Luncurkan CodeBuild proyek di seluruh akun AWS menggunakan Step Functions dan fungsi proxy Lambda

Richard Milner-Watts dan Amit Anjarlekar, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara meluncurkan proyek AWS secara asinkron CodeBuild di beberapa akun AWS dengan menggunakan AWS Step Functions dan fungsi proxy AWS Lambda. Anda dapat menggunakan contoh mesin status Step Functions pola untuk menguji keberhasilan CodeBuild proyek Anda.

CodeBuild membantu Anda menjalankan tugas operasional menggunakan AWS Command Line Interface (AWS CLI) dari lingkungan runtime yang dikelola sepenuhnya. Anda dapat mengubah perilaku CodeBuild proyek Anda saat runtime dengan mengganti variabel lingkungan. Selain itu, Anda dapat menggunakan CodeBuild untuk mengelola alur kerja. Untuk informasi selengkapnya, lihat Alat Service Catalog di situs web AWS Workshop dan Menjadwalkan pekerjaan di Amazon RDS for PostgreSQL CodeBuild menggunakan AWS EventBridge dan Amazon di Blog AWS Database.

Prasyarat dan batasan

Prasyarat

- Dua akun AWS aktif: akun sumber untuk menjalankan fungsi proxy Lambda dengan Step Functions dan akun target untuk membangun proyek sampel jarak jauh CodeBuild

Batasan

- Pola ini tidak dapat digunakan untuk menyalin artefak antar akun.

Arsitektur

Diagram berikut menunjukkan arsitektur yang dibangun pola ini.

Diagram menunjukkan alur kerja berikut:

1. Mesin status Step Functions mem-parsing peta input yang disediakan dan memanggil fungsi proxy Lambda (`codebuild-proxy-lambda`) untuk setiap akun, Wilayah, dan proyek yang Anda tentukan.
2. Fungsi proxy Lambda menggunakan AWS Security Token Service (AWS STS) untuk mengambil peran proxy IAM (`codebuild-proxy-role`), yang dikaitkan dengan kebijakan IAM (`codebuild-proxy-policy`) di akun target.
3. Menggunakan peran yang diasumsikan, fungsi Lambda meluncurkan CodeBuild proyek dan mengembalikan ID pekerjaan. CodeBuild Step Functions menyatakan mesin melakukan loop dan melakukan polling CodeBuild pekerjaan hingga menerima status sukses atau gagal.

Logika mesin negara ditunjukkan pada gambar berikut.

Tumpukan teknologi

- AWS CloudFormation
- CodeBuild
- IAM
- Lambda
- Step Functions
- X-Ray

Alat

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS CloudFormation Designer menyediakan editor JSON dan YAMAL terintegrasi yang membantu Anda melihat dan mengedit CloudFormation templat.
- AWS CodeBuild adalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi AWS Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis.
- AWS X-Ray membantu Anda mengumpulkan data tentang permintaan yang disajikan aplikasi Anda, dan menyediakan alat yang dapat Anda gunakan untuk melihat, memfilter, dan mendapatkan wawasan tentang data tersebut guna mengidentifikasi masalah dan peluang pengoptimalan.

Kode

Kode sampel untuk pola ini tersedia di repositori GitHub Cross Account CodeBuild Proxy. Pola ini menggunakan AWS Lambda Powertools for Python library untuk menyediakan fungsionalitas logging dan tracing. Untuk informasi selengkapnya tentang pustaka ini dan utilitasnya, lihat Powertools for AWS Lambda (Python).

Praktik terbaik

1. Sesuaikan nilai waktu tunggu di mesin status Fungsi Langkah untuk meminimalkan permintaan polling untuk status pekerjaan. Gunakan waktu eksekusi yang diharapkan untuk CodeBuild proyek.
2. Sesuaikan `MaxConcurrency` properti peta di Step Functions untuk mengontrol berapa banyak CodeBuild proyek yang dapat dijalankan secara paralel.
3. Jika diperlukan, tinjau kode sampel untuk kesiapan produksi. Pertimbangkan data apa yang mungkin dicatat oleh solusi dan apakah CloudWatch enkripsi Amazon default sudah cukup.

Epik

Buat fungsi proxy Lambda dan peran IAM terkait di akun sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rekam akun AWS IDs.	Akun AWS IDs diperlukan untuk mengatur akses di seluruh akun.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Catat ID akun AWS untuk akun sumber dan target Anda. Untuk informasi selengkap nya, lihat Menemukan ID akun AWS Anda di dokumentasi IAM.	
Unduh CloudFormation templat AWS.	1. Unduh CloudFormation template `sample_target_codebuild_template.yaml` AWS dari GitHub repositori untuk pola ini. 2. Unduh CloudFormation template `codebuild_lambda_proxy_template.yaml` AWS dari GitHub repositori untuk pola ini. **Note** Dalam CloudFormation templat AWS, `<SourceAccountId>` adalah ID akun AWS untuk akun sumber, dan `<TargetAccountId>` merupakan ID akun AWS untuk akun target.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan terapkan CloudFormation tumpukan AWS.	1. Masuk ke AWS Management Console untuk akun sumber Anda, buka CloudFormation konsol AWS, lalu pilih Stacks. 2. Pilih Buat tumpukan kemudian pilih Dengan sumber daya baru (standar). 3. Untuk Sumber templat, pilih Unggah file templat. 4. Untuk Unggah file templat, pilih file, lalu pilih `codebuild_lambda_proxy_template.yaml` file yang Anda unduh. Pilih Berikutnya. 5. Untuk nama Stack, masukkan nama untuk tumpukan (misalnya ,`codebuild-lambda-proxy`). 6. **Note** Ganti `crossAccountTargetRoleArn` parameter dengan Anda `<TargetAccountId>` (misalnya ,`<arn:aws:iam::1234`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	56789012:role/proxy-lambda-codebuild-role>). : Anda tidak diharuskan memperbarui nilai default untuk **targetCodeBuildProject** parameter. 7. Pilih Berikutnya, terima opsi pembuatan tumpukan default, lalu pilih Berikutnya. 8. Pilih kotak centang Saya mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM dengan nama khusus, lalu pilih Buat tumpukan. **Note** Anda harus membuat CloudFormation tumpukan AWS untuk fungsi Lambda proxy sebelum membuat sumber daya apa pun di akun target. Saat Anda membuat kebijakan kepercayaan di akun target,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	peran IAM diterjemahkan dari nama peran ke pengenal internal. Inilah sebabnya mengapa peran IAM harus sudah ada.	
Konfirmasikan pembuatan fungsi proxy dan mesin status.	1. Tunggu hingga AWS CloudFormation stack mencapai status CREATE_COMPLETE. Ini akan memakan waktu kurang dari satu menit. 2. Buka konsol AWS Lambda, pilih Fungsi, lalu temukan fungsinya. `lambda-proxy-ProxyLambda-<GUID>` 3. Buka konsol AWS Step Functions, pilih state machine, lalu temukan `sample-crossaccount-codebuild-state-machine` state machine.	AWS DevOps

Buat peran IAM di akun target dan luncurkan proyek sampel CodeBuild

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan terapkan CloudFormation tumpukan AWS.	1. Masuk ke AWS Management Console untuk akun target Anda, buka	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CloudFormation konsol AWS, lalu pilih Stacks. 2. Pilih Buat Tumpukan dan kemudian pilih Dengan sumber daya baru (standar) . 3. Untuk Sumber templat, pilih Unggah file templat. 4. Untuk Unggah file templat, pilih Pilih file, lalu pilih `sample_target_codebuild_template.yaml` file. Pilih Berikutnya. 5. Untuk nama Stack, masukkan nama untuk tumpukan (misalnya :`sample-codebuild-stack`). 6. Ganti `crossAccountSourceRoleArn` parameter dengan Anda `<SourceAccountId>` (misalnya,`<arn:aws:iam::123456789012:role/codebuild-proxy-lambda-role>`). 7. Pilih Berikutnya, terima opsi pembuatan tumpukan default, lalu pilih Berikutnya. 8. Pilih kotak centang Saya mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan nama khusus, lalu pilih Buat tumpukan.	
Verifikasi pembuatan CodeBuild proyek sampel.	1. Tunggu hingga AWS CloudFormation stack mencapai status CREATE_COMPLETE. Ini akan memakan waktu kurang dari satu menit. 2. Buka CodeBuild konsol AWS lalu temukan `sample-codebuild-project` proyeknya.	AWS DevOps

Uji fungsi proxy Lambda lintas akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan mesin negara.	1. Masuk ke AWS Management Console untuk akun sumber Anda, buka konsol AWS Step Functions, lalu pilih State machine. 2. Pilih mesin `sample-crossaccount-codebuild-state-machine` status dan kemudian pilih Mulai eksekusi. 3. Di editor Input, masukkan JSON berikut, dan ganti `<TargetAccountID>` dengan ID akun AWS dari akun yang berisi CodeBuild proyek.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>{
	  "crossAccountTargetRoleArns": [
	    {
	      "arn": "arn:aws:iam::<TargetAccountID>:role/proxy-lambda-codebuild-role",
	      "region": "eu-west-1",
	      "codeBuildProject": "sample-codebuild-project",
	      "SampleValue1": "Value1",
	      "SampleValue2": "Value2"
	    }
	  ]
	}</pre>
	Note
	Pasangan kunci-nilai dilewatkan sebagai variabel lingkungan dari fungsi di akun sumber ke CodeBuild proyek di akun target.
	4. Pilih Mulai Eksekusi.
	5. Pada tab Detail pada halaman mesin status, periksa apakah Status	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Eksekusi diatur ke Berhasil. Ini mengonfirmasi bahwa mesin status Anda sedang berjalan. Catatan: Diperlukan waktu sekitar 30 detik agar mesin status mencapai status Berhasil. 6. Untuk melihat output dan masukan langkah di mesin status, perluas langkah itu di bagian Riwayat peristiwa eksekusi. Misalnya, perluas langkah Lambda - CodeBuild Proxy — Mulai. Outputnya mencakup detail tentang variabel lingkungan yang diganti, muatan asli, dan ID pekerjaan. CodeBuild	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi variabel lingkungan.	1. Masuk ke AWS Management Console untuk akun target Anda. 2. Buka CodeBuild konsol AWS, perluas Build, lalu pilih Build projects. 3. Pilih `sample-codebuild-project` proyek, lalu pilih Lihat detail. 4. Pada tab Build history, pilih build terbaru proyek, lalu pilih Lihat log. 5. Dalam keluaran log, verifikasi bahwa variabel lingkungan yang dicetak ke STDOUT cocok dengan variabel lingkungan dari mesin status sampel Step Functions.	AWS DevOps

Pemecahan masalah

Isu	Solusi
Eksekusi Step Functions memakan waktu lebih lama dari yang diharapkan.	Sesuaikan `MaxConcurrency` properti peta di mesin status Step Function untuk mengontrol berapa banyak CodeBuild proyek yang dapat berjalan secara paralel.
Pelaksanaan CodeBuild pekerjaan memakan waktu lebih lama dari yang diharapkan.	1. Sesuaikan nilai waktu tunggu di mesin status Step Functions untuk meminimalkan permintaan polling untuk status pekerjaan.

Isu	Solusi
	Gunakan waktu eksekusi yang diharapkan untuk CodeBuild proyek. 2. Pertimbangkan CodeBuild apakah alat yang tepat untuk digunakan. Misalnya, waktu yang diperlukan untuk menginisialisasi CodeBuild pekerjaan bisa jauh lebih lama daripada AWS Lambda. Jika throughput tinggi dan waktu penyelesaian yang cepat merupakan persyaratan, pertimbangkan untuk memigrasikan logika bisnis ke AWS Lambda dan menggunakan arsitektur fan-out.

Kelola failover Multi-AZ untuk cluster EMR dengan menggunakan Application Recovery Controller

Aarti Rajput, Ashish Bhatt, Neeti Mishra, dan Nidhi Sharma, Amazon Web Services

Ringkasan

Pola ini menawarkan strategi pemulihan bencana yang efisien untuk beban kerja EMR Amazon untuk membantu memastikan ketersediaan tinggi dan konsistensi data di beberapa Availability Zone dalam satu. Wilayah AWS Desainnya menggunakan Amazon Application Recovery Controller dan Application Load Balancer untuk mengelola operasi failover dan distribusi lalu lintas untuk cluster EMR berbasis Apache Spark.

Dalam kondisi standar, Availability Zone utama menghosting cluster EMR aktif dan aplikasi dengan fungsionalitas penuh read/write . Jika Availability Zone gagal secara tak terduga, lalu lintas secara otomatis dialihkan ke Availability Zone sekunder, tempat klaster EMR baru diluncurkan. Kedua Availability Zone mengakses bucket Amazon Simple Storage Service (Amazon S3) bersama melalui titik akhir gateway khusus, yang memastikan pengelolaan data yang konsisten. Pendekatan ini meminimalkan waktu henti dan memungkinkan pemulihan cepat untuk beban kerja data besar yang penting selama kegagalan Availability Zone. Solusi ini berguna dalam industri seperti keuangan atau ritel, di mana analitik real-time sangat penting.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Amazon EMR di Amazon Elastic Compute Cloud (Amazon) EC2
- Akses dari simpul master cluster EMR ke Amazon S3.
- AWS Infrastruktur multi-AZ

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

- Amazon EMR 6.x dan rilis yang lebih baru

Arsitektur

Tumpukan teknologi target

- Kluster EMR Amazon
- Pengontrol Pemulihan Aplikasi Amazon
- Penyeimbang Beban Aplikasi
- Buket Amazon S3
- Titik akhir gateway untuk Amazon S3

Arsitektur target

Arsitektur ini memberikan ketahanan aplikasi dengan menggunakan beberapa Availability Zones dan menerapkan mekanisme pemulihan otomatis melalui Application Recovery Controller.

1. Application Load Balancer merutekan lalu lintas ke lingkungan EMR Amazon yang aktif, yang biasanya merupakan cluster EMR utama di Availability Zone utama.
2. Cluster EMR aktif memproses permintaan aplikasi dan terhubung ke Amazon S3 melalui titik akhir gateway Amazon S3 khusus untuk operasi baca dan tulis.
3. Amazon S3 berfungsi sebagai repositori data pusat dan berpotensi digunakan sebagai pos pemeriksaan atau sebagai penyimpanan bersama antara kluster EMR. Cluster EMR mempertahankan konsistensi data ketika mereka menulis langsung ke Amazon S3 melalui protokol `s3://` dan EMR File System (EMRFS).
4. Application Recovery Controller terus memantau kesehatan Availability Zone primer dan secara otomatis mengelola operasi failover bila diperlukan.
5. Jika Application Recovery Controller mendeteksi kegagalan dalam klaster EMR primer, dibutuhkan tindakan berikut:
   - Memulai proses failover ke cluster EMR sekunder di Availability Zone 2.
   - Memperbarui konfigurasi routing untuk mengarahkan lalu lintas ke cluster sekunder.

Alat

Layanan AWS

- Amazon Application Recovery Controller membantu Anda mengelola dan mengoordinasikan pemulihan aplikasi Anda di seluruh Wilayah AWS dan Availability Zone. Layanan ini menyederhanakan proses dan meningkatkan keandalan pemulihan aplikasi dengan mengurangi langkah-langkah manual yang diperlukan oleh alat dan proses tradisional.
- Application Load Balancer beroperasi pada layer aplikasi, yang merupakan lapisan ketujuh dari model Open Systems Interconnection (OSI). Ini mendistribusikan lalu lintas aplikasi yang masuk di beberapa target, seperti EC2 instance, di beberapa Availability Zone. Hal ini akan meningkatkan ketersediaan aplikasi Anda.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon EMR adalah platform data besar yang menyediakan pemrosesan data, analisis interaktif, dan pembelajaran mesin untuk kerangka kerja open source seperti Apache Spark, Apache Hive, dan Presto.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon S3 menyediakan antarmuka layanan web sederhana yang dapat Anda gunakan untuk menyimpan dan mengambil sejumlah data, kapan saja, dari mana saja. Dengan menggunakan layanan ini, Anda dapat dengan mudah membangun aplikasi yang menggunakan penyimpanan asli cloud.
- Titik akhir Gateway untuk Amazon S3 adalah gateway yang Anda tentukan dalam tabel rute untuk mengakses Amazon S3 dari virtual private cloud (VPC) melalui jaringan. AWS

Praktik terbaik

- Ikuti praktik AWS terbaik untuk keamanan, identitas, dan kepatuhan untuk memastikan arsitektur yang kuat dan aman.
- Sejajarkan arsitektur dengan AWS Well-Architected Framework.
- Gunakan Amazon S3 Access Grants untuk mengelola akses dari cluster EMR berbasis Spark ke Amazon S3. Untuk detailnya, lihat posting blog Gunakan Amazon EMR dengan Hibah Akses S3 untuk Menskalakan akses Spark ke Amazon S3.

- Tingkatkan kinerja Spark dengan Amazon S3.

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke Konsol Manajemen AWS.	Masuk ke Konsol Manajemen AWS sebagai pengguna IAM. Untuk instruksi, lihat AWS dokumentasi.	AWS DevOps
Konfigurasikan AWS CLI.	Instal AWS CLI atau perbarui ke versi terbaru sehingga Anda dapat berinteraksi dengan Layanan AWS di Konsol Manajemen AWS. Untuk instruksi, lihat AWS CLI dokumentasi.	AWS DevOps

Menerapkan aplikasi Spark di cluster EMR Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	1. Buat bucket S3 untuk menyimpan dataset input, log, aplikasi, dan data output. Untuk petunjuk, lihat dokumentasi Amazon S3. 2. Atur bucket ke dalam folder terpisah untuk input data (`dataset`), logs (`logs`), Spark application (`spark-app `), dan output data (`output`).	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat cluster EMR.	1. Gunakan AWS CLI perintah berikut untuk membuat klaster EMR (misalnya, versi 6.12 atau yang lebih baru) dengan instance yang menjangkau dua Availability Zone (seperti `us-east-1a` dan `us-east-1b`) untuk ketersediaan tinggi. Perintah menentukan jenis `m4.large` instance sebagai contoh.
	`aws emr create-cluster \ --ec2-attributes AvailabilityZone=<AZ-name-1> \ --release-label emr-6.12.0 \ --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.large`
	`aws emr create-cluster \ --ec2-attributes AvailabilityZone=<AZ-name-2> \ --release-label emr-6.12.0 \ --instance-groups InstanceGroupType=`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.large` Untuk informasi selengkapnya, lihat perintah create-cluster dan dokumentasi Amazon EMR. 2. Berikan key pair, peran layanan, dan profil instance dengan izin yang diperlukan, jika diperlukan.	
Konfigurasikan pengaturan keamanan untuk cluster EMR.	1. Identifikasi grup keamanan yang terkait dengan master node EMR cluster dengan menggunakan perintah AWS CLI describe-cluster: `aws emr describe-cluster --cluster-id j-XXXXXXXX` 2. Untuk meningkatkan keamanan, ubah pengaturan grup keamanan untuk mengizinkan akses SSH (port TCP 22) ke node master, tetapi batasi ke alamat IP spesifik Anda. Untuk informasi selengkapnya, lihat dokumentasi Amazon EMR.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Connect ke cluster EMR.	Connect ke master node dari EMR cluster melalui SSH dengan menggunakan key pair yang disediakan.
	
	Pastikan bahwa file key pair hadir di direktori yang sama dengan aplikasi Anda.
	
	Jalankan perintah berikut untuk mengatur izin yang benar untuk key pair dan untuk membuat koneksi SSH:
	
	`chmod 400 <key-pair-name>`
	`ssh -i ./<key-pair-name> hadoop@<master-node-public-dns>`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan aplikasi Spark.	Setelah Anda membuat koneksi SSH, Anda akan berada di konsol Hadoop.
	
	1. Buat atau edit file aplikasi Spark (`main.py`) dengan menggunakan editor teks seperti vim:
	
	`vim main.py`
	
	Untuk informasi selengkapnya tentang membuat dan memodifikasi aplikasi Spark, lihat dokumentasi Amazon EMR.
	2. Kirimkan aplikasi Spark ke cluster EMR, dengan menentukan data input dan lokasi data output di bucket S3:
	
	`spark-submit main.py –data_source <input-data-folder-in-s3> –output_uri <output-folder-in-s3>`
	
	Berikut adalah contoh berdasarkan folder yang Anda atur sebelumnya:
	
	`spark-submit main.py –data_source dataset –output_uri output`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pantau kemajuan aplikasi dengan memeriksa log aplikasi: `yarn logs -applicationId <application-id>`	
Pantau aplikasi Spark.	1. Buka jendela terminal lain dan buat terowongan SSH ke UI web pengelola sumber daya kluster EMR: `ssh -i <key-pair-name> -N -L 8157:<resource-manager-public-dns>:8088 hadoop@<resource-manager-public-dns>` 2. Untuk memantau aplikasi, akses UI web pengelola sumber daya dengan menavigasi ke `http://localhost:8157` browser web Anda.	AWS DevOps

Alihkan lalu lintas ke Availability Zone lain

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Application Load Balancer.	Siapkan grup target yang merutekan lalu lintas antara node master EMR Amazon yang digunakan di dua	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Availability Zone dalam file. Wilayah AWS Untuk petunjuk, lihat Membuat grup target untuk Application Load Balancer Anda dalam dokumentasi Elastic Load Balancing.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pergeseran zona di Application Recovery Controller.	Pada langkah ini, Anda akan menggunakan fitur zonal shift di Application Recovery Controller untuk mengalihkan lalu lintas ke Availability Zone lain. 1. Buka konsol Application Recovery Controller. 2. Di bawah Memulai, pilih Pergeseran zona, Mulai pergeseran zona. 3. Pilih Availability Zone yang ingin Anda alihkan lalu lintas. 4. Pilih sumber daya yang didukung (misalnya, Application Load Balancer) untuk pergeseran zona dari tabel Resources. 5. Untuk Mengatur kedaluwarsa pergeseran zona, pilih atau masukkan kedaluwarsa untuk pergeseran zona. Anda dapat mengatur durasi antara 1 menit dan tiga hari (72 jam). Semua pergeseran zona bersifat sementara. Anda harus menetapkan kedaluwarsa, tetapi Anda dapat memperbarui shift aktif nanti untuk menetapka	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n periode kedaluwarsa baru hingga tiga hari. 6. Masukkan komentar tentang pergeseran zona ini. 7. Pilih kotak centang untuk mengetahui bahwa memulai pergeseran zona akan mengurangi kapasitas yang tersedia untuk aplikasi Anda dengan mengalihkan lalu lintas dari Availability Zone. 8. Pilih Mulai. Untuk menggunakan AWS CLI, lihat Contoh penggunaan AWS CLI dengan pergeseran zona dalam dokumentasi Application Recovery Controller.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi konfigurasi dan kemajuan pergeseran zona.	1. Verifikasi sumber daya yang terdaftar dengan zonal shift:
	`aws arc-zonal-shift list-managed-resources --region <AWS-region-name>`
	Misalnya, output berikut mengonfirmasi bahwa sumber daya aktif dan berjalan di kedua Availability Zones.
	`"appliedWeights": { "use1-az1": 1.0, "use1-az2": 1.0 },`
	2. Untuk memvisualisasikan pergeseran zona, gunakan AWS CLI perintah berikut untuk memulai pergeseran zona:
	`aws arc-zonal-shift start-zonal-shift \ --resource-identifier <application-load-balancer-arn> \ --away-from <source-AZ> \ --expires-in 10m --comment "testing" \`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--region <AWS-regi
	on-name></pre>
	di mana <source-AZ> adalah pengenal Availability Zone yang ingin Anda alihkan lalu lintas, dan <application-load-balancer-arn> merupakan Amazon Resource Name (ARN) dari Application Load Balancer Anda.
	3. Verifikasi bahwa lalu lintas telah bergeser ke Availability Zone lain.
	<pre>aws arc-zonal-shift
	get-managed-resour
	ce \
	--resource-identif
	ier <application-
	load-balancer-arn> \
	--region <AWS-regi
	on-name></pre>
	Anda dapat melihat pergeseran zona yang dikonfirmasi oleh bobot ini:
	<pre>"appliedWeights": {
	"use1-az1":
	0.0,
	"use1-az2":
	1.0
	},</pre>	

Sumber daya terkait

- AWS CLI perintah:
  - buat-cluster
  - deskripsikan-cluster
  - arc-zonal-shift
- Mengonfigurasi jenis instans klaster EMR Amazon dan praktik terbaik untuk instans Spot (dokumentasi EMR Amazon)
- Praktik terbaik keamanan dalam IAM (dokumentasi IAM)
- Gunakan profil instance (dokumentasi IAM)
- Gunakan zonal shift dan zonal autoshift untuk memulihkan aplikasi di ARC (dokumentasi Application Recovery Controller)

Kelola blue/green penerapan layanan mikro ke beberapa akun dan Wilayah dengan menggunakan layanan kode AWS dan kunci Multi-wilayah AWS KMS

Balaji Vedagiri, Vanitha Dontireddy, Ashish Kumar, Faisal Shahdad, Vivek Thangamuthu, dan Anand Krishna Varanasi, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerapkan aplikasi layanan mikro global dari akun AWS pusat ke beberapa akun beban kerja dan Wilayah sesuai dengan strategi penerapan. blue/green Pola ini mendukung hal-hal berikut:

- Perangkat lunak dikembangkan di akun pusat, sedangkan beban kerja dan aplikasi tersebar di beberapa akun dan Wilayah AWS.
- Kunci multi-wilayah AWS Key Management System (AWS KMS) tunggal digunakan untuk enkripsi dan dekripsi guna menutupi pemulihan bencana.
- Kunci KMS adalah khusus Wilayah dan harus dipelihara atau dibuat di tiga Wilayah berbeda untuk artefak pipa. Kunci Multi-wilayah KMS membantu mempertahankan ID kunci yang sama di seluruh Wilayah.
- Model percabangan alur kerja Git diimplementasikan dengan dua cabang (pengembangan dan utama) dan kode digabungkan dengan menggunakan pull requests (). PRs Fungsi AWS Lambda yang diterapkan dari tumpukan ini membuat PR dari cabang pengembangan ke cabang utama. Penggabungan PR ke cabang utama memulai CodePipeline pipeline AWS, yang mengatur aliran integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) dan menyebarkan tumpukan di seluruh akun.

Pola ini menyediakan contoh infrastruktur sebagai penyiapan kode (IAc) melalui CloudFormation tumpukan AWS untuk mendemonstrasikan kasus penggunaan ini. blue/green Penyebaran layanan mikro diimplementasikan dengan menggunakan AWS. CodeDeploy

Prasyarat dan batasan

Prasyarat

- Empat akun AWS aktif:
  - Akun alat untuk mengelola pipeline kode dan memelihara CodeCommit repositori AWS.

  - Tiga akun beban kerja (pengujian) untuk menerapkan beban kerja layanan mikro.
- Pola ini menggunakan Wilayah berikut. Jika Anda ingin menggunakan Wilayah lain, Anda harus melakukan modifikasi yang sesuai pada tumpukan Multi-wilayah AWS CodeDeploy dan AWS KMS.
  - Akun Tools (AWS CodeCommit): `ap-south-1`
  - Akun beban kerja (tes) 1: `ap-south-1`
  - Akun beban kerja (tes) 2: `eu-central-1`
  - Akun beban kerja (tes) 3: `us-east-1`
- Tiga bucket Amazon Simple Storage Service (Amazon S3) untuk Wilayah penerapan di setiap akun beban kerja. (Ini disebut`S3BUCKETNAMETESTACCOUNT1`, `S3BUCKETNAMETESTACCOUNT2` dan `S3BUCKETNAMETESTACCOUNT3` kemudian dalam pola ini.)

  Misalnya, Anda dapat membuat bucket ini di akun dan Wilayah tertentu dengan nama bucket unik sebagai berikut (ganti xxxx dengan nomor acak):

```
##In Test Account 1
aws s3 mb s3://ecs-codepipeline-xxxx-ap-south-1 --region ap-south-1
##In Test Account 2
aws s3 mb s3://ecs-codepipeline-xxxx-eu-central-1 --region eu-central-1
##In Test Account 3
aws s3 mb s3://ecs-codepipeline-xxxx-us-east-1 --region us-east-1

#Example
##In Test Account 1
aws s3 mb s3://ecs-codepipeline-18903-ap-south-1 --region ap-south-1
##In Test Account 2
aws s3 mb s3://ecs-codepipeline-18903-eu-central-1 --region eu-central-1
##In Test Account 3
aws s3 mb s3://ecs-codepipeline-18903-us-east-1 --region us-east-1
```

Batasan

Pola ini menggunakan AWS CodeBuild dan file konfigurasi lainnya untuk menerapkan layanan mikro sampel. Jika Anda memiliki jenis beban kerja yang berbeda (misalnya, tanpa server), Anda harus memperbarui semua konfigurasi yang relevan.

Arsitektur

Tumpukan teknologi target

- AWS CloudFormation
- AWS CodeCommit
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

Arsitektur target

Otomatisasi dan skala

Penyiapan otomatis dengan menggunakan AWS CloudFormation stack templates (IAc). Ini dapat dengan mudah diskalakan untuk beberapa lingkungan dan akun.

Alat

Layanan AWS

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS CodeBuild adalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodeDeploy mengotomatiskan penerapan ke Amazon Elastic Compute Cloud (Amazon EC2) atau instans lokal, fungsi AWS Lambda, atau layanan Amazon Elastic Container Service (Amazon ECS).
- AWS CodePipeline membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.

- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat tambahan

- Git adalah sistem kontrol versi terdistribusi open-source yang bekerja dengan CodeCommit repositori AWS.
- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah. Pola ini menggunakan Docker untuk membangun dan menguji gambar kontainer secara lokal.
- cfn-lint dan cfn-nag adalah alat sumber terbuka yang membantu Anda meninjau CloudFormation tumpukan untuk setiap kesalahan dan masalah keamanan.

Repositori kode

Kode untuk pola ini tersedia di Blue/Green penerapan GitHub Global di beberapa wilayah dan repositori akun.

Epik

Mengatur variabel lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekspor variabel lingkungan untuk penyebaran CloudFormation tumpukan.	Tentukan variabel lingkungan yang akan digunakan sebagai masukan ke CloudFormation tumpukan nanti dalam pola ini. 1. Perbarui nama bucket yang Anda buat di tiga akun dan Wilayah seperti yang dijelaskan sebelumnya di bagian Prasyarat:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` export S3BUCKETN AMETESTACCOUNT1=<S 3BUCKETACCOUNT1> export S3BUCKETN AMETESTACCOUNT2=<S 3BUCKETACCOUNT2> export S3BUCKETN AMETESTACCOUNT3=<S 3BUCKETACCOUNT3> ``` 2. Tentukan string acak untuk membuat bucket artefak, karena nama bucket harus unik secara global: ``` export BUCKETSTA RTNAME=ecs-codepip eline-artifacts-19 992 ``` 3. Tentukan dan ekspor akun IDs dan Wilayah: ``` export TOOLSACCO UNT=<TOOLSACCOUNT> export CODECOMMI TACCOUNT=<CODECOMM ITACCOUNT> export CODECOMMI TREGION=ap-south-1 export CODECOMMI TREPONAME=Poc export TESTACCOU NT1=<TESTACCOUNT1> export TESTACCOU NT2=<TESTACCOUNT2> export TESTACCOU NT3=<TESTACCOUNT3> 	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`export TESTACCOU NT1REGION=ap-south -1`
	`export TESTACCOU NT2REGION=eu-centr al-1`
	`export TESTACCOU NT3REGION=us-east-1`
	`export TOOLSACCO UNTREGION=ap-south -1`
	`export ECRREPOSI TORYNAME=web`	

Package dan deploy CloudFormation tumpukan untuk infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori.	Kloning repositori sampel ke repositori baru di lokasi kerja Anda:
	`##In work location`
	`git clone https://g ithub.com/aws-samp les/ecs-blue-green -global-deployment- with-multiregion-cmk- codepipeline.git`
		AWS DevOps

Package resource Cloudform ation.	Pada langkah ini, Anda mengemas artefak lokal yang referensi CloudFormation template untuk membuat sumber daya infrastruktur yang diperlukan untuk layanan seperti Amazon Virtual Private	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Cloud (Amazon VPC) dan Application Load Balancer.
	
	Template tersedia di `Infra` folder repositori kode.
	
	`##In TestAccount1##`
	`aws cloudformation package \`
	`--template-file mainInfraStack.yaml \`
	`--s3-bucket $S3BUCKETNAMETESTACCOUNT1 \`
	`--s3-prefix infraStack \`
	`--region $TESTACCOUNT1REGION \`
	`--output-template-file infrastructure_${TESTACCOUNT1}.template`
	
	`##In TestAccount2##`
	`aws cloudformation package \`
	`--template-file mainInfraStack.yaml \`
	`--s3-bucket $S3BUCKETNAMETESTACCOUNT2 \`
	`--s3-prefix infraStack \`
	`--region $TESTACCOUNT2REGION \`
	`--output-template-file infrastructure_${TESTACCOUNT2}.template`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	``
	##In TestAccount3##
	aws cloudformation
	 package \
	    --template-file
	 mainInfraStack.yaml \
	    --s3-bucket
	 $S3BUCKETNAMETESTA
	CCOUNT3 \
	    --s3-prefix
	 infraStack \
	    --region $TESTACCO
	UNT3REGION \
	    --output-template-
	file infrastructure_
	${TESTACCOUNT3}.templ
	ate
	``	

Validasi template paket.	Validasi template paket:
	
	``
	aws cloudformation
	 validate-template \
	    --template-body
	 file://infrastruct
	ure_${TESTACCOUNT1
	}.template
	
	aws cloudformation
	 validate-template \
	    --template-body
	 file://infrastruct
	ure_${TESTACCOUNT2
	}.template
	
	aws cloudformation
	 validate-template \
	    --template-body
	 file://infrastruct
	ure_${TESTACCOUNT3
	}.template
	``
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan file paket ke akun beban kerja,	1. Perbarui nilai placeholder dan nama akun di `nfraParameters.json` skrip i berdasarkan pengaturan Anda.
	2. Terapkan templat paket ke dalam tiga akun beban kerja Anda.
	
	`##In TestAccount1##`
	`aws cloudformation deploy \`
	`--template-file infrastructure_${TESTACCOUNT1}.template \`
	`--stack-name mainInfrastack \`
	`--parameter-overrides file://infraParameters.json \`
	`--region $TESTACCOUNT1REGION \`
	`--capabilities CAPABILITY_IAM CAPABILITY_NAMED_IAM`
	
	`##In TestAccount2##`
	`aws cloudformation deploy \`
	`--template-file infrastructure_${TESTACCOUNT2}.template \`
	`--stack-name mainInfrastack \`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> --parameter-
	overrides file://in
	fraParameters.json \
	    --region $TESTACCO
	UNT2REGION \
	    --capabilities
	 CAPABILITY_IAM
	 CAPABILITY_NAMED_I
	AM
	
	##In TestAccount3##
	aws cloudformation
	 deploy \
	    --template-file
	 infrastructure_${T
	ESTACCOUNT3}.templ
	ate \
	    --stack-name
	 mainInfrastack \
	    --parameter-
	overrides file://in
	fraParameters.json \
	    --region $TESTACCO
	UNT3REGION \
	    --capabilities
	 CAPABILITY_IAM
	 CAPABILITY_NAMED_I
	AM</pre>	

Dorong gambar sampel dan skala Amazon ECS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dorong gambar sampel ke repositori Amazon ECR.	Dorong gambar sampel (NGINX) ke repositori web Amazon Elastic Container Registry (Amazon ECR) bernama (seperti yang	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	ditetapkan dalam parameter). Anda dapat menyesuaikan gambar sesuai kebutuhan.
	
	Untuk masuk dan menyetel kredensional untuk mendorong gambar ke Amazon ECR, ikuti petunjuk dalam dokumentasi Amazon ECR.
	
	Perintahnya adalah:
	
	`docker pull nginx`
	`docker images`
	`docker tag <imageid> aws_account_id.dkr.ecr.region.amazonaws.com/<web>:latest`
	`docker push <aws_account_id>.dkr.ecr.<region>.amazonaws.com/<web>:tag`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Skala Amazon ECS dan verifikasi akses.	1. Skala Amazon ECS untuk membuat dua replika: `aws ecs update-service --cluster QA-Cluster --service Poc-Service --desired-count 2` di mana `Poc-Service` mengacu pada aplikasi sampel Anda. 2. Verifikasi bahwa layanan dapat diakses dari Application Load Balancer dengan menggunakan nama domain yang sepenuhnya memenuhi syarat (FQDN) atau DNS dari browser atau dengan menggunakan perintah curl.	AWS DevOps

Siapkan layanan kode dan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodeCommit repositori di akun alat.	Buat CodeCommit repositori di akun alat dengan menggunakan `codecommit.yaml` template, yang ada di `code` folder repositori. GitHub Anda harus membuat repositori ini hanya di Wilayah tunggal di	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mana Anda berencana untuk mengembangkan kode. `aws cloudformation deploy --stack-name codecommitrepoStack --parameter-overrides CodeCommitReponame=$CODECOMMITREPONAME \ ToolsAccount=$TOOLSACCOUNT --template-file codecommit.yaml --region $TOOLSACCOUNTREGION \ --capabilities CAPABILITY_NAMED_IAM`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 untuk mengelola artefak yang dihasilkan oleh. CodePipeline	Buat bucket S3 untuk mengelola artefak yang dihasilkan CodePipeline dengan menggunakan `pre-reqs-bucket.yaml` template, yang ada di `code` folder repositori. GitHub Tumpukan harus digunakan di ketiga beban kerja (pengujian) dan akun alat dan Wilayah. `aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region $TESTACCOUNT1REGION --capabilities CAPABILITY_NAMED_IAM`  `aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketSta`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	rtName=$BUCKETSTAR TNAME \ TestAccount1=$TE STACCOUNT1 TestAccou nt2=$TESTACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=$BUCKETSTAR TNAME \ TestAccount1=$TE STACCOUNT1 TestAccou nt2=$TESTACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region $TOOLSACCOUNTREGION --capabilities CAPABILITY_NAMED_IAM`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kunci KMS Multi-wil ayah.	1. Buat kunci KMS Multi-wil ayah dengan kunci primer dan replika yang akan digunakan. CodePipel ine Dalam contoh kita, `ToolsAccount1regio n - ap-south-1` akan menjadi Wilayah utama. `aws cloudformation deploy --stack-name ecs-codepipeline-p re-reqs-KMS \ --template-file pre- reqs_KMS.yaml -- parameter-overrides \ TestAccount1=$TE STACCOUNT1 TestAccou nt2=$TESTACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT --region $TOOLSACC OUNTREGION` 2. Atur variabel CMKARN untuk diteruskan ke CodeBuild proyek. Nilai tersedia dalam output tumpukan template ecs- codepipeline-pre-reqs -KMS (ID kunci akan sama di semua Wilayah dan dimulai dengan`mrk-`). Atau, Anda bisa mendapatkan nilai	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	CMKARN dari akun alat. Ekspor mereka di semua sesi akun:
	
	`export CMKARN1=arn:aws:kms:ap-south-1:<TOOLSACCOUNTID>:key/mrk-xxx`
	`export CMKARN2=arn:aws:kms:eu-central-1:<TOOLSACCOUNTID>:key/mrk-xxx`
	`export CMKARN3=arn:aws:kms:us-east-1:<TOOLSACCOUNTID>:key/mrk-xxx`
	`export CMARNTOOLS=arn:aws:kms:ap-south-1:<TOOLSACCOUNTID>:key/mrk-xxx`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan CodeBuild proyek di akun alat.	1. Gunakan `codebuild_IAM.yaml` template dari `code` folder GitHub repositori untuk menyiapkan AWS Identity and Access Management (IAM) untuk AWS CodeBuild di satu Wilayah di akun alat:
	`#In ToolsAccount aws cloudformation deploy --stack-name ecs-codebuild-iam \ --template-file codebuild_IAM.yaml --region $TOOLSACCOUNTREGION \ --capabilities CAPABILITY_NAMED_IAM`
	2. Gunakan `codebuild.yaml` template CodeBuild untuk menyiapkan proyek build Anda. Menyebarkan template ini di ketiga Wilayah sebagai berikut:
	`aws cloudformation deploy --stack-name ecscodebuildstack --parameter-overrides ToolsAccount=$TOOLSACCOUNT \ CodeCommitRepoName=$CODECOMMITREPONAME ECRRepositoryName=`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`$ECRREPOSITORYNAME APPACCOUNTID=$TEST ACCOUNT1 \ TestAccount3=$TE STACCOUNT3 CodeCommi tRegion=$CODECOMMI TREGION CMKARN=$C MKARN1 \ --template-file codebuild.yaml --region $TESTACCO UNT1REGION --capabil ities CAPABILIT Y_NAMED_IAM` `aws cloudformation deploy --stack-name ecscodebuildstack -- parameter-overrides ToolsAccount=$TOOL SACCOUNT \ CodeCommitRepoName= $CODECOMMITREPONAME ECRRepositoryName= $ECRREPOSITORYNAME APPACCOUNTID=$TEST ACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tRegion=$CODECOMMI TREGION CMKARN=$C MKARN2 \ --template-file codebuild.yaml --region $TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM`  `aws cloudformation deploy --stack-name ecscodebuildstack --`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`parameter-overrides ToolsAccount=$TOOLSACCOUNT \ CodeCommitRepoName=$CODECOMMITREPONAME ECRRepositoryName=$ECRREPOSITORYNAME APPACCOUNTID=$TESTACCOUNT3 \ CodeCommitRegion=$CODECOMMITREGION CMKARN=$CMKARN3 \ --template-file codebuild.yaml --region $TESTACCOUNT3REGION --capabilities CAPABILITY_NAMED_IAM`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan CodeDeploy di akun beban kerja.	Gunakan `codedeploy.yaml` template di `code` folder GitHub repositori untuk mengatur CodeDeploy di ketiga akun beban kerja. Outputnya `mainInfraStack` mencakup Amazon Resource Names (ARNs) dari cluster Amazon ECS dan pendengar Application Load Balancer.
	
	Note
	Nilai dari tumpukan infrastruktur sudah diekspor, sehingga diimpor oleh template CodeDeploy tumpukan.
	
	`##WorkloadAccount1##`
	`aws cloudformation deploy --stack-name ecscodedeploystack \`
	`--parameter-overrides ToolsAccount=$TOOLSACCOUNT mainInfrastackname=mainInfrastack \`
	`--template-file codedeploy.yaml --region $TESTACCOUNT1REGION --capabilities CAPABILITY_NAMED_IAM`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` ##WorkloadAccount2## aws cloudformation deploy --stack-name ecscodedeploystack \ --parameter-overrides ToolsAccount=$TOOL SACCOUNT mainInfra stackname=mainInfr astack \ --template-file codedeploy.yaml --region $TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM ##WorkloadAccount3## aws cloudformation deploy --stack-name ecscodedeploystack \ --parameter-overrides ToolsAccount=$TOOL SACCOUNT mainInfra stackname=mainInfr astack \ --template-file codedeploy.yaml --region $TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM ```	

Siapkan CodePipeline di akun alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pipeline kode di akun alat.	Di akun alat, jalankan perintah:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws cloudformation deploy --stack-name ecscodepipelinestack --parameter-overrides \ TestAccount1=$TESTACCOUNT1 TestAccount1Region=$TESTACCOUNT1REGION \ TestAccount2=$TESTACCOUNT2 TestAccount2Region=$TESTACCOUNT2REGION \ TestAccount3=$TESTACCOUNT3 TestAccount3Region=$TESTACCOUNT3REGION \ CMKARNTools=$CMKTROOLSARN CMKARN1=$CMKARN1 CMKARN2=$CMKARN2 CMKARN3=$CMKARN3 \ CodeCommitRepoName=$CODECOMMITREPONAME BucketStartName=$BUCKETSTARTNAME \ --template-file codepipeline.yaml --capabilities CAPABILITY_NAMED_IAM`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan akses CodePipeline dan CodeBuild peran dalam kebijakan kunci AWS KMS dan kebijakan bucket S3.	1. Menyediakan akses CodePipeline dan CodeBuild peran dalam kebijakan kunci AWS KMS: `aws cloudformation deploy --stack-name ecs-codepipeline-pre-reqs-KMS \ --template-file pre-reqs_KMS.yaml --parameter-overrides \ CodeBuildCondition=true TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT --region $TOOLSACCOUNTREGION` 2. Perbarui kebijakan bucket S3 untuk mengizinkan akses CodePipeline dan CodeDeploy peran: `aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ PutS3BucketPolicy=true TestAccou`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nt1=$TESTACCOUNT1 TestAccount2=$TEST ACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT1REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=$BUCKETSTAR TNAME \ PutS3BucketPolic y=true TestAccou nt1=$TESTACCOUNT1 TestAccount2=$TEST ACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ PutS3BucketPolicy=true TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region $TESTACCOUNT3REGION --capabilities CAPABILITY_NAMED_IAM`  `aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ PutS3BucketPolicy=true TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region $TOOLSACCOUNTREGION --capabil`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ities CAPABILIT Y_NAMED_IAM`	

Panggil dan uji pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dorong perubahan ke CodeCommit repositori.	1. Kloning CodeCommit repositori yang dibuat di `codecommitrepoStac k` dengan menggunak an `git clone` perintah, seperti yang dijelaskan dalam dokumentasi AWS. CodeCommit 2. Perbarui artefak input dengan detail yang diperlukan: • File JSON: Perbarui `AccountID` file di tiga tempat file ini. Ganti nama ketiga file untuk menyertakan akun IDs. • File YAMB: Perbarui definisi tugas ARN dan versi. Ganti nama ketiga file untuk menyertakan akun IDs. 3. Ubah `index.html` file untuk membuat beberapa perubahan kecil pada halaman beranda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Salin file berikut ke repositori dan komit:
	
	`index.html`
	`Dockerfile`
	`buildspec.yaml`
	`appspec_<accountid>.yaml (3 files - one per account )`
	`taskdef<accountid>.json (3 files - one per account)`
	
	5. Mulai atau mulai ulang pipa dan verifikasi hasilnya.
	6. Akses layanan dari Application Load Balancer menggunakan menggunakan FQDN atau DNS, dan verifikasi bahwa pembaruan telah digunakan.	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan semua sumber daya yang digunakan.	1. Turunkan Amazon ECS ke nol instans: `aws ecs update-service --cluster QA-Cluster --service Poc-Service --desired-count 0`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Hapus CloudFormation tumpukan di setiap akun dan Wilayah:
	
	`##In Tools Account##`
	`aws cloudformation delete-stack --stack-name ecscodepipelinestack --region $TOOLSACCOUNTREGION`
	`aws cloudformation delete-stack --stack-name ecscodebuildstack --region $TESTACCOUNT1REGION`
	`aws cloudformation delete-stack --stack-name ecscodebuildstack --region $TESTACCOUNT2REGION`
	`aws cloudformation delete-stack --stack-name ecscodebuildstack --region $TESTACCOUNT3REGION`
	`aws cloudformation delete-stack --stack-name ecs-codepipeline-pre-reqs-KMS --region $TOOLSACCOUNTREGION`
	`aws cloudformation delete-stack --stack-name codecommitrepoStack --region $TOOLSACCOUNTREGION`
	`aws cloudformation delete-stack --stack-name pre-reqs-artifacts-bucket`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--region $TESTACCOUNT1REGION`
	`aws cloudformation delete-stack --stack-name pre-reqs-artifacts-bucket --region $TESTACCOUNT2REGION`
	`aws cloudformation delete-stack --stack-name pre-reqs-artifacts-bucket --region $TESTACCOUNT3REGION`
	`aws cloudformation delete-stack --stack-name pre-reqs-artifacts-bucket --region $TOOLSACCOUNTREGION`
	`aws cloudformation delete-stack --stack-name ecs-codebuild-iam --region $TOOLSACCOUNTREGION`
	
	`##NOTE: Artifact buckets will not get deleted if there are artifacts so it has to be emptied manually before deleting.##`	

	`##In Workload / Test Accounts##`
	`##Account:1##`
	`aws cloudformation delete-stack --stack-name ecscodede`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`ploystack --region $TESTACCOUNT1REGION`
	`aws cloudformation delete-stack --stack-name mainInfrastack --region $TESTACCOUNT1REGION`
	`##Account:2##`
	`aws cloudformation delete-stack --stack-name ecscodedeployStack --region $TESTACCOUNT2REGION`
	`aws cloudformation delete-stack --stack-name mainInfrastack --region $TESTACCOUNT2REGION`
	`##Account:3##`
	`aws cloudformation delete-stack --stack-name ecscodedeployStack --region $TESTACCOUNT3REGION`
	`aws cloudformation delete-stack --stack-name mainInfrastack --region $TESTACCOUNT3REGION`
	`##NOTE: Amazon ECR (web) will not get deleted if the registry still includes images. It can be manually cleaned up if not required.`	

Pemecahan masalah

Isu	Solusi
Perubahan yang Anda komit ke repositori tidak diterapkan.	• Periksa CodeBuild log untuk kesalahan dalam tindakan build Docker. Lihat informasi yang lebih lengkap dalam dokumentasi CodeBuild . • Periksa CodeDeploy penerapan untuk masalah penyebaran Amazon ECS.

Sumber daya terkait

- Mendorong gambar Docker (dokumentasi Amazon ECR)
- Connect ke CodeCommit repositori AWS (dokumentasi CodeCommit AWS)
- Memecahkan masalah AWS (dokumentasi CodeBuild CodeBuild AWS)

Pantau repositori Amazon ECR untuk izin wildcard menggunakan AWS dan AWS Config CloudFormation

Vikrant Telkar, Wassim Benhallam, dan Sajid Momin, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri image container terkelola yang mendukung repositori pribadi dengan izin berbasis sumber daya menggunakan AWS Identity and Access Management (IAM).

IAM mendukung wildcard * "" di atribut sumber daya dan tindakan, yang membuatnya lebih mudah untuk secara otomatis memilih beberapa item yang cocok. Di lingkungan pengujian, Anda dapat mengizinkan semua pengguna AWS yang diautentikasi mengakses repositori ECR Amazon dengan menggunakan izin `ecr:*` wildcard dalam elemen utama untuk pernyataan kebijakan repositori Anda. Izin `ecr:*` wildcard dapat berguna saat mengembangkan dan menguji di akun pengembangan yang tidak dapat mengakses data produksi Anda.

Namun, Anda harus memastikan bahwa izin `ecr:*` wildcard tidak digunakan di lingkungan produksi Anda karena dapat menyebabkan kerentanan keamanan yang serius. Pendekatan pola ini membantu Anda mengidentifikasi repositori Amazon ECR yang berisi izin `ecr:*` wildcard dalam pernyataan kebijakan repositori. Pola ini menyediakan langkah-langkah dan CloudFormation template AWS untuk membuat aturan khusus di AWS Config. Fungsi AWS Lambda kemudian memantau pernyataan kebijakan repositori Amazon ECR Anda untuk izin wildcard. `ecr:*` Jika menemukan pernyataan kebijakan repositori yang tidak sesuai, Lambda memberi tahu AWS Config untuk mengirim peristiwa ke Amazon EventBridge dan EventBridge kemudian memulai topik Amazon Simple Notification Service (Amazon SNS). Topik SNS memberi tahu Anda melalui email tentang pernyataan kebijakan repositori yang tidak sesuai.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang ini, lihat Menginstal, memperbarui, dan menghapus instalan AWS CLI dalam dokumentasi AWS CLI.
- Repositori ECR Amazon yang sudah ada dengan pernyataan kebijakan terlampir, diinstal dan dikonfigurasi di lingkungan pengujian Anda. Untuk informasi selengkapnya tentang ini, lihat

Membuat repositori pribadi dan Menyetel pernyataan kebijakan repositori dalam dokumentasi Amazon ECR.

- AWS Config, dikonfigurasi di Wilayah AWS pilihan Anda. Untuk informasi selengkapnya tentang ini, lihat Memulai AWS Config di dokumentasi AWS Config.
- `aws-config-cloudformation.template`File (terlampir), diunduh ke mesin lokal Anda.

Batasan

- Solusi pola ini bersifat Regional dan sumber daya Anda harus dibuat di Wilayah yang sama.

Arsitektur

Diagram berikut menunjukkan cara AWS Config mengevaluasi pernyataan kebijakan repositori Amazon ECR.

Diagram menunjukkan alur kerja berikut:

1. AWS Config memulai aturan khusus.
2. Aturan kustom memanggil fungsi Lambda untuk mengevaluasi kepatuhan pernyataan kebijakan repositori Amazon ECR. Fungsi Lambda kemudian mengidentifikasi pernyataan kebijakan repositori yang tidak sesuai.
3. Fungsi Lambda mengirimkan status ketidakpatuhan ke AWS Config.
4. AWS Config mengirimkan acara ke. EventBridge
5. EventBridge menerbitkan pemberitahuan ketidakpatuhan ke topik SNS.
6. Amazon SNS mengirimkan peringatan email kepada Anda atau pengguna yang berwenang.

Otomatisasi dan skala

Solusi pola ini dapat memantau sejumlah pernyataan kebijakan repositori Amazon ECR, tetapi semua sumber daya yang ingin Anda evaluasi harus dibuat di Wilayah yang sama.

Alat

- AWS CloudFormation — AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Anda dapat mengelola dan menyediakan tumpukan di beberapa akun AWS dan Wilayah AWS.
- AWS Config — AWS Config memberikan tampilan terperinci tentang konfigurasi sumber daya AWS di akun AWS Anda. Ini mencakup bagaimana sumber daya terkait satu sama lain dan bagaimana sumber daya tersebut dikonfigurasi di masa lalu sehingga Anda dapat melihat bagaimana konfigurasi dan hubungan berubah dari waktu ke waktu.
- Amazon ECR — Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola AWS yang aman, terukur, dan andal. Amazon ECR mendukung repositori pribadi dengan izin berbasis sumber daya menggunakan IAM.
- Amazon EventBridge — Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge mengirimkan aliran data real-time dari aplikasi Anda, aplikasi perangkat lunak sebagai layanan (SaaS), dan layanan AWS ke target seperti fungsi AWS Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun lain.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) mengoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien, termasuk server web dan alamat email. Pelanggan menerima semua pesan yang dipublikasikan ke topik tempat mereka berlangganan, dan semua pelanggan ke suatu topik menerima pesan yang sama.

Kode

Kode untuk pola ini tersedia dalam `aws-config-cloudformation.template` file (terlampir).

Epik

Buat CloudFormation tumpukan AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan AWS.	Buat CloudFormation tumpukan AWS dengan menjalankan perintah berikut di AWS CLI: `$ aws cloudformation create-stack --stack-name=AWSConfigECR \ --template-body file://aws-config-cloudformation.template \ --parameters ParameterKey=<email>,ParameterValue=<myemail@example.com> \ --capabilities CAPABILITY_NAMED_IAM`	AWS DevOps

Uji aturan kustom AWS Config

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aturan kustom AWS Config.	1. Masuk ke AWS Management Console, buka konsol AWS Config, lalu pilih Resources. 2. Pada halaman Inventaris sumber daya, Anda dapat memfilter berdasarkan kategori sumber daya, jenis	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber daya, dan status kepatuhan. 3. Repositori Amazon ECR yang berisi `ecr:*` is `NON-COMPLIANT?` dan repositori Amazon ECR yang tidak berisi adalah. `ecr:*` `COMPLIANT` 4. Alamat email yang berlangganan topik SNS menerima pemberitahuan jika repositori Amazon ECR berisi pernyataan kebijakan yang tidak sesuai.	

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Optimalkan penerapan tanpa server multi-akun dengan menggunakan alur kerja dan Tindakan AWS CDK GitHub

Sarat Chandra Pothula dan VAMSI KRISHNA SUNKAVALLI, Amazon Web Services

Ringkasan

Organizations yang menerapkan infrastruktur tanpa server di berbagai lingkungan Akun AWS dan sering menghadapi tantangan seperti duplikasi kode, proses manual, dan praktik yang tidak konsisten. Solusi pola ini menunjukkan cara menggunakan alur kerja AWS Cloud Development Kit (AWS CDK) dalam Go and GitHub Actions yang dapat digunakan kembali untuk merampingkan manajemen infrastruktur tanpa server multi-akun. Solusi ini menunjukkan bagaimana Anda dapat mendefinisikan sumber daya cloud sebagai kode, menerapkan proses integration/continuous penerapan berkelanjutan standar (CI/CD), dan membuat komponen modular yang dapat digunakan kembali.

Dengan menggunakan alat ini, organisasi dapat mengelola sumber daya lintas akun secara efisien, menerapkan pipeline penyebaran yang konsisten, dan menyederhanakan arsitektur tanpa server yang kompleks. Pendekatan ini juga meningkatkan keamanan dan kepatuhan dengan menegakkan praktik standar untuk digunakan Akun AWS, pada akhirnya meningkatkan produktivitas dan mengurangi kesalahan dalam pengembangan dan penerapan aplikasi tanpa server.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Identity and Access Management (IAM) peran dan izin ada untuk proses penerapan. Ini termasuk izin untuk mengakses repositori Amazon Elastic Container Registry (Amazon ECR) Registry, membuat AWS Lambda fungsi, dan sumber daya lain yang diperlukan di seluruh target. Akun AWS
- AWS Command Line Interface (AWS CLI) versi 2.9.11 atau yang lebih baru, diinstal dan dikonfigurasi.
- AWS Cloud Development Kit (AWS CDK) versi 2.114.1 atau yang lebih baru, diinstal dan di-bootstrap.
- Go 1.22 atau yang lebih baru, diinstal.
- Docker 24.0.6 atau yang lebih baru, diinstal.

Batasan

- Kompatibilitas bahasa — Go adalah bahasa populer untuk aplikasi tanpa server. Namun, selain Go, AWS CDK mendukung bahasa pemrograman lainnya, termasuk C #, Java, Python, dan. TypeScript Jika organisasi Anda memiliki basis kode atau keahlian dalam bahasa lain, Anda mungkin perlu mengadaptasi atau mempelajari Go untuk sepenuhnya menggunakan solusi yang dijelaskan dalam pola.
- Kurva belajar — Mengadopsi alur kerja AWS CDK, Go (jika baru bagi organisasi), dan alur kerja yang GitHub dapat digunakan kembali mungkin melibatkan kurva pembelajaran untuk pengembang dan tim. DevOps Pelatihan dan dokumentasi mungkin diperlukan untuk memastikan kelancaran adopsi dan penggunaan teknologi ini secara efektif.

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Solusi ini melakukan langkah-langkah berikut:

1. Pengembang mengkloning repositori, membuat cabang baru, dan membuat perubahan pada kode aplikasi di lingkungan lokal mereka.
2. Pengembang melakukan perubahan ini dan mendorong cabang baru ke repositori. GitHub
3. Pengembang membuat permintaan tarik di GitHub repositori, mengusulkan untuk menggabungkan fitur atau cabang fitur baru mereka ke cabang utama.
4. Permintaan tarik ini memicu alur kerja GitHub Tindakan integrasi berkelanjutan (CI). CI dan alur kerja penerapan berkelanjutan (CD) dalam pola ini menggunakan alur kerja yang dapat digunakan kembali, yang merupakan templat modular yang telah ditentukan sebelumnya yang dapat dibagikan dan dieksekusi di berbagai proyek atau repositori. Alur kerja yang dapat digunakan kembali mempromosikan standardisasi dan efisiensi dalam proses. CI/CD
5. Alur kerja CI menyiapkan lingkungan yang diperlukan, menghasilkan tag Docker untuk gambar, dan membangun image Docker menggunakan kode aplikasi.
6. Alur kerja CI mengautentikasi AWS dengan menggunakan peran Akun AWS GitHub OIDC pusat. Untuk alur kerja CI, peran Akun AWS GitHub OIDC pusat menggunakan AWS Security Token Service (AWS STS) untuk mendapatkan kredensil sementara. Kredensi ini memungkinkan peran untuk membangun dan mendorong gambar Docker ke repositori Amazon ECR pusat. Akun AWS
7. Alur kerja CI mendorong image Docker yang dibangun ke Amazon ECR.

8. Alur kerja CI menyimpan tag gambar ke Systems Manager Parameter Store.
9. Setelah alur kerja CI berhasil diselesaikan, tag gambar Docker akan dikeluarkan.
10. Saat memicu alur kerja CD, pengembang secara manual memasukkan tag gambar gambar Docker yang ingin mereka terapkan. Tag gambar ini sesuai dengan tag yang dihasilkan dan didorong ke Amazon ECR selama alur kerja CI.
11. Pengembang secara manual memicu alur kerja CD, yang menggunakan alur kerja CD yang dapat digunakan kembali.
12. Alur kerja CD mengautentikasi dengan AWS menggunakan peran Akun AWS GitHub OIDC pusat. Untuk alur kerja CD, pertama kali AWS STS digunakan untuk mengasumsikan peran Akun AWS GitHub OIDC pusat. Kemudian, peran ini mengasumsikan peran bootstrap CDK untuk penerapan akun target.
13. Alur kerja CD menggunakan AWS CDK untuk mensintesis template AWS CloudFormation .
14. Alur kerja CD menyebarkan aplikasi ke target Akun AWS dengan menggunakan penyebaran CDK, menggunakan tag gambar yang ditentukan secara manual untuk fungsi Lambda.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS CloudFormation merupakan bagian integral dari proses AWS CDK penyebaran. CDK mensintesis CloudFormation template dan kemudian digunakan CloudFormation untuk membuat atau memperbarui sumber daya di lingkungan. AWS
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

- AWS Systems Manager Parameter Store menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- GitHub Actions adalah platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang terintegrasi erat dengan GitHub repositori. Anda dapat menggunakan GitHub Actions untuk mengotomatiskan pipeline build, test, dan deployment.
- Go adalah bahasa pemrograman open source yang didukung Google.

Repositori kode

Kode untuk pola ini tersedia di cicd-github-actions repositori GitHub aws-cdk-golang-serverless-.

Praktik terbaik

- Desain modular - Atur AWS CDK kode Anda ke dalam konstruksi atau tumpukan modular dan dapat digunakan kembali, mempromosikan penggunaan kembali kode dan pemeliharaan di beberapa akun dan proyek.
- Pemisahan kekhawatiran — Pisahkan kode infrastruktur dari kode aplikasi, memungkinkan penyebaran dan pengelolaan independen setiap komponen.
- Pembuatan versi dan kekekalan — Perlakukan infrastruktur Anda sebagai kode (IAc), dan gunakan Git untuk kontrol versi. Merangkul prinsip-prinsip infrastruktur yang tidak dapat diubah dengan menciptakan sumber daya baru alih-alih memodifikasi yang sudah ada.
- Pengujian dan validasi — Menerapkan strategi pengujian komprehensif, termasuk pengujian unit, pengujian integrasi, dan end-to-end pengujian, untuk membantu mendukung kebenaran dan keandalan AWS CDK kode dan penerapan Anda.
- Keamanan dan kepatuhan — Ikuti praktik terbaik AWS keamanan, seperti akses hak istimewa, komunikasi aman, dan enkripsi data. Menerapkan pemeriksaan kepatuhan dan mekanisme audit untuk memastikan kepatuhan terhadap kebijakan organisasi dan persyaratan peraturan. Menerapkan praktik terbaik keamanan untuk gambar kontainer, seperti memindai kerentanan, menegakkan penandatanganan gambar, dan mematuhi persyaratan kepatuhan untuk organisasi Anda.

- Pemantauan dan pencatatan - Siapkan mekanisme pemantauan dan pencatatan untuk melacak kesehatan dan kinerja aplikasi dan infrastruktur tanpa server Anda. Gunakan Layanan AWS seperti Amazon CloudWatch, AWS CloudTrail, dan AWS X-Ray untuk tujuan pemantauan dan audit.
- Otomatisasi dan CI/CD — Gunakan alur kerja yang GitHub dapat digunakan kembali dan CI/CD alat lain untuk mengotomatiskan proses pembuatan, pengujian, dan penerapan, yang dapat membantu mendukung penerapan yang konsisten dan berulang di beberapa akun.
- Manajemen lingkungan - Mempertahankan lingkungan yang terpisah (misalnya, pengembangan, pementasan, dan produksi). Menerapkan strategi untuk mempromosikan perubahan antar lingkungan, memastikan pengujian dan validasi yang tepat sebelum penerapan produksi.
- Dokumentasi dan kolaborasi — Dokumentasikan kode infrastruktur, proses penyebaran, dan praktik terbaik untuk memfasilitasi berbagi pengetahuan dan kolaborasi dalam tim Anda.
- Optimalisasi biaya — Menerapkan strategi pemantauan dan pengoptimalan biaya, seperti rightsizing sumber daya, memanfaatkan auto-scaling, dan memanfaatkan layanan pengoptimalan biaya seperti dan. AWS AWS Budgets AWS Cost Explorer
- Pemulihan dan pencadangan bencana - Rencanakan skenario pemulihan bencana dengan menerapkan mekanisme pencadangan dan pemulihan untuk aplikasi dan sumber daya infrastruktur tanpa server Anda.
- Peningkatan berkelanjutan - Tinjau secara teratur dan perbarui praktik, alat, dan proses Anda agar selaras dengan praktik terbaik terbaru, rekomendasi keamanan, dan kemajuan teknologi dalam ekosistem tanpa server.
- Tingkatkan postur keamanan — Gunakan AWS PrivateLinkuntuk meningkatkan postur keamanan virtual private cloud (VPC) Anda dengan mengonfigurasi titik akhir VPC antarmuka untuk Amazon AWS Lambda ECR,, dan Parameter Store. AWS Systems Manager

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR di pusat. Akun AWS	Untuk berbagi gambar kontainer di beberapa Akun AWS, Anda harus mengonfig urasi akses lintas akun untuk Amazon ECR. Pertama, buat	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	repositori Amazon ECR di pusat. Akun AWS Untuk membuat repositor i Amazon ECR, jalankan perintah berikut: `aws ecr create-re pository --repository- name sample-repo` Dalam tugas selanjutnya, berikan akses tarik ke yang lain Akun AWS yang perlu menggunakan gambar kontainer.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan izin lintas akun ke repositori Amazon ECR.	Untuk menambahkan izin lintas akun ke repositori Amazon ECR di pusat Akun AWS, jalankan kode berikut: `{ "Version": "2008-10-17", "Statement": [{ "Sid": "LambdaECRImageRetrievalPolicy", "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": ["ecr:BatchGetImage", "ecr:GetDownloadUrlForLayer",], "Condition": { "StringLike": { "aws:sourceArn": "arn:aws:lambda:<Target_Region>:<Target_Account_ID>:function:*" } } }, { "Sid": "new statement", "Effect": "Allow", "Principal": {`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"AWS": "arn:aws:iam::<Target_Account_ID>:root"
	},
	"Action": [
	"ecr:BatchGetImage",
	"ecr:GetDownloadUrlForLayer",
	],
	}
	]
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan peran untuk peran GitHub OIDC di pusat. Akun AWS	1. Konfigurasikan AWS ke GitHub OIDC trust sebagai identitas federasi, yang mencakup penambaha n penyedia GitHub OIDC ke AWS dan mengonfig urasi kebijakan peran dan kepercayaan di IAM. Untuk melakukannya, ikuti petunjuk dalam Mengonfig urasi OpenID Connect di Amazon Web Services GitHub dalam dokumentasi. 2. Setelah peran dibuat, tambahkan izin yang diperlukan ke peran. Misalnya, tambahkan izin untuk Amazon ECR dan AWS Systems Manager Parameter Store. Untuk informasi selengkapnya, lihat Mengonfigurasi peran untuk penyedia identitas GitHub OIDC dalam dokumentasi IAM.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap AWS lingkungan di target Akun AWS.	Siapkan lingkungan CDK secara spesifik Akun AWS dan Wilayah AWS yang memungkinkan penyebaran lintas akun dari akun pusat dan menerapkan prinsip hak istimewa paling sedikit ke peran eksekusi. CloudFormation Untuk bootstrap AWS lingkungan, jalankan perintah berikut: `cdk bootstrap aws://<Target_Account_ID>/<Target_Region> --trust <Central_Account_ID> --cloudformation-execution-policies arn:aws:iam::aws:policy/<Least_Privilege_Policy>`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan akses peran Akun AWS OIDC pusat ke peran Akun AWS bootstrap target.	Bootstrap CDK membuat peran IAM berikut yang dirancang untuk diasumsikan oleh pusat Akun AWS selama berbagai tahap proses penyebaran CDK: • Peran penerbitan file • Peran penerbitan gambar • Peran pencarian • Peran penyebaran Setiap peran memiliki izin khusus yang disesuaikan dengan tujuannya, mengikuti prinsip hak istimewa paling sedikit. `Target_Account_ID` Dan `Target_Region` di setiap nama peran membantu menunjukkan bahwa peran ini unik di berbagai wilayah Akun AWS dan wilayah. Pendekatan ini mendukung identifikasi dan manajemen yang jelas dalam pengaturan multi-akun, Multi-wilayah. `Target Account CDK Bootstrap Roles arn:aws:iam::<Target_Account_ID>:role/cdk-deploy-role-<Target_Account_ID>-<Target_Region>`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>arn:aws:iam::<Target_
	Account_ID>:role/cdk-
	file-publishing-role-
	<Target_Account_ID>-<
	Target_Region>
	arn:aws:iam::<Target_
	Account_ID>:role/cdk-
	image-publishing-role-
	<Target_Account_ID>-
	<Target_Region>
	arn:aws:iam::<Target
	_Account_ID>:role/
	cdk-lookup-role-<T
	arget_Account_ID>-
	<Target_Region></pre>
	• Perbarui kebijakan izin untuk peran OIDC di akun pusat untuk memberiny a kemampuan untuk mengambil peran di akun target. Konfigurasi ini memungkinkan penyebara n tumpukan CDK di berbagai tempat. Akun AWS Dengan mengizink an peran OIDC akun pusat untuk mengadopsi izin yang diperlukan dari akun target, Anda membuat jembatan aman untuk penyebaran CDK lintas akun. Pendekata n ini mempertahankan kontrol akses yang tepat sambil memfasilitasi manajemen infrastruktur multi-akun yang mulus.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk memperbarui kebijakan izin untuk peran OIDC di pusat Akun AWS, gunakan kode berikut: `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": ["arn:aws:iam::<Target_Account_ID>:role/cdk-deploy-role-<Target_Account_ID>-<Target_Region>", "arn:aws:iam::<Target_Account_ID>:role/cdk-file-publishing-role-<Target_Account_ID>-<Target_Region>", "arn:aws:iam::<Target_Account_ID>:role/cdk-image-publishing-role-<Target_Account_ID>-<Target_Region>", "arn:aws:iam::<Target_Account_ID>:role/cdk-lookup-role-<Target_Account_ID>-<Target_Region>"]`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre> }] }</pre>	

Bangun gambar Docker

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori proyek.	Untuk mengkloning GitHub repositori pola ini, jalankan perintah berikut: <pre>git clone https://github.com/aws-samples/aws-cdk-golang-serverless-cicd-github-actions.git</pre>	AWS DevOps
Pergi ke jalur Dockerfile.	Untuk menavigasi ke jalur Dockerfile, jalankan perintah berikut: <pre>cd lambda</pre>	AWS DevOps
Otentikasi Docker dengan Amazon ECR.	Amazon ECR memerlukan akses aman ke repositori kontainer pribadi Anda. Dengan masuk dengan cara ini, Anda mengizinkan Docker di mesin atau CI/CD lingkungan lokal Anda untuk berinteraksi dengan Amazon ECR dengan aman.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk mengautentikasi Docker dengan Amazon ECR, jalankan perintah berikut: `aws ecr get-login -password --region <AWS_REGION> | docker login --username AWS --password-stdin <AWS_ACCOUNT_ID>.dkr.ecr.<AWS_REGION>.amazonaws.com` Merevisi placeholder `AWS_REGION` dan `AWS_Account_ID` dengan informasi Anda.	
membuat gambar Docker.	Untuk membangun image Docker, jalankan perintah berikut: `docker build --platform linux/arm64 -t sample-app .`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tandai dan dorong Gambar Docker.	Untuk menandai dan mendorong image Docker ke repositori Amazon ECR, jalankan perintah berikut:
	`docker tag sample-ap p:latest <AWS_ACCO UNT_ID>.dkr.ecr.<A WS_REGION>.amazona ws.com/<ECR_REPOSI TORY>:<DOCKER_TAG>`
	`docker push <AWS_ACCO UNT_ID>.dkr.ecr.<A WS_REGION>.amazona ws.com/<ECR_REPOSI TORY>:<DOCKER_TAG>`
	Merevisi placehold erAWS_Account_ID ,, AWS_REGION ECR_REPOS ITORY , dan DOCKER_TAG dengan informasi Anda.
		AWS DevOps

Menerapkan aplikasi AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sintesis tumpukan CDK dengan variabel khusus lingkungan.	Untuk menghasilkan CloudFormation template untuk infrastruktur Anda seperti yang didefinisikan dalam kode CDK Anda, jalankan perintah berikut:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ENV=<environment> IMAGETAG=<image_tag> ECR_ARN=<ecr_repo_arn> cdk synth` Merevisi placeholder berikut dengan informasi Anda: • `environment` — Diganti dengan nama lingkungan tertentu sepertidev,staging, atauprod. • `image_tag` — Ganti dengan tag tertentu untuk gambar Docker seperti `v1.0.0` ataulatest. • `ecr_repo_arn` — Ganti dengan Nama Sumber Daya Amazon (ARN) dari repositori Amazon ECR.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan tumpukan CDK.	Untuk menyebarkan tumpukan CDK ke Anda Akun AWS, jalankan perintah berikut. `--require-approval never`Bendera berarti bahwa CDK akan secara otomatis menyetujui dan mengeksek usi semua perubahan. Ini termasuk perubahan yang biasanya ditandai CDK sebagai membutuhkan tinjauan manual (seperti perubahan kebijakan IAM atau penghapusan sumber daya). Pastikan kode CDK dan CI/CD pipeline Anda telah teruji dengan baik dan aman sebelum menggunak an `--require-approval never` flag di lingkungan produksi. `ENV=<environment> IMAGETAG=<image_ta g> ECR_ARN=<ecr_repo_ arn> cdk deploy -- require-approval never`	AWS DevOps

Otomatiskan CI/CD menggunakan alur kerja GitHub Tindakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cabang fitur, dan tambahkan perubahan Anda.	Gunakan repositori kloning yang Anda buat sebelumny	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	a, buat cabang fitur, lalu tambahkan perubahan Anda ke kode aplikasi. Gunakan salah satu perintah berikut ini:
	`git checkout -b <feature_branch>`
	`git add .`
	`git commit -m "add your changes"`
	`git push origin <feature_branch>`
	Berikut ini adalah contoh perubahan:
	• Perubahan pada logika fungsi Lambda
	• Menambahkan fitur atau fungsionalitas baru ke kode Lambda
	• Memperbaiki bug atau mengoptimalkan kode yang ada dalam fungsi Lambda
	GitHub Tindakan akan menggunakan alur kerja yang dapat digunakan kembali dan memicu saluran pipa. CI/CD	

Gabungkan perubahan Anda.	Buat permintaan tarik, dan gabungkan perubahan Anda ke main.	AWS DevOps

Pemecahan masalah

Isu	Solusi
`AccessDenied` kesalahan saat menerapkan sumber daya di seluruh Akun AWS, misalnya, `AccessDenied: User not authorized to perform: "sts:AssumeRole"` .	Untuk membantu mengatasi masalah ini, lakukan hal berikut untuk memverifikasi izin lintas akun: • Pastikan bahwa peran dan kebijakan IAM yang diperlukan tersedia untuk penerapan lintas akun. • Periksa apakah izin `assume` peran dikonfigurasi dengan benar.
Masalah kompatibilitas karena ketidakcocokan versi, misalnya, `undefined: awscdkStack` kesalahan dengan versi CDK yang sudah ketinggalan zaman.	Untuk membantu mengatasi masalah ini, lakukan hal berikut untuk memverifikasi bahwa Anda menggunakan versi AWS CDK dan Go yang diperlukan: • Pastikan Anda menggunakan versi yang kompatibel dari AWS CDK dan Go. • Periksa masalah yang diketahui atau melanggar perubahan dalam versi terbaru.
Kegagalan pipa CI/CD, misalnya, `Error: No such file or directory` karena konfigurasi YAMB yang salah atau `Permission denied` untuk cabang yang dilindungi.	Untuk membantu menyelesaikan masalah dengan konfigurasi GitHub Tindakan, verifikasi bahwa alur kerja yang dapat digunakan kembali direferensikan dan dikonfigurasi dengan benar.

Sumber daya terkait

Sumber daya AWS

- AWS Praktik Terbaik untuk Keamanan, Identitas, & Kepatuhan
- AWS CDK Lokakarya
- AWS Perpustakaan Cloud Development Kit
- Buat fungsi Lambda menggunakan gambar kontainer

- Identity and Access Management untuk Amazon Elastic Container Registry
- Bekerja dengan AWS CDK di Go

Sumber daya lainnya

- Mengkonfigurasi OpenID Connect di Amazon Web GitHub Services (dokumentasi)
- Dokumentasi Golang
- Quickstart untuk GitHub Tindakan (GitHub dokumentasi)
- Menggunakan kembali alur kerja (dokumentasi) GitHub

Menyediakan AWS Service Catalog produk berdasarkan AWS CloudFormation template dengan menggunakan GitHub Actions

Ashish Bhatt dan Ruchika Modi, Amazon Web Services

Ringkasan

Pola ini memberi organisasi pendekatan yang efisien menggunakan AWS Service Catalogproduk dan portofolio untuk penyediaan standar dan kepatuhan di seluruh tim. Layanan AWS AWS CloudFormationmembantu menggabungkan komponen penting dalam produk dan portofolio Service Catalog untuk penyediaan infrastruktur jaringan dasar. AWS CloudPola ini juga mempromosikan DevOps praktik dengan mengintegrasikan infrastruktur sebagai kode (IAc) ke dalam alur kerja pengembangan otomatis dengan menggunakan Tindakan. GitHub

AWS Service Catalog memungkinkan organisasi untuk membuat dan mengelola layanan TI yang disetujui AWS, menawarkan manfaat seperti standardisasi, kontrol terpusat, penyediaan layanan mandiri, dan manajemen biaya. Dengan mengotomatiskan penyebaran portofolio dan produk Service Catalog melalui GitHub Actions, perusahaan dapat melakukan hal berikut:

- Mencapai penerapan yang konsisten dan berulang.
- Gunakan kontrol versi untuk IAc.
- Integrasikan manajemen sumber daya cloud dengan alur kerja pengembangan yang ada.

Kombinasi ini merampingkan operasi cloud, menegakkan kepatuhan, dan mempercepat pengiriman layanan yang disetujui sekaligus mengurangi kesalahan manual dan meningkatkan efisiensi secara keseluruhan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Akses ke GitHub repositori
- Pemahaman dasar AWS CloudFormation dan AWS Service Catalog
- Bucket Amazon Simple Storage Service (Amazon S3) untuk meng-host template CloudFormation
- Peran AWS Identity and Access Management (IAM) bernama `github-actions` yang digunakan untuk konektivitas antara dan GitHub AWS

Keterbatasan

- Kode pola yang dapat digunakan kembali ini telah diuji hanya dengan GitHub Tindakan.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

Solusi pola ini dibuat dengan menggunakan tindakan GitHub Marketplace berikut dan versinya masing-masing:

- `actions/checkout@v4`
- `aws-actions/configure-aws-credentials@v2`
- `aws-actions/aws-cloudformation-github-deploy@v1.2.0`

Arsitektur

Diagram berikut menunjukkan arsitektur untuk solusi ini.

1. Administrator atau insinyur platform mendorong CloudFormation templat standar ke GitHub repositori, tempat templat dipertahankan. GitHub Repo juga berisi alur kerja yang mengotomatiskan penyediaan menggunakan Tindakan. AWS Service Catalog GitHub
2. GitHub Tindakan memicu alur kerja yang terhubung ke penyedia AWS Cloud OpenID Connect (OIDC) untuk menyediakan Service Catalog.
3. Service Catalog berisi portofolio dan produk yang dapat langsung digunakan pengembang untuk menyediakan sumber daya standar AWS . Pola ini menggabungkan AWS sumber daya seperti virtual private cloud (VPCs), subnet, NAT dan gateway internet, dan tabel rute.
4. Setelah pengembang membuat produk Service Catalog, Service Catalog mengubahnya menjadi sumber daya yang telah dikonfigurasi sebelumnya dan terstandarisasi AWS . Akibatnya, pengembang menghemat waktu karena mereka tidak perlu menyediakan sumber daya individu dan mengonfigurasinya secara manual.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS Ini adalah layanan infrastruktur sebagai kode (IAc) yang dapat dengan mudah digunakan sebagai salah satu jenis produk dengan AWS Service Catalog.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Service Catalogmembantu Anda mengelola katalog layanan TI secara terpusat yang disetujui. AWS Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Lainnya

- GitHub Actions adalah platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang terintegrasi erat dengan GitHub repositori. Anda dapat menggunakan GitHub Actions untuk mengotomatiskan pipeline build, test, dan deployment.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub service-catalog-with-github-actions. Repo berisi file-file menarik berikut:

- `github/workflows`:
  - `e2e-test.yaml`— Panggilan file ini`workflow.yaml`, yang merupakan alur kerja yang dapat digunakan kembali. Alur kerja ini dipicu segera setelah ada komit dan push pada cabang.
  - `workflow.yaml`— File ini berisi alur kerja yang dapat digunakan kembali untuk solusi ini dan dikonfigurasi `workflow_call` sebagai pemicunya. Sebagai alur kerja yang dapat digunakan kembali, `workflow.yaml` dapat dipanggil dari alur kerja lainnya.
- `templates`:

- `servicecatalog-portfolio.yaml`— CloudFormation Template ini mencakup sumber daya yang menyediakan portofolio Service Catalog dan produk Service Catalog. Template berisi serangkaian parameter yang digunakan saat menyediakan portofolio dan produk Service Catalog. Satu parameter menerima URL file Amazon S3 tempat `vpc.yaml` template diunggah. Meskipun pola ini menyertakan `vpc.yaml` file untuk menyediakan AWS sumber daya, Anda juga dapat menggunakan URL file parameter S3 untuk konfigurasi.
- `vpc.yaml`— CloudFormation Template ini berisi AWS sumber daya yang akan ditambahkan dalam produk Service Catalog. AWS sumber daya termasuk VPCs, subnet, gateway internet, gateway NAT, dan tabel rute. `vpc.yaml`Template adalah contoh bagaimana Anda dapat menggunakan CloudFormation template apa pun dengan produk Service Catalog dan template portofolio.

Praktik terbaik

- Lihat Praktik Terbaik Keamanan untuk AWS Service Catalog dalam AWS Service Catalog dokumentasi.
- Lihat Pengerasan keamanan untuk GitHub Tindakan dalam GitHub dokumentasi.

Epik

Siapkan workstation lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Git di workstation lokal Anda.	Untuk menginstal dan mengkonfigurasi Git di workstation lokal Anda, gunakan instruksi Memulai — Menginstal Git dalam dokumentasi Git.	Pengembang aplikasi
Kloning repo GitHub proyek.	Untuk mengkloning repo GitHub proyek, lakukan hal berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	1. Buka GitHub repo untuk pola ini
	2. Pilih Kode untuk melihat opsi kloning, dan salin URL yang disediakan di tab HTTPS.
	3. Buat folder untuk proyek Anda di workstation Anda.
	4. Buka terminal, dan arahkan ke folder ini
	5. Untuk mengkloning GitHub repo, jalankan perintah berikut menggunakan URL yang Anda salin di langkah 2:
	`git clone https://github.com/aws-samples/service-catalog-with-github-actions.git`
	6. Saat kloning selesai, untuk mengubah repo kloning di folder proyek Anda, jalankan perintah berikut:
	`cd <folder-name>/service-catalog-with-github-actions`
	7. Buka proyek di lingkungan pengembangan terintegrasi (IDE) pilihan Anda.	

Siapkan penyedia OIDC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan penyedia OIDC.	Buat penyedia OpenID Connect (OIDC) yang memungkinkan alur kerja GitHub Actions mengakses sumber daya AWS, tanpa perlu menyimpan kredensia lnya sebagai rahasia yang berumur panjang. AWS GitHub Untuk petunjuk, lihat Mengonfigurasi OpenID Connect di Amazon Web Services GitHub dalam dokumentasi. Setelah penyedia OIDC dikonfigurasi, kebijakan kepercayaan peran IAM`github-actions` , yang disebutkan sebelumnya dalam Prasyarat, akan diperbarui.	Administrator AWS, AWS DevOps, AWS Umum

Pipeline Trigger GitHub Actions untuk menyebarkan portofolio dan produk Service Catalog

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbaruie`e2e-test.yaml` .	`e2e-test.yaml` File memicu alur kerja yang dapat digunakan kembali di. `workflow.yaml` Perbarui dan validasi nilai untuk parameter input berikut die`e2e-test.yaml` :	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• `aws_account_id` — Tentukan yang benar Akun AWS.
	• `aws_region` — Tentukan yang benar Wilayah AWS.
	• `s3BucketName` — Tentukan bucket Amazon S3 untuk menampung template. CloudFormation
	• File alur kerja membutuhkan dua peran IAM sebagai input:
	  • `LaunchConstraintRole` - Peran IAM yang AWS Service Catalog mengasumsikan ketika pengguna akhir meluncurkan, memperbarui, atau mengakhiri suatu produk.
	  • `PrincipalArn` - Nama Sumber Daya Amazon (ARN) dari prinsipal (pengguna, peran, atau grup IAM) yang akan dikaitkan dengan Portofolio Service Catalog. Jika ya`IAM`, nilai yang didukung adalah Nama Sumber Daya Amazon IAM (ARN) yang ditentukan sepenuhnya. `PrincipalType` Jika ya`IAM_PATTERN` ,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nilai yang didukung adalah ARN IAM tanpa dalam `AccountID` format berikut: `PrincipalType arn:partition:iam::resource-type/resource-id`	

Validasi penerapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi sumber daya Service Catalog.	Untuk memvalidasi resource Service Catalog, lakukan hal berikut: 1. Masuk ke bagian Konsol Manajemen AWS dari Akun AWS, dan verifikasi bahwa Wilayah AWS sudah benar. 2. Arahkan ke AWS Service Catalogdan validasi bahwa portofolio hadir di bawah Administrasi, Portofolio. 3. Pilih portofolio, dan validasi informasi pada tab Produk, Kendala, dan Akses.	AWS DevOps

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus CloudFormation tumpukan.	Untuk menghapus CloudFormation tumpukan, lakukan hal berikut: 1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation. 2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS tempat tumpukan berada. 3. Pada halaman Stacks, pilih tumpukan yang ingin Anda hapus. Tumpukan pasti sedang berjalan. 4. Di panel detail tumpukan, pilih Hapus. 5. Pilih Hapus tumpukan saat diminta. Untuk informasi selengkapnya, lihat Menghapus tumpukan dari CloudFormation konsol di CloudFormation dokumentasi	DevOps insinyur, administrator AWS

Pemecahan masalah

Isu	Solusi

`e2e-test`
 	
`Can't find 'action.yml', 'action.yaml' or 'Dockerfile' under '*/home/runner/work/service-catalog-with-github-actions/service-catalog-with-github-actions`
 	
`Did you forget to run actions/checkout before running your local action?`
	Untuk memastikan bahwa Anda mengaktifkan pengaturan repositori yang benar, lakukan hal berikut:
	
	1. Arahkan ke repositori Github, tab Pengaturan.
	2. Pilih Tindakan, Umum dari menu di sisi kiri
	3. Buka bagian Akses, dan pilih opsi Dapat diakses dari repositori di organisasi 'XXX'.

Sumber daya terkait

AWS dokumentasi

- Ikhtisar Service Catalog

Sumber daya lainnya

- Tentang peristiwa yang memicu alur kerja (GitHub dokumentasi)
- Menggunakan kembali alur kerja (dokumentasi) GitHub

Informasi tambahan

Untuk melihat tangkapan layar yang terkait dengan Epik, buka folder Gambar di repo pola ini. GitHub Tangkapan layar berikut tersedia:

- AWS Service Catalog portofolio, bagian Administrasi
- AWS Service Catalog produk, bagian Administrasi
- AWS Service Catalog produk, User/Provisioning bagian

Menyediakan peran IAM dengan hak istimewa paling sedikit dengan menerapkan solusi mesin penjual otomatis peran

Benjamin Morris, Nima Fotouhi, Aman Kaur Gandhi, dan Cad Moon, Amazon Web Services

Ringkasan

Izin peran over-scoped AWS Identity and Access Management (IAM) untuk jaringan pipa dapat menimbulkan risiko yang tidak perlu bagi organisasi. Pengembang terkadang memberikan izin luas selama pengembangan tetapi mengabaikan untuk mengurangi izin setelah memecahkan masalah kode mereka. Hal ini menyebabkan masalah di mana peran yang kuat hadir tanpa kebutuhan bisnis dan mungkin tidak pernah ditinjau oleh insinyur keamanan.

Pola ini menawarkan solusi untuk masalah ini: mesin penjual otomatis peran (RVM). Menggunakan model penyebaran yang aman dan terpusat, RVM mendemonstrasikan cara menyediakan peran IAM dengan hak istimewa paling sedikit untuk saluran pipa repositori individu dengan sedikit usaha dari pengembang. GitHub Karena RVM adalah solusi utama, Anda dapat mengonfigurasi tim keamanan Anda sebagai pengulas yang diperlukan untuk menyetujui perubahan. Pendekatan ini memungkinkan keamanan untuk menolak permintaan peran pipeline yang memiliki izin berlebih.

RVM mengambil kode Terraform sebagai input dan menghasilkan peran IAM siap pipeline sebagai output. Input yang diperlukan adalah Akun AWS ID, nama GitHub repositori, dan kebijakan izin. RVM menggunakan input ini untuk membuat kebijakan kepercayaan dan kebijakan izin peran. Kebijakan kepercayaan yang dihasilkan memungkinkan GitHub repositori yang ditentukan untuk mengambil peran dan menggunakannya untuk operasi pipa.

RVM menggunakan peran IAM (dikonfigurasi selama bootstrap). Peran ini memiliki izin untuk mengasumsikan a role-provisioning-role di setiap akun di organisasi. Peran dikonfigurasi melalui AWS Control Tower Account Factory for Terraform (AFT) atau. AWS CloudFormation StackSets Itu role-provisioning-roles adalah peran yang benar-benar menciptakan peran pipeline untuk pengembang.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Sebuah GitHub organisasi yang digunakan untuk menyebarkan infrastruktur sebagai kode (IAc) melalui GitHub Actions. (GitHub Enterprise/Premium/Ultimatetidak diperlukan.)

- AWS Lingkungan multi-akun. Lingkungan ini tidak perlu menjadi bagian dari AWS Organizations.
- Mekanisme untuk menerapkan peran IAM di semua Akun AWS (misalnya, AFT atau CloudFormation StackSets).
- Terraform versi 1.3 atau yang terakhir diinstal dan dikonfigurasi.
- AWS Penyedia Terraform versi 4 atau yang lebih baru diinstal dan dikonfigurasi.

Batasan

- Kode pola ini khusus untuk GitHub Actions dan Terraform. Namun, konsep umum pola dapat digunakan kembali dalam kerangka kerja integrasi dan pengiriman berkelanjutan (CI/CD) lainnya.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menggambarkan alur kerja untuk pola ini.

Alur kerja untuk penggunaan tipikal mesin penjual otomatis peran terdiri dari langkah-langkah berikut:

1. Pengembang mendorong kode yang berisi kode Terraform untuk peran IAM yang baru diminta ke repositori RVM. GitHub Tindakan ini memicu pipeline Tindakan RVM GitHub .
2. Pipeline menggunakan kebijakan kepercayaan OpenID Connect (OIDC) untuk mengasumsikan peran asumsi peran RVM.
3. Saat pipeline RVM berjalan, ia mengasumsikan peran alur kerja RVM di akun tempat ia menyediakan peran IAM baru pengembang. (Peran alur kerja RVM disediakan dengan menggunakan AFT atau.) CloudFormation StackSets
4. RVM menciptakan peran IAM pengembang dengan izin dan kepercayaan yang sesuai, sehingga peran tersebut dapat diasumsikan oleh pipeline aplikasi lain.
5. Pengembang aplikasi dapat mengonfigurasi pipeline aplikasi mereka untuk mengambil peran yang disediakan RVM ini.

Peran yang dibuat mencakup izin yang diminta oleh pengembang dan `ReadOnlyAccess` kebijakan. Peran ini hanya dapat diasumsikan oleh pipeline yang berjalan melawan `main` cabang repositori

yang ditentukan pengembang. Pendekatan ini membantu memastikan bahwa perlindungan dan tinjauan cabang dapat diperlukan untuk menggunakan peran tersebut.

Otomatisasi dan skala

Izin hak istimewa paling sedikit memerlukan perhatian terhadap detail untuk setiap peran yang disediakan. Model ini mengurangi kompleksitas yang diperlukan untuk membuat peran ini, memungkinkan pengembang untuk membuat peran yang mereka butuhkan tanpa banyak pembelajaran atau usaha tambahan.

Alat

Layanan AWS

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.

Alat-alat lainnya

- Git adalah sistem kontrol versi terdistribusi open source. Ini termasuk kemampuan untuk membuat akun organisasi.
- GitHub Actions adalah platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang terintegrasi erat dengan GitHub repositori. Anda dapat menggunakan GitHub Actions untuk mengotomatiskan pipeline build, test, dan deployment.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di GitHub role-vending-machinerepositori.

Praktik terbaik

- Buat jalan yang benar mudah dan cara yang salah sulit — Buatlah mudah untuk melakukan hal yang benar. Jika pengembang berjuang dengan proses penyediaan RVM, mereka mungkin

mencoba membuat peran melalui cara lain, yang merusak sifat sentral RVM. Pastikan tim keamanan Anda memberikan panduan yang jelas tentang cara menggunakan RVM dengan aman dan efektif.

Anda juga harus menyulitkan pengembang untuk melakukan hal yang salah. Gunakan kebijakan kontrol layanan (SCPs) atau batas izin untuk membatasi peran apa yang dapat membuat peran lain. Pendekatan ini dapat membantu membatasi pembuatan peran hanya untuk RVM dan sumber tepercaya lainnya.

- Berikan contoh yang baik - Tidak dapat dihindari, beberapa pengembang akan mengadaptasi peran yang ada di repositori RVM sebagai templat informal untuk memberikan izin untuk peran baru mereka. Jika Anda memiliki contoh izin paling sedikit yang dapat disalin, hal itu dapat mengurangi risiko pengembang meminta izin yang luas dan berat wildcard. Jika Anda memulai dengan peran yang sangat diizinkan dengan banyak wildcard, masalah itu dapat berlipat ganda seiring berjalannya waktu.
- Gunakan konvensi dan ketentuan penamaan — Bahkan jika pengembang tidak mengetahui semua nama sumber daya yang akan dibuat aplikasi mereka, mereka tetap harus membatasi izin peran dengan menggunakan konvensi penamaan. Misalnya, jika mereka membuat bucket Amazon S3, nilai kunci sumber daya mereka mungkin terlihat seperti `arn:aws:s3:::myorg-myapp-dev-*` sehingga peran mereka tidak memiliki izin di luar bucket yang cocok dengan nama itu. Menegakkan konvensi penamaan melalui kebijakan IAM memiliki manfaat tambahan untuk meningkatkan kepatuhan terhadap konvensi penamaan. Peningkatan ini terjadi karena sumber daya yang tidak cocok tidak akan diizinkan untuk dibuat.
- Required pull request (PR) review — Nilai solusi RVM adalah menciptakan lokasi pusat di mana peran pipeline baru dapat ditinjau. Namun, desain ini hanya berguna jika ada pagar pembatas yang membantu memastikan keamanan, kode berkualitas tinggi berkomitmen untuk RVM. Lindungi cabang yang digunakan untuk menyebarkan kode (misalnya,`main`) dari dorongan langsung dan memerlukan persetujuan untuk setiap permintaan gabungan yang menargetkannya.
- Konfigurasikan peran hanya-baca — Secara default, RVM menyediakan `readonly` versi dari setiap peran yang diminta. Peran ini dapat digunakan dalam CI/CD pipeline yang tidak menulis data, seperti alur kerja `terraform plan` pipeline. Pendekatan ini membantu mencegah perubahan yang tidak diinginkan jika alur kerja hanya-baca berperilaku buruk.

  Secara default, `ReadOnlyAccess` kebijakan AWS terkelola dilampirkan ke peran hanya-baca dan peran baca-tulis. Kebijakan ini mengurangi kebutuhan iterasi saat menentukan izin yang diperlukan, tetapi mungkin terlalu permisif untuk beberapa organisasi. Jika mau, Anda dapat menghapus kebijakan dari kode Terraform.

- Berikan izin minimum - Ikuti prinsip hak istimewa paling sedikit dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Siapkan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin repositori sampel ke organisasi Anda GitHub .	Kloning repositori pola ini atau fork repositori ini ke GitHub organisasi Anda sehingga Anda dapat menyesuaikannya dengan kebutuhan Anda. • Jika Anda memilih untuk mengkloning repositori ini, Anda dapat menggunakan perintah berikut: `git clone https://github.com/aws-samples/role-vending-machine` • Jika Anda memilih untuk melakukan fork repositori ke GitHub organisasi Anda, Anda dapat menggunakan perintah berikut: `gh repo fork aws-samples/role-vending-machine --org YOUR_ORGANIZATION_NAME`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan Akun AWS untuk RVM.	Tentukan penyebaran infrastruktur mana yang Akun AWS akan digunakan untuk RVM. Jangan gunakan akun manajemen atau root.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Izinkan jaringan pipa organisasi untuk dibuat PRs.	**Note** Langkah ini hanya diperlukan jika Anda ingin mengizinkan `generate_providers_and_account_vars` alur kerja dibuat PRs. Untuk memungkinkan pipeline organisasi Anda dibuat PRs, gunakan langkah-langkah berikut: 1. Kunjungi `https://github.com/organizations/YOUR_ORG/settings/actions` . 2. Pilih Izinkan GitHub Tindakan untuk membuat dan menyetujui permintaan tarik. Untuk informasi selengkapnya, lihat Mengelola setelan GitHub Tindakan untuk repositori dalam dokumentasi. GitHub	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan izin hanya-baca ke akun RVM.	Buat kebijakan delegasi di akun manajemen Anda yang memberikan izin hanya-baca akun RVM Anda. Ini memungkinkan GitHub alur kerja RVM Anda menarik daftar akun AWS organisasi secara dinamis saat skrip berjalan. `generate_providers_and_account_vars.py` Gunakan kode berikut dan ganti `<YOUR RVM Account ID>` dengan Akun AWS ID yang Anda pilih di Langkah 2: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "Statement", "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::<YOUR RVM Account ID>:root" }, "Action": [ "organizations:ListAccounts", "organizations:DescribeOrganization",`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"organizations:DescribeOrganizationalUnit",
	"organizations:ListRoots",
	"organizations:ListAWSServiceAccessForOrganization",
	"organizations:ListDelegatedAdministrators"
	],
	"Resource": "*"
	}
	]
	}</pre>	

Perbarui nilai default dari repo sampel.	Untuk mengonfigurasi RVM agar beroperasi di lingkungan spesifik Anda dan Wilayah AWS, lakukan hal berikut: 1. Perbarui `scripts/generate_providers_and_account_vars.py` dan file lainnya (seperti `bootstrap` folder) dengan Wilayah yang sesuai tempat Anda beroperasi. 2. Perbarui `.github/workflows/.env` file dengan Akun AWS ID, Wilayah, dan variabel lain yang ingin Anda tentukan.	DevOps insinyur

Inisialisasi infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap repo RVM.	Langkah ini diperlukan untuk menciptakan kepercayaan OIDC dan peran IAM yang digunakan oleh pipa RVM itu sendiri, sehingga dapat mulai beroperasi dan menjual peran lain. Dalam konteks akun RVM Anda, jalankan `terraform apply` perintah secara manual dari direktori. `scripts/bootstrap` Berikan nilai yang diperluka n berdasarkan dokumentasi variabel.	DevOps insinyur

Konfigurasikan operasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan `github-wo rkflow-rvm` dan `github- workflow-rvm-readonl y` peran ke semua akun.	Pilih metode penerapan yang selaras dengan praktik organisasi Anda, seperti AFT atau. StackSets Gunakan metode tersebut untuk menerapkan dua peran IAM dalam `scripts/a ssumed_role/main.tf` file (nama default `github-wo rkflow-rvm` dan`github- workflow-rvm-readonl`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	y) ke setiap akun tempat Anda ingin RVM dapat membuat peran pipeline. Peran IAM ini memiliki kebijakan kepercayaan yang memungkinkan peran asumsi peran akun RVM (atau yang setara) untuk mengasums ikannya`readonly`. Peran juga memiliki kebijakan izin IAM yang memungkinkan mereka membaca dan menulis (kecuali menggunakan `readonly` peran) peran yang cocok`github-workflow-role-*` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan `generate_providers_and_account_vars` alur kerja.	Untuk mengonfigurasi RVM Anda sehingga siap untuk membuat peran pipeline, lakukan hal berikut: 1. Jalankan alur kerja dengan menggunakan `generate_providers_and_account_vars` workflow_dispatch. 2. Gabungkan PR yang dibuat alur kerja. Setelah alur kerja selesai, RVM siap untuk: • Terima permintaan untuk peran pipeline baru. • Buat peran read-only atau read-write seperti yang diminta. • Menyebarkan peran ke yang ditentukan Akun AWS.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Saya membuat peran dengan menggunakan RVM, tetapi GitHub tidak dapat mengasumsikan itu.	Verifikasi bahwa nama GitHub repositori cocok dengan nama yang diberikan ke modul. `github_workflow_roles` Peran dicakup sehingga hanya satu repositori yang dapat mengasumsikan mereka.

Isu	Solusi
	Demikian pula, verifikasi bahwa cabang yang digunakan dalam GitHub pipeline cocok dengan nama cabang yang disediakan untuk `github_workflow_roles` modul. Biasanya, peran yang dibuat RVM dengan izin tulis hanya dapat digunakan oleh alur kerja yang dicakup ke `main` cabang (yaitu, penerapan yang bersumber dari). `main`
Peran hanya-baca saya gagal menjalank an pipeline karena tidak memiliki izin untuk membaca sumber daya tertentu.	Meskipun `ReadOnlyAccess` kebijakan menyediakan izin hanya-baca yang luas, kebijakan tersebut tidak memiliki beberapa tindakan baca (misalnya, tindakan tertentu AWS Security Hub CSPM). Anda dapat menambahkan izin tindakan tertentu dengan menggunakan `inline_po licy_readonly` parameter `github-wo rkflow-roles` modul.

Sumber daya terkait

- Praktik terbaik untuk menggunakan AWS CloudFormation StackSets
- Mengatur AWS Lingkungan Anda Menggunakan Beberapa Akun
- Ikhtisar AWS Control Tower Account Factory untuk Terraform (AFT)
- Praktik terbaik kebijakan

Informasi tambahan

Menggunakan GitHub lingkungan

GitHub lingkungan adalah pendekatan alternatif untuk pembatasan berbasis cabang untuk akses peran. Jika Anda lebih suka menggunakan GitHub lingkungan, berikut adalah contoh sintaks untuk kondisi tambahan dalam kebijakan kepercayaan IAM. Sintaks ini menetapkan bahwa peran hanya dapat digunakan ketika GitHub tindakan berjalan di `Production` lingkungan.

```
"StringLike": {
    "token.actions.githubusercontent.com:sub": "repo:octo-org/octo-
repo:environment:Production"
}
```

Contoh sintaks menggunakan nilai placeholder berikut:

- `octo-org`adalah nama GitHub organisasi.
- `octo-repo`adalah nama repositori.
- `Production`adalah nama GitHub lingkungan yang spesifik.

Publikasikan CloudWatch metrik Amazon ke file CSV

Abdullahi Olaoye, Amazon Web Services

Ringkasan

Pola ini menggunakan skrip Python untuk mengambil metrik CloudWatch Amazon dan mengonversi informasi metrik menjadi file nilai yang dipisahkan koma (CSV) untuk meningkatkan keterbacaan. Skrip mengambil layanan AWS yang metriknya harus diambil sebagai argumen yang diperlukan. Anda dapat menentukan profil kredensi AWS Region dan AWS sebagai argumen opsional. Jika Anda tidak menentukan argumen tersebut, skrip menggunakan Region default dan profil yang dikonfigurasi untuk workstation tempat skrip dijalankan. Setelah skrip berjalan, ia menghasilkan dan menyimpan file CSV di direktori yang sama.

Lihat bagian Lampiran untuk skrip dan file terkait yang disediakan dengan pola ini.

Prasyarat dan batasan

Prasyarat

- Python 3.x
- Antarmuka Baris Perintah AWS (AWS CLI)

Batasan

Skrip saat ini mendukung layanan AWS berikut:

- AWS Lambda
- Awan Komputasi Elastis Amazon (Amazon EC2)
  - Secara default, skrip tidak mengumpulkan metrik volume Amazon Elastic Block Store (Amazon EBS). Untuk mengumpulkan metrik Amazon EBS, Anda harus memodifikasi file terlampir`metrics.yaml`.
- Amazon Relational Database Service (Amazon RDS)
  - Namun, skrip tidak mendukung Amazon Aurora.
- Penyeimbang Beban Aplikasi
- Penyeimbang Beban Jaringan
- Amazon API Gateway

Alat

- Amazon CloudWatch adalah layanan pemantauan yang dibangun untuk DevOps insinyur, pengembang, insinyur keandalan situs (SREs), dan manajer TI. CloudWatch menyediakan data dan wawasan yang dapat ditindaklanjuti untuk membantu Anda memantau aplikasi Anda, menanggapi perubahan kinerja di seluruh sistem, mengoptimalkan pemanfaatan sumber daya, dan mendapatkan pandangan terpadu tentang kesehatan operasional. CloudWatch mengumpulkan data pemantauan dan operasional dalam bentuk log, metrik, dan peristiwa, serta memberikan tampilan terpadu tentang sumber daya, aplikasi, dan layanan AWS yang berjalan di AWS dan server lokal.

Epik

Instal dan konfigurasikan prasyarat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal prasyarat.	Jalankan perintah berikut: `$ pip3 install -r requirements.txt`	Developer
Mengonfigurasi AWS CLI.	Jalankan perintah berikut: `$ aws configure`	Developer

Konfigurasikan skrip Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka naskahnya.	Untuk mengubah konfigurasi default skrip, buka`metrics.yaml` .	Developer
Atur periode untuk skrip.	Ini adalah periode waktu untuk mengambil. Periode	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	default adalah 5 menit (300 detik). Anda dapat mengubah periode waktu, tetapi perhatikan batasan berikut: • Jika nilai jam yang Anda tentukan adalah antara 3 jam dan 15 hari yang lalu, gunakan kelipatan 60 detik (1 menit) untuk periode tersebut. • Jika nilai jam yang Anda tentukan adalah antara 15 jam dan 63 hari yang lalu, gunakan kelipatan 300 detik (5 menit) untuk periode tersebut. • Jika nilai jam yang Anda tentukan lebih besar dari 63 hari yang lalu, gunakan kelipatan 3.600 detik (1 jam) untuk periode tersebut. Jika tidak, operasi API tidak akan mengembalikan titik data apa pun.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Atur jam untuk skrip.	Nilai ini menentukan berapa jam metrik yang ingin Anda ambil. Pengaturan default-nya adalah 1 jam. Untuk mengambil beberapa hari metrik, berikan nilai dalam jam. Misalnya, selama 2 hari, tentukan 48.	Developer
Ubah nilai statistik untuk skrip.	(Opsional) Nilai statistik global adalah`Average`, yang digunakan saat mengambil metrik yang tidak memiliki nilai statistik tertentu yang ditetapkan. Skrip mendukung nilai statistik `Maximum`,`SampleCount` , dan`Sum`.	Developer

Jalankan skrip Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan penulisan.	Gunakan perintah berikut ini. `$ python3 cwreport.py <service>` Untuk melihat daftar nilai layanan dan opsional `region` dan `profile` parameter, jalankan perintah berikut:	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`$ python3 cwreport.py -h` Untuk informasi selengkapnya tentang parameter opsional, lihat bagian Informasi tambahan.	

Sumber daya terkait

- Mengkonfigurasi AWS CLI
- Menggunakan CloudWatch metrik Amazon
- CloudWatch Dokumentasi Amazon
- EC2 CloudWatch Metrik-metrik
- Metrik AWS Lambda
- Metrik Amazon RDS
- Metrik Application Load Balancer
- Metrik Network Load Balancer
- Metrik Amazon API Gateway

Informasi tambahan

Penggunaan skrip

```
$ python3 cwreport.py -h
```

Contoh sintaks

```
python3 cwreport.py <service> <--region=Optional Region> <--profile=Optional credential
 profile>
```

Parameter

- layanan (wajib) - Layanan yang ingin Anda jalankan skrip. Skrip saat ini mendukung layanan ini: AWS Lambda, Amazon, EC2 Amazon RDS, Application Load Balancer, Network Load Balancer, dan API Gateway.
- region (opsional) - Wilayah AWS untuk mengambil metrik. Wilayah default adalah`ap-southeast-1`.
- profil (opsional) - Profil bernama AWS CLI untuk digunakan. Jika parameter ini tidak ditentukan, profil kredensi default yang dikonfigurasi akan digunakan.

Contoh

- Untuk menggunakan Region default `ap-southeast-1` dan kredensional yang dikonfigurasi default untuk mengambil metrik Amazon EC2 : `$ python3 cwreport.py ec2`
- Untuk menentukan Wilayah dan mengambil metrik API Gateway: `$ python3 cwreport.py apigateway --region us-east-1`
- Untuk menentukan profil AWS dan mengambil EC2 metrik Amazon: `$ python3 cwreport.py ec2 --profile testprofile`
- Untuk menentukan Wilayah dan profil untuk mengambil EC2 metrik Amazon: `$ python3 cwreport.py ec2 --region us-east-1 --profile testprofile`

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Hapus EC2 entri Amazon di Akun AWS seberang AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi

Dr. Rahul Sharad Gaikwad dan Tamilselvan P, Amazon Web Services

Ringkasan

Active Directory (AD) adalah alat skrip Microsoft yang mengelola informasi domain dan interaksi pengguna dengan layanan jaringan. Ini banyak digunakan di antara penyedia layanan terkelola (MSPs) untuk mengelola kredensi karyawan dan izin akses. Karena penyerang AD dapat menggunakan akun tidak aktif untuk mencoba dan meretas organisasi, penting untuk menemukan akun yang tidak aktif dan menonaktifkannya pada jadwal pemeliharaan rutin. Dengan AWS Directory Service for Microsoft Active Directory, Anda dapat menjalankan Microsoft Active Directory sebagai layanan terkelola. Pola ini dapat membantu Anda mengonfigurasi AWS Lambda otomatisasi untuk menemukan dan menghapus akun yang tidak aktif dengan cepat.

Jika skenario berikut berlaku untuk organisasi Anda, pola ini dapat membantu Anda:

- Manajemen AD terpusat — Jika organisasi Anda memiliki beberapa Akun AWS, masing-masing dengan penerapan AD sendiri, mungkin sulit untuk mengelola akun pengguna dan izin akses secara konsisten di semua akun. Dengan solusi pembersihan AD lintas akun, Anda dapat menonaktifkan atau menghapus akun yang tidak aktif dari semua instans AD secara terpusat.
- Restrukturisasi atau migrasi AD — Jika organisasi Anda berencana untuk merestrukturisasi atau memigrasikan penerapan AD, solusi pembersihan AD lintas akun dapat membantu Anda mempersiapkan lingkungan. Solusinya dapat membantu Anda menghapus akun yang tidak perlu atau tidak aktif, menyederhanakan proses migrasi, dan mengurangi potensi konflik atau masalah.

Bila Anda menggunakan pola ini, Anda bisa mendapatkan manfaat berikut:

- Tingkatkan kinerja database dan server, dan perbaiki kerentanan keamanan Anda dari akun yang tidak aktif.
- Jika server iklan Anda di-host di cloud, menghapus akun yang tidak aktif juga dapat mengurangi biaya penyimpanan sekaligus meningkatkan kinerja. Tagihan bulanan Anda mungkin berkurang karena biaya bandwidth dan sumber daya komputasi keduanya dapat turun.
- Jauhkan penyerang potensial dengan Active Directory yang bersih.

Prasyarat dan batasan

Prasyarat

- Orang tua yang aktif Akun AWS dan satu atau beberapa akun anak. Dalam pola ini, akun induk adalah tempat Active Directory dibuat. Akun anak meng-host server Windows dan bergabung melalui akun induk Active Directory.
- Git diinstal dan dikonfigurasi pada workstation lokal.
- Terraform diinstal dan dikonfigurasi pada workstation lokal.
- AWS Managed Microsoft AD direktori dikonfigurasi di akun induk dan dibagikan ke semua akun anak. Untuk detail selengkapnya, lihat Tutorial: Berbagi AWS Managed Microsoft AD direktori Anda untuk EC2 domain-join yang mulus di Panduan Administrasi.AWS Directory Service
- Koneksi peering cloud pribadi virtual (VPC) atau AWS Transit Gateway koneksi yang tersedia antara VPC AWS Directory Service (akun induk) dan VPC dari instans Amazon Elastic Compute Cloud (Amazon EC2) (akun anak). Untuk detail selengkapnya, lihat Mengkonfigurasi koneksi peering VPC antara pemilik direktori dan akun konsumen direktori di Panduan Administrasi AWS Directory Service .
- Mesin Windows dikonfigurasi dengan `EC2WindowsUserdata` skrip pada semua akun induk dan anak. File skrip tersedia di root repositori kode pola ini.
- Peran lintas akun AWS Identity and Access Management (IAM) tersedia di setiap akun anak yang dikonfigurasi dengan kebijakan kepercayaan untuk mengizinkan penggunaan AWS Lambda fungsi dari akun induk. Untuk informasi selengkapnya, lihat Mengirim dan menerima acara antara Akun AWS EventBridge di Amazon di Panduan EventBridge Pengguna Amazon.
- Nilai rahasia berikut tersedia di AWS Systems Manager Parameter Store dari akun induk:
  - `domainJoinUser`— Nama pengguna layanan direktori
  - `domainJoinPassword`— Kata sandi layanan direktori

  Untuk informasi selengkapnya tentang rahasia, lihat Membuat AWS Secrets Manager rahasia di Panduan AWS Secrets Manager Pengguna.

Batasan

- Membuat sumber daya di akun anak tidak otomatis dengan Terraform. Anda harus membuat sumber daya berikut secara manual dengan menggunakan Konsol Manajemen AWS:
  - EventBridge Aturan Amazon untuk mengirim peristiwa EC2 penghentian Amazon ke akun induk

  - Pembuatan peran EC2 lintas akun Amazon di akun anak dengan kebijakan kepercayaan
  - Pengintip VPC atau koneksi Transit Gateway
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Terraform versi 1.1.9 atau yang lebih baru
- Terraform AWS Provider versi 3.0 atau lebih tinggi

Arsitektur

Diagram berikut menampilkan arsitektur tingkat tinggi dari solusi.

Diagram arsitektur menggambarkan proses berikut:

1. Di akun anak, EventBridge aturan mengumpulkan semua peristiwa EC2 penghentian Amazon. Aturan mengirimkan peristiwa-peristiwa EventBridge yang ada di akun induk.
2. Dari akun induk, EventBridge kumpulkan semua peristiwa dan berisi aturan untuk memicu fungsi `ADcleanup-Lambda` Lambda.
3. Akun induk menerima peristiwa penghentian apa pun dari akun induk atau anak dan memicu fungsi Lambda.
4. Fungsi Lambda membuat panggilan ke grup Amazon Auto EC2 Scaling menggunakan modul boto Python dan mendapatkan ID instance acak. Instance ID digunakan untuk menjalankan perintah Systems Manager.
5. Fungsi Lambda membuat panggilan lain ke Amazon EC2 menggunakan modul boto. Fungsi Lambda mendapatkan alamat IP pribadi dari server Windows yang sedang berjalan dan menyimpan alamat dalam variabel sementara. Pada langkah 5.1 dan 5.2, EC2 instance Windows yang berjalan dikumpulkan dari akun anak.
6. Fungsi Lambda membuat panggilan lain ke Systems Manager untuk mendapatkan informasi komputer yang terhubung. AWS Directory Service

7. AWS Systems Manager Dokumen membantu menjalankan PowerShell perintah di server Amazon EC2 Windows untuk mendapatkan alamat IP pribadi komputer yang terhubung ke AD. (Dokumen Systems Manager menggunakan ID instance yang diperoleh pada langkah 4.)
8. Nama pengguna dan kata sandi domain AD disimpan di AWS Systems Manager Parameter Store. AWS Lambda dan Systems Manager melakukan panggilan ke Parameter Store dan mendapatkan nilai nama pengguna dan kata sandi yang akan digunakan untuk terhubung ke AD.
9. Menggunakan dokumen Systems Manager, PowerShell skrip dijalankan di server Amazon EC2 Windows menggunakan id instance yang diperoleh sebelumnya di langkah 4.
10. Amazon EC2 terhubung AWS Directory Service dengan menggunakan PowerShell perintah dan menghapus komputer yang tidak digunakan atau tidak aktif.

Alat

AWS layanan

- AWS Directory Servicemenyediakan beberapa cara untuk menggunakan Microsoft Active Directory (AD) dengan yang lain Layanan AWS seperti Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS) untuk SQL Server, dan FSx Amazon untuk Windows File Server.
- AWS Directory Service for Microsoft Active Directorymemungkinkan beban kerja dan sumber daya yang sadar direktori Anda AWS untuk menggunakan Microsoft Active Directory di file. AWS Cloud
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya. Dengan IAM, Anda dapat menentukan siapa atau apa yang dapat mengakses layanan dan sumber daya AWS, mengelola izin berbutir halus secara terpusat, dan menganalisis akses untuk menyempurnakan izin. AWS

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar.
- AWS Systems Manager dokumen menentukan tindakan yang dilakukan Systems Manager pada instans terkelola Anda. Systems Manager mencakup lebih dari 100 dokumen pra-konfigurasi yang dapat Anda gunakan dengan menentukan parameter di runtime.
- AWS Systems Manager Parameter Store adalah kemampuan AWS Systems Manager dan menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.

Alat-alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- PowerShelladalah program manajemen otomatisasi dan konfigurasi Microsoft yang berjalan di Windows, Linux, dan macOS.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub aws-lambda-ad-cleanup-terraform-samples.

Praktik terbaik

- Bergabung dengan domain secara otomatis.Saat Anda meluncurkan instance Windows yang akan menjadi bagian dari Directory Service domain, bergabunglah dengan domain selama proses pembuatan instance alih-alih menambahkan instance secara manual nanti. Untuk bergabung dengan domain secara otomatis, pilih direktori yang benar dari daftar dropdown direktori Gabung Domain saat meluncurkan instance baru. Untuk detail selengkapnya, lihat Menggabungkan instans Amazon EC2 Windows dengan mulus ke Direktori AWS Managed Microsoft AD Aktif Anda di Panduan Directory Service Administrasi.

- Hapus akun yang tidak digunakan.Adalah umum untuk menemukan akun di AD yang belum pernah digunakan. Seperti akun yang dinonaktifkan atau tidak aktif yang tetap berada di sistem, akun yang tidak digunakan yang diabaikan dapat memperlambat sistem iklan Anda atau membuat organisasi Anda rentan terhadap pelanggaran data.
- Otomatiskan pembersihan Direktori Aktif.Untuk membantu mengurangi risiko keamanan dan mencegah akun usang memengaruhi kinerja AD, lakukan pembersihan AD harus dilakukan secara berkala. Anda dapat menyelesaikan sebagian besar tugas manajemen dan pembersihan AD dengan menulis skrip. Contoh tugas termasuk menghapus akun yang dinonaktifkan dan tidak aktif, menghapus grup kosong dan tidak aktif, dan menemukan akun pengguna dan kata sandi yang kedaluwarsa.

Epik

Siapkan akun anak

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran lintas akun di akun anak.	Untuk membuat peran lintas akun di akun anak, lakukan hal berikut: 1. Untuk setiap akun anak, buat peran bernama `ec2crossaccountrole` menggunakan kebijakan terkelola bernama`AmazonEC2ReadOnlyAccess` . (Untuk informasi selengkapnya, lihat Membuat peran menggunakan kebijakan kepercayaan khusus dalam dokumentasi IAM.) 2. Di bagian Kebijakan kepercayaan khusus, tambahkan kode berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Principal": {
	"Service": "ec2.amazonaws.com"
	},
	"Action": "sts:AssumeRole"
	},
	{
	"Effect": "Allow",
	"Principal": {
	"AWS": "arn:aws:iam::${Parentaccountid}:role/ADcleanuprole"
	},
	"Action": "sts:AssumeRole"
	}
	]
	}	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan acara di akun anak.	Untuk membuat EventBridge aturan untuk setiap akun anak, lakukan hal berikut: 1. Masuk ke anak Akun AWS, lalu buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/. 2. Di panel navigasi, pilih Aturan. 3. Pilih Buat aturan. 4. Masukkan Nama dan, secara opsional, deskripsi untuk aturan tersebut. 5. Untuk bus Acara, pilih bus acara default AWS. 6. Untuk jenis Aturan, pilih Aturan dengan pola acara. 7. Pilih Selanjutnya. 8. Untuk pola Event, paste kode berikut: `{ "source": ["aws.ec2"], "detail-type": ["EC2 Instance State-change Notification"], "detail": { "state": ["terminated"] } }`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	9. Pilih Berikutnya. 10. Untuk jenis Target, pilih bus Acara di akun atau Wilayah yang berbeda.Untuk bus acara sebagai target, masukkan bus acara Amazon Resource Name (ARN) untuk akun induk. 11. Untuk peran Eksekusi, pilih Buat peran baru untuk sumber daya khusus ini. 12. Pilih Berikutnya untuk meninjau detail aturan baru, lalu pilih Buat. Untuk detail selengkapnya, lihat Membuat aturan yang bereaksi terhadap peristiwa EventBridge di Amazon di Panduan EventBridge Pengguna Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat EC2 instance dan bergabunglah dengan AD.	Untuk membuat EC2 instance untuk Windows, lakukan hal berikut: 1. Gunakan `EC2WindowsUserdata` skrip yang tersedia di repositori kode pola ini. 2. Dalam skrip data pengguna, revisi kode berikut untuk menggunakan `Directory service addresses` nilai Anda dari akun induk: `set-DnsClientServerAddress -InterfaceIndex 6 -ServerAddresses $(Directory service addresses)`	DevOps insinyur

Siapkan workstation lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat folder proyek dan tambahkan file.	Untuk mengkloning repositori dan membuat folder proyek, lakukan hal berikut: 1. Buka GitHub repositori pola ini. 2. Pilih tombol Kode untuk melihat opsi untuk	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	mengkloning di dropdown Clone.
	3. Pada tab HTTPS, salin URL yang disediakan di Clone menggunakan URL web.
	4. Buat folder di mesin Anda, dan beri nama dengan nama proyek Anda.
	5. Buka terminal di mesin lokal Anda, dan arahkan ke folder ini.
	6. Untuk mengkloning repositori git, gunakan perintah berikut.
	`git clone <repository-URL>.git`
	7. Setelah repositori telah dikloning, gunakan perintah berikut untuk pergi ke direktori kloning.
	`cd <directory name>/terraform-aws-lambda-ad-cleanup/multiple-account-cleanup`
	8. Di repositori kloning, buka proyek ini di lingkungan pengembangan terintegrasi (IDE) pilihan Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membangun `adcleanup .zip` file.	Untuk mengompres `lambda_function.py` file, jalankan perintah berikut: `zip -r adcleanup.zip lambda_function.py`	DevOps insinyur

Menyediakan arsitektur target menggunakan konfigurasi Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan nilai untuk variabel Terraform.	Untuk akun anak, berikan nilai untuk `arn` variabel berikut sebagai tipe string dalam `terraform.tfvars` file: • `lambda_env_cross_role_arn` • `child_account_cross_role_arn`	DevOps insinyur
Inisialisasi konfigurasi Terraform.	Untuk menginisialisasi direktori kerja Anda yang berisi file Terraform, jalankan perintah berikut: `terraform init`	DevOps insinyur
Pratinjau perubahan.	Anda dapat melihat pratinjau perubahan yang akan dilakukan Terraform pada infrastruktur sebelum infrastruktur Anda diterapkan. Untuk memvalidasi bahwa Terraform akan membuat perubahan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sesuai kebutuhan, jalankan perintah berikut: `terraform plan –– var-file=examples/terraform.tfvars`	
Jalankan tindakan yang diusulkan.	Untuk memverifikasi bahwa hasil dari `terraform plan` perintah seperti yang diharapkan, lakukan hal berikut: 1. Jalankan perintah berikut: `terraform apply` 2. Masuk ke Konsol Manajemen AWS, dan verifikasi bahwa sumber daya ada.	DevOps insinyur

Verifikasi penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan dan uji fungsi Lambda.	Untuk memverifikasi bahwa penerapan berhasil terjadi, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan buka konsol Lambda. Buka halaman Fungsi, dan pilih nama fungsi yang dimulai dengan ADcleanup-Lambda-*.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pada halaman ikhtisar fungsi, pilih Uji pada tab Kode di bagian Sumber kode. 3. Untuk menyimpan acara pengujian, berikan nama untuk acara tersebut dan pilih Simpan. Untuk menguji acara, pilih Uji lagi. Hasil eksekusi menunjukkan output dari fungsi.	
Lihat hasil eksekusi EventBridge aturan dari akun induk.	Untuk melihat hasil EventBridge aturan yang didasarkan pada peristiwa EC2 penghentian Amazon dari akun induk, lakukan hal berikut: 1. Mengakhiri EC2 instance dari akun induk. 2. Buka konsol Lambda dari akun induk. Buka halaman Fungsi, dan pilih nama fungsi yang dimulai dengan ADcleanup-Lambda-*. 3. Pilih tab Monitor, dan pilih Lihat CloudWatch log. Di CloudWatch konsol, halaman Grup log menunjukkan hasil fungsi Lambda.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat hasil eksekusi EventBridge aturan dari akun anak.	Untuk melihat hasil EventBridge aturan yang didasarkan pada peristiwa EC2 penghentian Amazon dari akun anak, lakukan hal berikut: 1. Mengakhiri EC2 instance dari akun anak. 2. Buka konsol Lambda dari akun induk. Buka halaman Fungsi, dan pilih nama fungsi yang dimulai dengan ADcleanup-Lambda-*. 3. Pilih tab Monitor, dan pilih Lihat CloudWatch log. Di CloudWatch konsol, halaman Grup log menunjukkan hasil fungsi Lambda.	DevOps insinyur

Membersihkan infrastruktur setelah digunakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan infrastruktur.	Untuk membersihkan infrastruktur yang Anda buat, gunakan perintah berikut: `terraform destroy` Untuk mengkonfirmasi `destroy` perintah, ketik`yes`.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi setelah pembersih an.	Verifikasi bahwa sumber daya berhasil dihapus.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Masalah koneksi antara AWS Directory Service (akun induk) dan EC2 instans Amazon (akun anak) - Anda tidak dapat bergabung dengan komputer akun anak ke AD meskipun peering VPC tersedia.	Tambahkan perutean di. VPCs Untuk petunjuk, lihat Mengkonfigurasi koneksi peering VPC antara pemilik direktori dan akun konsumen direktori dalam dokumentasi. AWS Directory Service

Sumber daya terkait

AWS dokumentasi

- Amazon EventBridge dan AWS Identity and Access Management
- Konfigurasikan izin instans yang diperlukan untuk Systems Manager
- Identitas dan manajemen akses untuk Directory Service
- Kebijakan IAM berbasis identitas untuk Lambda
- Menggabungkan instans Amazon EC2 Windows secara manual ke AWS Managed Microsoft AD Active Directory
- Hapus EC2 entri Amazon dalam hal yang sama Akun AWS dari AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi

Sumber daya lainnya

- AWS Penyedia (dokumentasi Terraform)
- Konfigurasi Backend (dokumentasi Terraform)
- Instal Terraform (dokumentasi Terraform)
- Modul boto Python (repositori Indeks Paket Python)

- Unduhan biner Terraform (dokumentasi Terraform)

Hapus EC2 entri Amazon dalam hal yang sama Akun AWS dari AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi

Dr. Rahul Sharad Gaikwad dan Tamilselvan P, Amazon Web Services

Ringkasan

Active Directory (AD) adalah alat skrip Microsoft yang mengelola informasi domain dan interaksi pengguna dengan layanan jaringan. Ini banyak digunakan di antara penyedia layanan terkelola (MSPs) untuk mengelola kredensi karyawan dan izin akses. Karena penyerang AD dapat menggunakan akun tidak aktif untuk mencoba dan meretas organisasi, penting untuk menemukan akun yang tidak aktif dan menonaktifkannya pada jadwal pemeliharaan rutin. Dengan AWS Directory Service for Microsoft Active Directory, Anda dapat menjalankan Microsoft Active Directory sebagai layanan terkelola.

Pola ini dapat membantu Anda mengonfigurasi AWS Lambda otomatisasi untuk menemukan dan menghapus akun yang tidak aktif dengan cepat. Bila Anda menggunakan pola ini, Anda bisa mendapatkan manfaat berikut:

- Tingkatkan kinerja database dan server, dan perbaiki kerentanan keamanan Anda dari akun yang tidak aktif.
- Jika server iklan Anda di-host di cloud, menghapus akun yang tidak aktif juga dapat mengurangi biaya penyimpanan sekaligus meningkatkan kinerja. Tagihan bulanan Anda mungkin berkurang karena biaya bandwidth dan sumber daya komputasi keduanya dapat turun.
- Jauhkan penyerang potensial dengan Active Directory yang bersih.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Git diinstal dan dikonfigurasi pada workstation lokal.
- Terraform diinstal dan dikonfigurasi pada workstation lokal.
- Komputer Windows dengan modul Active Directory (`ActiveDirectory`).
- Sebuah direktori di AWS Managed Microsoft AD dan kredensional disimpan dalam parameter di AWS Systems Manager Parameter Store.

- AWS Identity and Access Management (IAM) peran dengan izin untuk yang Layanan AWS tercantum di Alat. Untuk informasi selengkapnya tentang IAM, lihat Sumber daya terkait.

Batasan

- Pola ini tidak mendukung penyiapan lintas akun.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Terraform versi 1.1.9 atau yang lebih baru
- Terraform AWS Provider versi 3.0 atau lebih tinggi

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Diagram menunjukkan alur kerja berikut:

1. Amazon EventBridge memicu AWS Lambda fungsi berdasarkan ekspresi cron. (Untuk pola ini, jadwal ekspresi cron adalah sekali per hari.)
2. Peran dan kebijakan IAM yang diperlukan dibuat dan dilampirkan AWS Lambda melalui Terraform.
3. AWS Lambda Fungsi ini dijalankan dan panggilan ke Grup Auto Scaling Amazon Elastic Compute Cloud (Amazon EC2) dengan menggunakan modul boto Python. Fungsi Lambda mendapatkan id instance acak. Id instance digunakan untuk menjalankan AWS Systems Manager perintah.
4. AWS Lambda membuat panggilan lain ke Amazon EC2 menggunakan modul boto dan mendapatkan alamat IP pribadi dari server Windows yang sedang berjalan dan menyimpan alamat dalam variabel sementara.
5. AWS Lambda membuat panggilan lain ke Systems Manager untuk mendapatkan informasi komputer yang terhubung ke Directory Service.
6. AWS Systems Manager Dokumen membantu mengeksekusi PowerShell skrip di server Amazon EC2 Windows untuk mendapatkan alamat IP pribadi komputer yang terhubung dengan AD.

7. Nama pengguna dan kata sandi domain AD disimpan di AWS Systems Manager Parameter Store. AWS Lambda dan Systems Manager melakukan panggilan ke Parameter Store dan mendapatkan nilai nama pengguna dan kata sandi yang akan digunakan untuk menghubungkan AD.
8. Menggunakan dokumen Systems Manager, PowerShell skrip dijalankan di server Amazon EC2 Windows menggunakan id instance yang diperoleh sebelumnya di langkah 3.
9. Amazon EC2 terhubung Directory Service dengan menggunakan PowerShell perintah dan menghapus komputer yang tidak digunakan atau tidak aktif.

Alat

Layanan AWS

- AWS Directory Servicemenyediakan beberapa cara untuk menggunakan Microsoft Active Directory (AD) dengan yang lain Layanan AWS seperti Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS) untuk SQL Server, dan FSx Amazon untuk Windows File Server.
- AWS Directory Service for Microsoft Active Directorymemungkinkan beban kerja dan sumber daya yang sadar direktori Anda AWS untuk menggunakan Microsoft Active Directory di file. AWS Cloud
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya. Dengan IAM, Anda dapat menentukan siapa atau apa yang dapat mengakses layanan dan sumber daya AWS, mengelola izin berbutir halus secara terpusat, dan menganalisis akses untuk menyempurnakan izin. AWS
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu

untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar.

- AWS Systems Manager dokumen menentukan tindakan yang dilakukan Systems Manager pada instans terkelola Anda. Systems Manager mencakup lebih dari 100 dokumen pra-konfigurasi yang dapat Anda gunakan dengan menentukan parameter di runtime.
- AWS Systems Manager Parameter Store adalah kemampuan AWS Systems Manager dan menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.

Alat-alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sumber terbuka sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- PowerShelladalah program manajemen otomatisasi dan konfigurasi Microsoft yang berjalan di Windows, Linux, dan macOS.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori C ode

Kode untuk pola ini tersedia di repositori solusi GitHub Custom AD Cleanup Automation.

Praktik terbaik

- Bergabung dengan domain secara otomatis.Saat Anda meluncurkan instance Windows yang akan menjadi bagian dari Directory Service domain, bergabunglah dengan domain selama proses pembuatan instance alih-alih menambahkan instance secara manual nanti. Untuk bergabung dengan domain secara otomatis, pilih direktori yang benar dari daftar dropdown direktori Gabung Domain saat meluncurkan instance baru. Untuk detail selengkapnya, lihat Menggabungkan instans Amazon EC2 Windows dengan mulus ke Direktori AWS Managed Microsoft AD Aktif Anda di Panduan Directory Service Administrasi.
- Hapus akun yang tidak digunakan.Adalah umum untuk menemukan akun di AD yang belum pernah digunakan. Seperti akun yang dinonaktifkan atau tidak aktif yang tetap berada di sistem, akun yang tidak digunakan yang diabaikan dapat memperlambat sistem iklan Anda atau membuat organisasi Anda rentan terhadap pelanggaran data.
- Otomatiskan pembersihan Direktori Aktif.Untuk membantu mengurangi risiko keamanan dan mencegah akun usang memengaruhi kinerja AD, lakukan pembersihan AD harus dilakukan secara

berkala. Anda dapat menyelesaikan sebagian besar tugas manajemen dan pembersihan AD dengan menulis skrip. Contoh tugas termasuk menghapus akun yang dinonaktifkan dan tidak aktif, menghapus grup kosong dan tidak aktif, dan menemukan akun pengguna dan kata sandi yang kedaluwarsa.

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat folder proyek, dan tambahkan file.	Untuk mengkloning repositori dan membuat folder proyek, lakukan hal berikut: 1. Buka GitHub repositori pola ini. 2. Pilih tombol Kode untuk melihat opsi untuk mengkloning di dropdown Clone. 3. Pada tab HTTPS, salin URL yang disediakan di Clone menggunakan URL web. 4. Buat folder di mesin Anda, dan beri nama dengan nama proyek Anda. 5. Buka terminal di mesin lokal Anda, dan arahkan ke folder ini. 6. Untuk mengkloning repositori git, gunakan perintah berikut. `git clone <repository-URL>.git`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Setelah repositori telah dikloning, gunakan perintah berikut untuk pergi ke direktori kloning. `cd <directory name>` 8. Di repositori kloning, buka proyek ini di lingkungan pengembangan terintegrasi (IDE) pilihan Anda.	

Menyediakan arsitektur target dengan menggunakan konfigurasi Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi konfigurasi Terraform.	Untuk menginisialisasi direktori kerja Anda yang berisi file Terraform, jalankan perintah berikut. `terraform init`	DevOps insinyur
Pratinjau perubahan.	Anda dapat melihat pratinjau perubahan yang akan dilakukan Terraform pada infrastruktur sebelum infrastruktur Anda diterapkan. Untuk memvalidasi bahwa Terraform akan membuat perubahan sesuai kebutuhan, jalankan perintah berikut. `terraform plan`	DevOps insinyur
Jalankan tindakan yang diusulkan.	Untuk memverifikasi bahwa hasil dari `terraform`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`plan` perintah seperti yang diharapkan, lakukan hal berikut: 1. Jalankan perintah berikut. `terraform apply` 2. Masuk ke Konsol Manajemen AWS, dan verifikasi bahwa sumber daya ada.	
Bersihkan infrastruktur.	Untuk membersihkan infrastruktur yang Anda buat, gunakan perintah berikut. `terraform destroy` Untuk mengonfirmasi perintah hancurkan, ketik`yes`.	DevOps insinyur

Verifikasi penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan dan uji fungsi Lambda.	Untuk memverifikasi bahwa penerapan berhasil terjadi, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS dan buka konsol. Buka halaman Fungsi, dan pilih nama fungsi yang dimulai dengan ADcleanup-Lambda-*.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pada halaman ikhtisar fungsi, pilih Uji pada tab Kode di bagian Sumber kode. 3. Untuk menyimpan acara pengujian, berikan nama untuk acara tersebut dan pilih Simpan. Kemudian untuk menguji acara, pilih Uji lagi. Hasil eksekusi menunjukkan output dari fungsi.	
Lihat hasil fungsi Lambda.	Dalam pola ini, EventBridge aturan mengeksekusi fungsi Lambda sekali per hari. Untuk melihat hasil fungsi Lambda, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol. Buka halaman Fungsi dan pilih nama fungsi yang dimulai dengan ADcleanup-Lambda-*. 2. Pilih tab Monitor dan pilih Lihat CloudWatch log. Di CloudWatch konsol, halaman Grup log menunjukkan hasil fungsi Lambda.	DevOps insinyur

Membersihkan infrastruktur setelah digunakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan infrastruktur.	Untuk membersihkan infrastruktur yang Anda buat, gunakan perintah berikut. `terraform destroy` Untuk mengonfirmasi perintah hancurkan, ketik`yes`.	DevOps insinyur
Verifikasi setelah pembersihan.	Verifikasi bahwa sumber daya berhasil dihapus.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Jika Anda mencoba menghapus komputer AD, Anda mendapatkan pesan “Akses Ditolak”. Komputer AD tidak dapat dihapus karena, secara default, tindakan mencoba menghapus dua alamat IP pribadi yang terhubung sebagai bagian dari layanan AD.	Untuk menghindari kesalahan ini, gunakan operasi Python berikut untuk mengabaikan dua komputer pertama ketika Anda mencantumkan perbedaan antara output komputer AD dan output dari mesin Anda yang menjalankan Windows. `Difference = Difference[2:]`
Ketika Lambda mengeksekusi PowerShell skrip pada server Windows, ia mengharapkan modul Active Directory tersedia secara default. Jika modul tidak tersedia, fungsi Lambda membuat kesalahan yang menyatakan “Get- AdComputer is not installed on instance”.	Untuk menghindari kesalahan ini, instal modul yang diperlukan dengan menggunakan data pengguna EC2 instance. Gunakan EC2WindowsUserdataskrip yang ada di GitHub repositori pola ini.

Sumber daya terkait

AWS dokumentasi

- Amazon EventBridge dan AWS Identity and Access Management
- Konfigurasikan izin instans yang diperlukan untuk Systems Manager
- Identitas dan manajemen akses untuk Directory Service
- Menggabungkan instans Amazon EC2 Windows secara manual ke Direktori AWS Managed Microsoft AD Aktif Anda
- Bekerja dengan kebijakan IAM berbasis identitas di AWS Lambda

Sumber daya lainnya

- AWS Penyedia (dokumentasi Terraform)
- Konfigurasi Backend (dokumentasi Terraform)
- Instal Terraform (dokumentasi Terraform)
- Modul boto Python (repositori Indeks Paket Python)
- Unduhan biner Terraform (dokumentasi Terraform)

Jalankan pengujian unit untuk pekerjaan ETL Python dalam AWS Glue menggunakan kerangka pytest

Praveen Kumar Jeyarajan dan Vaidy Sankaran, Amazon Web Services

Ringkasan

Anda dapat menjalankan pengujian unit untuk pekerjaan ekstrak, transformasi, dan muat Python (ETL) AWS Glue di lingkungan pengembangan lokal, tetapi mereplikasi pengujian tersebut dalam DevOps pipeline bisa jadi sulit dan memakan waktu. Pengujian unit bisa sangat menantang ketika Anda memodernisasi proses ETL mainframe pada tumpukan teknologi. AWS Pola ini menunjukkan kepada Anda cara menyederhanakan pengujian unit, sambil menjaga fungsionalitas yang ada tetap utuh, menghindari gangguan pada fungsionalitas aplikasi utama saat Anda merilis fitur baru, dan memelihara perangkat lunak berkualitas tinggi. Anda dapat menggunakan langkah-langkah dan contoh kode dalam pola ini untuk menjalankan pengujian unit untuk pekerjaan ETL Python AWS Glue dengan menggunakan kerangka pytest di. AWS CodePipeline Anda juga dapat menggunakan pola ini untuk menguji dan menyebarkan beberapa AWS Glue pekerjaan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- URI gambar Amazon Elastic Container Registry (Amazon ECR) Registry ECR) untuk perpustakaan AWS Glue Anda, diunduh dari Galeri Publik Amazon ECR
- Terminal Bash (pada sistem operasi apa pun) dengan profil untuk target Akun AWS dan Wilayah AWS
- Python 3.10 atau yang lebih baru
- Pytest
- Pustaka Moto Python untuk pengujian Layanan AWS

Arsitektur

Diagram berikut menjelaskan cara menggabungkan pengujian unit untuk proses AWS Glue ETL yang didasarkan pada Python ke dalam pipa skala perusahaan yang khas AWS DevOps .

Diagram menunjukkan alur kerja berikut:

1. Pada tahap sumber, AWS CodePipeline gunakan bucket Amazon Simple Storage Service (Amazon S3) berversi versi untuk menyimpan dan mengelola aset kode sumber. Aset ini termasuk contoh Python ETL job (`sample.py`), unit test file (`test_sample.py`), dan template. AWS CloudFormation Kemudian, CodePipeline transfer kode terbaru dari cabang utama ke AWS CodeBuild proyek untuk diproses lebih lanjut.
2. Pada tahap build dan publish, kode terbaru dari tahap sumber sebelumnya diuji unit dengan bantuan gambar ECR Amazon AWS Glue publik. Kemudian, laporan pengujian dipublikasikan untuk CodeBuild melaporkan kelompok. Gambar kontainer di repositori ECR Amazon publik untuk AWS Glue pustaka mencakup semua binari yang diperlukan untuk menjalankan dan tugas ETL PySparkberbasis pengujian unit secara lokal. AWS Glue Repositori kontainer publik memiliki tiga tag gambar, satu untuk setiap versi yang didukung oleh. AWS Glue Untuk tujuan demonstrasi, pola ini menggunakan tag `glue_libs_4.0.0_image_01` gambar. Untuk menggunakan gambar kontainer ini sebagai gambar runtime CodeBuild, salin URI gambar yang sesuai dengan tag gambar yang ingin Anda gunakan, lalu perbarui `pipeline.yml` file di GitHub repositori untuk sumber daya. `TestBuild`
3. Pada tahap penerapan, CodeBuild proyek diluncurkan dan menerbitkan kode ke bucket Amazon S3 jika semua tes lulus.
4. Pengguna menyebarkan AWS Glue tugas dengan menggunakan CloudFormation template di `deploy` folder.

Alat

Layanan AWS

- AWS CodeBuildadalah layanan build yang dikelola sepenuhnya yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- AWS Glueadalah layanan ETL yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri.

Alat lainnya

- Python adalah bahasa pemrograman tujuan umum tingkat tinggi yang ditafsirkan.
- Moto adalah perpustakaan Python untuk pengujian. Layanan AWS
- Pytest adalah kerangka kerja untuk menulis tes unit kecil yang skala untuk mendukung pengujian fungsional yang kompleks untuk aplikasi dan pustaka.
- Python ETL library untuk AWS Glue adalah repositori untuk pustaka Python yang digunakan dalam pengembangan lokal pekerjaan batch untuk. PySpark AWS Glue

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub aws-glue-jobs-unit-testing. Repositori mencakup sumber daya berikut:

- Contoh AWS Glue pekerjaan berbasis Python di folder `src`
- Kasus uji unit terkait (dibuat menggunakan kerangka pytest) di folder `tests`
- CloudFormation Template (ditulis dalam YAMAL) di folder `deploy`

Praktik terbaik

Keamanan untuk CodePipeline sumber daya

Ini adalah praktik terbaik untuk menggunakan enkripsi dan otentikasi untuk repositori sumber yang terhubung ke saluran pipa Anda. CodePipeline Untuk informasi selengkapnya, lihat Praktik terbaik keamanan dalam CodePipeline dokumentasi.

Pemantauan dan pencatatan CodePipeline sumber daya

Ini adalah praktik terbaik untuk menggunakan fitur AWS logging untuk menentukan tindakan apa yang diambil pengguna di akun Anda dan sumber daya apa yang mereka gunakan. File log menunjukkan hal berikut:

- Waktu dan tanggal tindakan
- Alamat IP sumber tindakan

- Tindakan mana yang gagal karena izin yang tidak memadai

Fitur logging tersedia di AWS CloudTrail dan Amazon CloudWatch Events. Anda dapat menggunakan CloudTrail untuk mencatat panggilan AWS API dan peristiwa terkait yang dibuat oleh atau atas nama Anda Akun AWS. Untuk informasi selengkapnya, lihat Logging panggilan CodePipeline API dengan AWS CloudTrail CodePipeline dokumentasi.

Anda dapat menggunakan CloudWatch Acara untuk memantau AWS Cloud sumber daya dan aplikasi yang sedang berjalan AWS. Anda juga dapat membuat peringatan di CloudWatch Acara. Untuk informasi selengkapnya, lihat Memantau CodePipeline peristiwa dalam CodePipeline dokumentasi.

Epik

Menyebarkan kode sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan arsip kode untuk penyebaran.	1. Unduh `code.zip` dari repositori GitHub aws-glue-jobs-unit-testing, atau buat file.zip sendiri dengan menggunakan alat baris perintah. Misalnya, Anda dapat membuat file.zip di Linux atau Mac dengan menjalankan perintah berikut di terminal:
	`git clone https://github.com/aws-samples/aws-glue-jobs-unit-testing.git`
	`cd aws-glue-jobs-unit-testing`
	`git checkout master`
	`zip -r code.zip src/ tests/ deploy/`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Masuk ke Konsol Manajemen AWSdan pilih Wilayah AWS pilihan Anda. 3. Buat bucket Amazon S3, lalu unggah paket dan `code.zip` file .zip (diunduh sebelumnya) ke bucket Amazon S3 yang Anda buat.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan.	1. Masuk ke Konsol Manajemen AWS dan kemudian buka CloudFormation konsol. 2. Pilih Buat tumpukan, lalu pilih Dengan sumber daya yang ada (sumber daya impor). 3. Di bagian Tentukan templat dari halaman Buat tumpukan, pilih Unggah file templat, lalu pilih template pipeline.yml. (diunduh dari repositori). GitHub Lalu, pilih Selanjutnya. 4. Untuk nama Stack, masukkan glue-unit-testing-pipeline, atau pilih nama tumpukan pilihan Anda. 5. Untuk ApplicationStackName, gunakan nama yang sudah terisi sebelumnya glue-codepipeline-app. Ini adalah nama CloudFormation tumpukan yang dibuat oleh pipeline. 6. Untuk BucketName, gunakan nama bucket aws-glue-artifacts-us-east-1 yang telah diisi sebelumnya. Ini adalah nama bucket Amazon S3 yang berisi file.zip dan digunakan oleh	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pipeline untuk menyimpan artefak kode. 7. Untuk CodeZipFile, gunakan nilai code.zip yang telah diisi sebelumnya. Ini adalah nama kunci dari kode sampel objek Amazon S3. Objek harus berupa file.zip. 8. Untuk TestRepor tGroupName, gunakan nama yang sudah terisi sebelumnya glue-unittest-report. Ini adalah nama grup laporan CodeBuild pengujian yang dibuat untuk menyimpan laporan pengujian unit. 9. Pilih Berikutnya, lalu pilih Berikutnya lagi di halaman Configure stack options. 10. Pada halaman Tinjauan, di bawah Kemampuan, pilih opsi Saya mengakui yang CloudFormation mungkin membuat sumber daya IAM dengan nama kustom. 11. Pilih Kirim. Setelah pembuatan tumpukan selesai, Anda dapat melihat sumber daya yang dibuat di tab Sumber Daya. Pembuatan tumpukan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membutuhkan waktu sekitar 5-7 menit. Tumpukan membuat CodePipeline tampilan menggunakan Amazon S3 sebagai sumbernya. Pada langkah-langkah di atas, pipa adalah aws-glue-unit-test-pipeline.	

Jalankan pengujian unit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan pengujian unit di dalam pipa.	1. Untuk menguji pipeline yang digunakan, masuk ke Konsol Manajemen AWS, lalu buka CodePipeline konsol. 2. Pilih pipeline yang dibuat oleh CloudFormation tumpukan, lalu pilih Rilis perubahan. Pipeline mulai berjalan (menggunakan kode terbaru di bucket Amazon S3). 3. Setelah fase test_and_build selesai, pilih tab Details, dan kemudian periksa log. 4. Pilih tab Laporan, lalu pilih laporan pengujian dari	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Riwayat laporan untuk melihat hasil pengujian unit. 5. Setelah tahap penerapan selesai, jalankan dan pantau AWS Glue pekerjaan yang diterapkan di konsol. AWS Glue Untuk informasi selengkapnya, lihat Pemantauan AWS Glue dalam AWS Glue dokumentasi.	

Bersihkan semua AWS sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan sumber daya di lingkungan Anda.	Untuk menghindari biaya infrastruktur tambahan, pastikan Anda menghapus tumpukan setelah bereksper imen dengan contoh yang disediakan dalam pola ini. 1. Buka CloudFormation konsol, lalu pilih tumpukan yang Anda buat. 2. Pilih Hapus. Ini menghapus semua sumber daya yang dibuat stack Anda, termasuk peran AWS Identity and Access Management (IAM), kebijakan IAM, dan proyek. CodeBuild	AWS DevOps, DevOps insinyur

Pemecahan masalah

Isu	Solusi
Peran CodePipeline layanan tidak dapat mengakses bucket Amazon S3.	• Untuk kebijakan yang dilampirkan pada peran CodePipeline layanan Anda, tambahkan `s3:ListBucket` ke daftar tindakan dalam kebijakan Anda. Untuk petunjuk cara melihat kebijakan peran layanan, lihat Melihat ARN pipeline dan peran layanan ARN (konsol). Edit pernyataan kebijakan untuk peran layanan Anda seperti yang dijelaskan di Tambahkan izin ke peran CodePipeline layanan. • Untuk kebijakan berbasis sumber daya yang dilampirkan ke bucket artefak Amazon S3 untuk pipeline Anda, juga disebut kebijakan bucket artefak, tambahkan pernyataan yang memungkinkan peran layanan menggunakan izin. CodePipeline `s3:ListBucket`
CodePipeline mengembalikan kesalahan bahwa bucket Amazon S3 tidak berversi.	CodePipeline mengharuskan bucket Amazon S3 sumber diberi versi. Aktifkan pembuatan versi di bucket Amazon S3 Anda. Untuk petunjuk, lihat Mengaktifkan pembuatan versi pada bucket.

Sumber daya terkait

- AWS Glue
- Mengembangkan dan menguji AWS Glue pekerjaan secara lokal
- AWS CloudFormation untuk AWS Glue

Informasi tambahan

Selain itu, Anda dapat menerapkan AWS CloudFormation template dengan menggunakan AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat Menerapkan templat dengan cepat dengan transformasi dalam dokumentasi. CloudFormation

Siapkan CI/CD pipeline dengan menggunakan AWS CodePipeline dan AWS CDK

Konstantin Zarudaev, Yasha Dabas, Lars Kinder, dan Cizer Pereira, Amazon Web Services

Rumah

Mengotomatiskan proses pembuatan dan rilis perangkat lunak Anda dengan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) supports repeatable builds and rapid delivery of new features to your users. You can quickly and easily test each code change, and you can catch and fix bugs before releasing your software. By running each change through your staging and release process, you can verify the quality of your application or infrastructure code. CI/CDmewujudkan budaya, seperangkat prinsip operasi, dan kumpulan praktik yang membantu tim pengembangan aplikasi untuk memberikan perubahan kode lebih sering dan andal. Implementasinya juga dikenal sebagai pipa CI/CD.

Pola ini mendefinisikan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang dapat digunakan kembali di Amazon Web Services (AWS) dengan repositori AWS. CodeCommit CodePipeline Pipeline AWS ditulis menggunakan AWS Cloud Development Kit (AWS CDK) v2.

Dengan menggunakan CodePipeline, Anda dapat memodelkan berbagai tahapan proses rilis perangkat lunak Anda melalui antarmuka AWS Management Console, AWS Command Line Interface (AWS CLI), AWS, atau CloudFormation AWS. SDKs Pola ini menunjukkan implementasi CodePipeline dan komponennya menggunakan AWS CDK. Selain membangun pustaka, AWS CDK menyertakan toolkit (perintah CLI`cdk`), yang merupakan alat utama untuk berinteraksi dengan aplikasi AWS CDK Anda. Di antara fungsi lainnya, toolkit menyediakan kemampuan untuk mengonversi satu atau lebih tumpukan ke CloudFormation templat dan menerapkannya ke akun AWS.

Pipeline mencakup pengujian untuk memvalidasi keamanan pustaka pihak ketiga Anda, dan membantu memastikan rilis otomatis yang dipercepat di lingkungan yang ditentukan. Anda dapat meningkatkan keamanan keseluruhan aplikasi Anda dengan menempatkannya melalui proses validasi.

Maksud dari pola ini adalah untuk mempercepat penggunaan CI/CD pipeline Anda untuk menyebarkan kode Anda sambil memastikan sumber daya yang Anda terapkan mematuhi praktik terbaik. DevOps Setelah menerapkan kode contoh, Anda akan memiliki AWS CodePipeline dengan proses linting, pengujian, pemeriksaan keamanan, penerapan, dan pasca-penerapan.

Pola ini juga mencakup langkah-langkah untuk Makefile. Menggunakan Makefile, pengembang dapat mereproduksi CI/CD langkah-langkah secara lokal dan meningkatkan kecepatan proses pengembangan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Pemahaman dasar sebagai berikut:
  - AWS CDK
  - AWS CloudFormation
  - AWS CodePipeline
  - TypeScript

Batasan

Pola ini TypeScript hanya menggunakan AWS CDK. Ini tidak mencakup bahasa lain yang didukung oleh AWS CDK.

Versi produk

Gunakan versi terbaru dari alat-alat berikut:

- Antarmuka Baris Perintah AWS (AWS CLI)
- cfn_nag
- git-remote-codecommit
- Node.js

Arsitektur

Tumpukan teknologi target

- AWS CDK
- AWS CloudFormation
- AWS CodeCommit
- AWS CodePipeline

Arsitektur target

Pipeline dipicu oleh perubahan dalam AWS CodeCommit repository ()`SampleRepository`. Pada awalnya, CodePipeline membangun artefak, memperbarui dirinya sendiri, dan memulai proses penyebaran. Pipeline yang dihasilkan menyebarkan solusi ke tiga lingkungan independen:

- Dev - Pemeriksaan kode tiga langkah di lingkungan pengembangan aktif
- Uji - Integrasi dan lingkungan uji regresi
- Prod - Lingkungan produksi

Tiga langkah yang termasuk dalam tahap Dev adalah linting, keamanan, dan pengujian unit. Langkah-langkah ini berjalan secara paralel untuk mempercepat proses. Untuk memastikan bahwa pipa hanya menyediakan artefak yang berfungsi, itu akan berhenti berjalan setiap kali langkah dalam proses gagal. Setelah penerapan tahap Dev, pipeline menjalankan tes validasi untuk memverifikasi hasilnya. Jika berhasil, pipeline kemudian akan menyebarkan artefak ke lingkungan Pengujian, yang berisi validasi pasca-penerapan. Langkah terakhir adalah menyebarkan artefak ke lingkungan Prod.

Diagram berikut menunjukkan alur kerja dari CodeCommit repositori ke proses build dan update yang dilakukan oleh CodePipeline, tiga langkah lingkungan Dev, dan penerapan dan validasi berikutnya di masing-masing dari tiga lingkungan.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS. Dalam pola ini CloudFormation template dapat digunakan untuk membuat CodeCommit repositori dan pipa CodePipeline CI/CD.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodePipeline adalah CI/CD layanan yang membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.

Alat lainnya

- cfn_nag adalah alat sumber terbuka yang mencari pola dalam CloudFormation templat untuk mengidentifikasi potensi masalah keamanan.
- git-remote-codecommitadalah utilitas untuk mendorong dan menarik kode dari CodeCommit repositori dengan memperluas Git.
- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.

Kode

Kode untuk pola ini tersedia di GitHub AWS CodePipeline dengan repositori CI/CD praktik.

Praktik terbaik

Tinjau sumber daya, seperti kebijakan AWS Identity and Access Management (IAM), untuk mengonfirmasi bahwa mereka selaras dengan praktik terbaik organisasi Anda.

Epik

Instal alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal alat di macOS atau Linux.	Jika Anda menggunakan macOS atau Linux, Anda dapat menginstal alat dengan menjalankan perintah berikut di terminal pilihan Anda atau menggunakan Homebrew untuk Linux.
	`brew install`
	`brew install git-remot`
	`e-codecommit`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`brew install ruby brew-gem` `brew-gem install cfn-nag`	
Siapkan AWS CLI.	Untuk menyiapkan AWS CLI, gunakan petunjuk untuk sistem operasi Anda: • Windows: Langkah-langkah penyiapan untuk koneksi HTTPS ke CodeCommit repositori AWS di Windows dengan pembantu kredensi AWS CLI • Linux, macOS, Unix: Langkah-langkah penyiapan untuk koneksi HTTPS ke CodeCommit repositori AWS di Linux, macOS, atau Unix dengan pembantu kredensi AWS CLI	DevOps insinyur

Siapkan penerapan awal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh atau kloning kode.	Untuk mendapatkan kode yang digunakan oleh pola ini, lakukan salah satu hal berikut: • Unduh kode sumber terbaru dari rilis di GitHub repo, dan unzip file yang diunduh ke dalam folder.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Kloning proyek dengan menjalankan perintah berikut.
	
	`git clone --depth 1 https://github.com/aws-samples/aws-codepipeline-cicd.git`
	
	Hapus `.git` direktori dari repositori kloning.
	
	`cd ./aws-codepipeline-cicd`
	`rm -rf ./.git`
	
	Nantinya, Anda akan menggunakan CodeCommit repositori AWS yang baru dibuat sebagai remote origin.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Connect ke akun AWS.	Anda dapat terhubung dengan menggunakan token keamanan sementara atau otentikasi landing zone. Untuk mengonfirmasi bahwa Anda menggunakan akun dan Wilayah AWS yang benar, jalankan perintah berikut.
	`AWS_REGION="eu-west-1"`
	`ACCOUNT_NUMBER=$(aws sts get-caller-identity --query Account --output text)`
	`echo "${ACCOUNT_NUMBER}"`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bootstrap lingkungan.	Untuk mem-bootstrap lingkungan AWS CDK, jalankan perintah berikut.
	`npm install`
	`npm run cdk bootstrap "aws://${ACCOUNT_NUMBER}/${AWS_REGION}"`
	Setelah Anda berhasil mem-bootstrap lingkungan, output berikut akan ditampilkan.
	`# Bootstrapping environment aws://{account}/{region}...`
	`# Environment aws://{account}/{region} bootstrapped`
	Untuk informasi selengkapnya tentang bootstrap AWS CDK, lihat dokumentasi AWS CDK.
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Sintesis template.	Untuk mensintesis aplikasi AWS CDK, gunakan perintah. `cdk synth`
	`npm run cdk synth`
	Anda akan melihat output berikut.
	`Successfully synthesized to <path-to-directory>/aws-codepipeline-cicd/cdk.out`
	`Supply a stack id (CodePipeline, Dev-MainStack) to display its template.`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan CodePipeline tumpukan.	Sekarang setelah Anda melakukan bootstrap dan mensintesis CloudFormation template, Anda dapat menerapkannya. Penyebaran akan membuat CodePipeline pipa dan CodeCommit repositori, yang akan menjadi sumber dan pemicu pipa.
	`npm run cdk -- deploy CodePipeline --require-approval never`
	Setelah Anda menjalankan perintah, Anda akan melihat penyebaran informasi CodePipeline tumpukan dan output yang berhasil. `CodePipeline.RepositoryName` Ini memberi Anda nama CodeCommit repositori di akun AWS.
	`CodePipeline: deploying...`
	`CodePipeline: creating CloudFormation changeset...`
	`# CodePipeline Outputs:`
	`CodePipeline.RepositoryName = SampleRepository`
	`Stack ARN: arn:aws:cloudformation:REGION:ACCOUNT-ID`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`:stack/CodePipeline/STACK-ID`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan CodeCommit repositori dan cabang jarak jauh.	Setelah penerapan berhasil, CodePipeline akan memulai proses pertama pipeline, yang dapat Anda temukan di konsol CodePipeline AWS. Karena AWS CDK dan CodeCommit tidak memulai cabang default, proses pipeline awal ini akan gagal dan mengembalikan pesan kesalahan berikut.
	`The action failed because no branch named main was found in the selected AWS CodeCommit repository SampleRepository. Make sure you are using the correct branch name, and then try again. Error: null`
	Untuk memperbaiki kesalahan ini, siapkan remote origin sebagai`SampleRepository`, dan buat `main` cabang yang diperlukan.
	`RepoName=$(aws cloudformation describe-stacks --stack-name CodePipeline --query "Stacks[0].Outputs[?OutputKey=='RepositoryName'].OutputValue" --output text)`
	`echo "${RepoName}"`
	`#`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`git init`
	`git branch -m master main`
	`git remote add origin codecommit://${RepoName}`
	`git add .`
	`git commit -m "Initial commit"`
	`git push -u origin main`	

Uji pipa yang digunakan CodePipeline

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan perubahan untuk mengaktifkan pipeline.	Setelah penerapan awal berhasil, Anda harus memiliki CI/CD pipeline lengkap dengan `main` cabang untuk `SampleRepository` sebagai cabang sumber. Segera setelah Anda melakukan perubahan ke `main` cabang, pipeline akan memulai dan menjalankan urutan tindakan berikut: 1. Dapatkan kode Anda dari CodeCommit repositori. 2. Bangun kode Anda. 3. Perbarui pipeline itu sendiri (`UpdatePipeline`). 4. Jalankan tiga pekerjaan paralel untuk pemeriksa	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an linting, keamanan, dan pengujian unit. 5. Jika berhasil, pipeline akan menyebarkan `Main` tumpukan dari `./lib/main-stack.ts` ke lingkungan Dev. 6. Jalankan pemeriksaan pasca-penerapan untuk sumber daya yang diterapkan. Anda dapat mengikuti semua CodePipeline langkah dan hasil di CodePipeline konsol. 7. Jika berhasil, pipeline akan mengulangi penerapan dan validasi untuk lingkungan Test dan Prod.	

Uji secara lokal dengan menggunakan Makefile

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan proses pengembangan dengan menggunakan Makefile.	Anda dapat menjalankan seluruh pipeline secara lokal dengan menggunakan `make` perintah, atau Anda dapat menjalankan langkah individual (misalnya,`make linting`). Untuk menguji penggunaan `make`, lakukan tindakan berikut:	Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Menerapkan pipa lokal: `make` • Jalankan hanya pengujian unit: `make unittest` • Terapkan ke akun saat ini: `make deploy` • Membersihkan lingkungan: `make clean`	

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya aplikasi AWS CDK.	Untuk membersihkan aplikasi AWS CDK Anda, jalankan perintah berikut. `cdk destroy --all` Ketahuilah bahwa bucket Amazon Simple Storage Service (Amazon S3) yang dibuat selama bootstrap tidak dihapus secara otomatis. Mereka memerlukan kebijakan retensi yang memungkinkan penghapusan, atau Anda perlu menghapusnya secara manual di akun AWS Anda.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Template tidak berfungsi seperti yang diharapkan.	Jika ada yang tidak beres dan template tidak berfungsi, pastikan Anda memiliki yang berikut ini: • Versi alat yang tepat. • Akses ke akun AWS target (konektivitas jaringan). • Izin yang cukup untuk akun AWS target.

Sumber daya terkait

- Memulai tugas umum di IAM Identity Center
- CodePipeline Dokumentasi AWS
- AWS CDK

Siapkan logging terpusat pada skala perusahaan dengan menggunakan Terraform

Aarti Rajput, Yashwant Patel, dan Nishtha Yadav, Amazon Web Services

Ringkasan

Pencatatan terpusat sangat penting untuk infrastruktur cloud organisasi, karena memberikan visibilitas ke dalam operasi, keamanan, dan kepatuhannya. Saat organisasi Anda menskalakan AWS lingkungannya di beberapa akun, strategi manajemen log terstruktur menjadi dasar untuk menjalankan operasi keamanan, memenuhi persyaratan audit, dan mencapai keunggulan operasional.

Pola ini menyediakan kerangka kerja yang dapat diskalakan dan aman untuk memusatkan log dari beberapa Akun AWS dan layanan, untuk memungkinkan manajemen logging skala perusahaan di seluruh penerapan yang kompleks. AWS Solusinya otomatis dengan menggunakan Terraform, yang merupakan alat infrastruktur sebagai kode (IAc) HashiCorp yang memastikan penerapan yang konsisten dan berulang, serta meminimalkan konfigurasi manual. Dengan menggabungkan Amazon CloudWatch Logs, Amazon Data Firehose, dan Amazon Simple Storage Service (Amazon S3), Anda dapat menerapkan alur agregasi dan analisis log yang kuat yang memberikan:

- Manajemen log terpusat di seluruh organisasi Anda di AWS Organizations
- Pengumpulan log otomatis dengan kontrol keamanan bawaan
- Pemrosesan log yang dapat diskalakan dan penyimpanan yang tahan lama
- Pelaporan kepatuhan dan jejak audit yang disederhanakan
- Wawasan dan pemantauan operasional waktu nyata

Solusi ini mengumpulkan log dari wadah, fungsi, dan instans database Amazon Elastic Kubernetes Service (Amazon EKS AWS Lambda), dan instans database Amazon Relational Database Service (Amazon RDS) melalui Log. CloudWatch Secara otomatis meneruskan log ini ke akun logging khusus dengan menggunakan filter CloudWatch berlangganan. Firehose mengelola pipeline streaming log throughput tinggi ke Amazon S3 untuk penyimpanan jangka panjang. Amazon Simple Queue Service (Amazon Simple Service) dikonfigurasi untuk menerima pemberitahuan peristiwa Amazon S3 setelah pembuatan objek. Ini memungkinkan integrasi dengan layanan analitik, termasuk:

- OpenSearch Layanan Amazon untuk penelusuran log, visualisasi, dan analitik waktu nyata
- Amazon Athena untuk kueri berbasis SQL

- Amazon EMR untuk pemrosesan skala besar
- Lambda untuk transformasi kustom
- Amazon Quick Sight untuk dasbor

Semua data dienkripsi dengan menggunakan AWS Key Management Service (AWS KMS), dan seluruh infrastruktur diterapkan dengan menggunakan Terraform untuk konfigurasi yang konsisten di seluruh lingkungan.

Pendekatan logging terpusat ini memungkinkan organisasi untuk meningkatkan postur keamanan mereka, mempertahankan persyaratan kepatuhan, dan mengoptimalkan efisiensi operasional di seluruh AWS infrastruktur mereka.

Prasyarat dan batasan

Prasyarat

- Sebuah landing zone untuk organisasi Anda yang dibangun dengan menggunakan AWS Control Tower
- Account Factory for Terraform (AFT), digunakan dan dikonfigurasi dengan akun yang diperlukan
- Terraform untuk penyediaan infrastruktur
- AWS Identity and Access Management (IAM) peran dan kebijakan untuk akses lintas akun

Untuk petunjuk pengaturan AWS Control Tower, AFT, dan akun Aplikasi, lihat bagian Epik.

Akun yang diperlukan

Organisasi Anda AWS Organizations harus menyertakan akun-akun ini:

- Akun aplikasi - Satu atau beberapa akun sumber tempat Layanan AWS (Amazon EKS, Lambda, dan Amazon RDS) menjalankan dan menghasilkan log
- Akun Arsip Log - Akun khusus untuk penyimpanan dan manajemen log terpusat

Versi produk

- AWS Control Tower versi 3.1 atau yang lebih baru
- Terraform versi 0.15.0 atau yang lebih baru

Arsitektur

Diagram berikut menggambarkan arsitektur logging AWS terpusat yang menyediakan solusi terukur untuk mengumpulkan, memproses, dan menyimpan log dari beberapa akun Aplikasi ke akun Arsip Log khusus. Arsitektur ini secara efisien menangani log dari Layanan AWS, termasuk Amazon RDS, Amazon EKS, dan Lambda, dan merutekkannya melalui proses yang disederhanakan ke bucket Regional S3 di akun Arsip Log.

Alur kerja mencakup lima proses:

1. Proses aliran log
   - Proses alur log dimulai di akun Aplikasi, di mana Layanan AWS menghasilkan berbagai jenis log, seperti umum, kesalahan, audit, log kueri lambat dari Amazon RDS, log bidang kontrol dari Amazon EKS, dan eksekusi fungsi dan log kesalahan dari Lambda.
   - CloudWatch berfungsi sebagai titik pengumpulan awal. Ini mengumpulkan log ini di tingkat grup log dalam setiap akun aplikasi.
   - Di CloudWatch, filter langganan menentukan log mana yang harus diteruskan ke akun pusat. Filter ini memberi Anda kontrol granular atas penerusan log, sehingga Anda dapat menentukan pola log yang tepat atau aliran log lengkap untuk sentralisasi.
2. Transfer log lintas akun
   - Log pindah ke akun Arsip Log. CloudWatch filter berlangganan memfasilitasi transfer lintas akun dan mempertahankan konteks Regional.
   - Arsitektur menetapkan beberapa aliran paralel untuk menangani sumber log yang berbeda secara efisien, untuk memastikan kinerja dan skalabilitas yang optimal.
3. Pemrosesan log di akun Arsip Log
   - Di akun Arsip Log, Firehose memproses aliran log yang masuk.
   - Setiap Wilayah mengelola aliran pengiriman Firehose khusus yang dapat mengubah, mengonversi, atau memperkaya log sesuai kebutuhan.
   - Aliran Firehose ini mengirimkan log yang diproses ke bucket S3 di akun Arsip Log, yang terletak di Wilayah yang sama dengan akun Aplikasi sumber (Wilayah A dalam diagram) untuk mempertahankan persyaratan kedaulatan data.
4. Pemberitahuan dan alur kerja tambahan
   - Saat log mencapai bucket S3 tujuan, arsitektur mengimplementasikan sistem notifikasi dengan menggunakan Amazon SQS.

- Antrian SQS Regional memungkinkan pemrosesan asinkron dan dapat memicu alur kerja tambahan, analitik, atau sistem peringatan berdasarkan log yang disimpan.

5. AWS KMS untuk keamanan

   Arsitektur menggabungkan AWS KMS untuk keamanan. AWS KMS menyediakan kunci enkripsi untuk bucket S3. Ini memastikan bahwa semua log yang disimpan mempertahankan enkripsi saat istirahat sambil menjaga enkripsi Regional untuk memenuhi persyaratan residensi data.

Alat

Layanan AWS

- Amazon CloudWatch adalah layanan pemantauan dan observabilitas yang mengumpulkan data pemantauan dan operasional dalam bentuk log, metrik, dan peristiwa. Ini memberikan tampilan terpadu AWS sumber daya, aplikasi, dan layanan yang berjalan di AWS dan server lokal.
- CloudWatch Filter langganan log adalah ekspresi yang cocok dengan pola dalam peristiwa log masuk dan mengirimkan peristiwa log yang cocok ke AWS sumber daya yang ditentukan untuk diproses atau dianalisis lebih lanjut.
- AWS Control Tower Account Factory For Terraform (AFT) menyiapkan pipeline Terraform untuk membantu Anda menyediakan dan menyesuaikan akun. AWS Control Tower AFT menyediakan penyediaan akun berbasis Terraform sambil memungkinkan Anda untuk mengatur akun Anda. AWS Control Tower
- Amazon Data Firehose memberikan data streaming real-time ke tujuan seperti Amazon S3, Amazon Redshift, dan Amazon Service. OpenSearch Ini secara otomatis menskalakan agar sesuai dengan throughput data Anda dan tidak memerlukan administrasi yang berkelanjutan.
- Amazon Elastic Kubernetes Service (Amazon EKS) adalah layanan orkestrasi kontainer terkelola yang memudahkan penerapan, pengelolaan, dan skala aplikasi kontainer dengan menggunakan Kubernetes. Secara otomatis mengelola ketersediaan dan skalabilitas node bidang kontrol Kubernetes.
- AWS Key Management Service (AWS KMS) membuat dan mengontrol kunci enkripsi untuk mengenkripsi data Anda. AWS KMS terintegrasi dengan yang lain Layanan AWS untuk membantu Anda melindungi data yang Anda simpan dengan layanan ini.
- AWS Lambdaadalah layanan komputasi tanpa server yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. Ini secara otomatis menskalakan aplikasi Anda dengan menjalankan kode sebagai respons terhadap setiap pemicu, dan hanya mengenakan biaya untuk waktu komputasi yang Anda gunakan.

- Amazon Relational Database Service (Amazon RDS) adalah layanan database relasional terkelola yang memudahkan pengaturan, pengoperasian, dan skala database relasional di cloud. Ini memberikan kapasitas hemat biaya dan dapat diubah ukurannya sambil mengotomatiskan tugas administrasi yang memakan waktu.
- Amazon Simple Queue Service (Amazon SQS) adalah layanan antrian pesan yang memungkinkan Anda memisahkan dan menskalakan layanan mikro, sistem terdistribusi, dan aplikasi tanpa server. Ini menghilangkan kompleksitas mengelola dan mengoperasikan middleware berorientasi pesan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja. Ini dapat menyimpan dan mengambil sejumlah data dari mana saja di web.

Alat-alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Kode

Kode untuk pola ini tersedia di repositori logging GitHub terpusat.

Praktik terbaik

- Gunakan beberapa Akun AWS dalam satu organisasi di AWS Organizations. Praktik ini memungkinkan manajemen terpusat dan pencatatan standar di seluruh akun.
- Konfigurasikan bucket S3 dengan pembuatan versi, kebijakan siklus hidup, dan replikasi lintas wilayah. Menerapkan enkripsi dan log akses untuk keamanan dan kepatuhan.
- Menerapkan standar logging umum dengan menggunakan format JSON dengan stempel waktu dan bidang standar. Gunakan struktur awalan dan korelasi yang konsisten IDs untuk memudahkan pelacakan dan analisis.
- Aktifkan kontrol keamanan dengan AWS KMS enkripsi dan akses hak istimewa paling sedikit. Pertahankan AWS CloudTrail pemantauan dan rotasi kunci reguler untuk keamanan yang ditingkatkan.
- Siapkan CloudWatch metrik dan peringatan untuk pelacakan pengiriman. Pantau biaya dan kinerja dengan notifikasi otomatis.
- Konfigurasikan kebijakan retensi Amazon S3 untuk memenuhi persyaratan kepatuhan dan aktifkan pencatatan akses server Amazon S3 untuk melacak semua permintaan yang dibuat ke bucket

S3 Anda. Menjaga dokumentasi untuk kebijakan bucket S3 dan aturan siklus hidup. Melakukan tinjauan berkala terhadap log akses, izin bucket, dan konfigurasi penyimpanan untuk membantu memastikan kepatuhan dan praktik terbaik keamanan.

Epik

Mengatur AWS Control Tower, AFT, dan akun Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS Control Tower lingkungan dengan AFT.	1. Terapkan AWS Control Tower dengan mengikuti instruksi dalam AWS Control Tower dokumentasi. 2. Menyebarkan AFT dengan mengikuti instruksi dalam AWS Control Tower dokumentasi.	Administrator AWS
Aktifkan berbagi sumber daya untuk organisasi.	1. Configure AWS Command Line Interface (AWS CLI) dengan kredenal akun manajemen, yang memberikan izin administr atif untuk dikelola. AWS Control Tower 2. Jalankan AWS CLI perintah berikut di salah satu Wilayah AWS: `aws ram enable-sharing-with-aws-organization` Ini memungkinkan berbagi sumber daya dalam organisasi Anda di AWS	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Organizations seluruh Wilayah yang mendukung AWS Resource Access Manager (AWS RAM).	
Verifikasi atau berikan akun Aplikasi.	Untuk menyediakan akun Aplikasi baru untuk kasus penggunaan Anda, buat melalui AFT. Untuk informasi selengkapnya, lihat Menyediakan akun baru dengan AFT dalam AWS Control Tower dokumentasi.	Administrator AWS

Mengatur file konfigurasi untuk akun Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin isi `Application_account` folder ke dalam `aft-account-customizations` repositori.	1. Buat folder bernama `Application_account` di jalur root `aft-account-customizations` repositori. Repositori ini dibuat secara otomatis saat Anda mengatur AFT (lihat epik sebelumnya). 2. Arahkan ke direktori root scale-using-terraform repositori centralised-logging-at-enterprise-, salin isi `aft/account` direktori, lalu tempelkan ke `Application_account` direktori yang Anda buat	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pada langkah 1 di repositori. `aft-account-customizations` 3. Dari direktori root `centralised-logging-at-enterprise-scale-using-terraform` repositori, salin isi `Application_account` direktori ke `Application_account/terraform` direktori di repositori. `aft-account-customizations` 4. Dalam `aft-account-customizations/Application_account/terraform.tfvars` file, konfirmasikan bahwa semua parameter diteruskan sebagai argumen dalam file konfigurasi Terraform yang sesuai.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau dan edit parameter input untuk menyiapkan akun Aplikasi.	Pada langkah ini, Anda menyiapkan file konfigurasi untuk membuat sumber daya di akun Aplikasi, termasuk grup CloudWatch log, filter CloudWatch langganan, peran dan kebijakan IAM, serta detail konfigurasi untuk fungsi Amazon RDS, Amazon EKS, dan Lambda. Di `aft-account-customizations` repositori Anda, di `Application_account` folder, konfigurasikan parameter input dalam `terraform.tfvars` file berdasarkan persyaratan organisasi Anda: • `environment` : Nama lingkungan (misalnya ,, `prod` `dev` `staging`) tempat sumber daya akan digunakan. • `account_name` : Nama Akun AWS tempat sumber daya akan dibuat. • `log_archive_account_id` : Akun AWS ID tempat log akan diarsipkan. • `admin_role_name` : Nama peran administratif	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang akan digunakan untuk mengelola sumber daya. • `tags`: Peta pasangan kunci-nilai yang mewakili tag umum untuk diterapkan ke semua sumber daya. • `rds_config` : Objek yang berisi detail konfigurasi untuk instans Amazon RDS. • `allowed_cidr_blocks` : Daftar blok CIDR yang diizinkan untuk mengakses sumber daya. • `destination_name:` Variabel yang digunakan untuk membuat Nama Sumber Daya Amazon (ARN) CloudWatch tujuan tempat log akan dialirkan. • `rds_parameters:` Objek yang berisi pengaturan grup parameter Amazon RDS. • `vpc_config` : Objek yang berisi detail konfigurasi VPC. • `eks_config` : Objek yang berisi detail konfigurasi untuk kluster Amazon EKS. • `lambda_config` : Objek yang berisi detail konfigurasi untuk fungsi Lambda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `restrictive_cidr_range` : Daftar rentang CIDR yang membatasi untuk aturan grup keamanan. • `target_account_id` : Akun AWS ID akun Arsip Log target tempat sumber daya akan digunakan.	

Mengatur file konfigurasi untuk akun Arsip Log

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin isi `Log_archive_account` folder ke dalam `aft-account-customizations` repositori.	1. Buat folder bernama `Log_archive_account` di jalur root repositori, `aft-account-customizations` Repositori ini dibuat secara otomatis ketika Anda mengatur AFT. 2. Arahkan ke direktori root `centralised-logging-at-enterprise-scale-using-terraform` repositori, salin isi `aft/account` direktori, lalu tempelkan ke `Log_archive_account` direktori yang Anda buat pada langkah sebelumnya di repositori. `aft-account-customizations` 3. Dari direktori root `centralised-loggin`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`g-at-enterprise-scale-using-terraform` repositori, salin isi `Log_archive_account` direktori ke `Log_archive_account/terraform` direktori di repositori. `aft-account-customizations` 4. Dalam `aft-account-customizations/Log_archive_account/terraform.tfvars` file, konfirmasikan bahwa semua parameter diteruskan sebagai argumen dalam file konfigurasi Terraform yang sesuai.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau dan edit parameter input untuk menyiapkan akun Arsip Log.	Pada langkah ini, Anda menyiapkan file konfigura si untuk membuat resource di akun Arsip Log, termasuk aliran pengiriman Firehose, bucket S3, antrian SQS, serta peran dan kebijakan IAM. Di `Log_archive_accoun t` folder `aft-accou nt-customizations` repositori Anda, konfigura sikan parameter input dalam `terraform.tfvars` file berdasarkan persyaratan organisasi Anda: • `environment` : Nama lingkungan (misalnya ,, `proddev`,`staging`) tempat sumber daya akan digunakan. • `destination_name` : Variabel yang digunakan untuk membuat ARN CloudWatch tujuan tempat log akan dialirkan. • `source_account_ids` : Daftar Akun AWS IDs yang diizinkan untuk menempatk an filter berlangganan di tujuan log. Anda dapat memasukkan akun IDs sebanyak yang ingin Anda	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	aktifkan untuk pencatatan terpusat.	

Jalankan perintah Terraform untuk menyediakan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 1 - Menyebarkan file konfigurasi Terraform dari AFT.	Di AFT, pipeline AFT dipicu setelah Anda mendorong kode dengan perubahan konfigura si ke GitHub `aft-account-customizations` repositor i. AFT secara otomatis mendeteksi perubahan dan memulai proses kustomisasi akun.
	
	Setelah Anda membuat perubahan pada file Terraform (`terraform.tfvars`) Anda, komit dan dorong perubahan Anda ke repositor i Anda`aft-account-customizations` :
	
	`$ git add *`
	`$ git commit -m "update message"`
	`$ git push origin main`
	
	Note
	Jika Anda menggunak an cabang yang berbeda (sepertidev),
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ganti `main` dengan nama cabang Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 2 - Menyebarkan file konfigurasi Terraform secara manual.	Jika Anda tidak menggunakan AFT atau ingin menerapkan solusi secara manual, Anda dapat menggunakan perintah Terraform berikut dari folder dan`Application_account` : `Log_archive_account`
	1. Kloning GitHub repositori dan konfigurasikan parameter input dalam file. `terraform.tfvars`
	2. Jalankan perintah berikut:
	`$ terraform init`
	3. Perubahan pratinjau:
	`$ terraform plan`
	Perintah ini mengevaluasi konfigurasi Terraform untuk menentukan status sumber daya yang diinginkan dan membandingkannya dengan status infrastruktur Anda saat ini.
	4. Terapkan perubahan:
	`$ terraform apply`
	5. Tinjau perubahan yang direncanakan dan ketik
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ya pada prompt untuk melanjutkan aplikasi.	

Validasi sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi filter langganan.	Untuk memverifikasi bahwa langganan memfilter meneruskan log dengan benar dari grup log akun Aplikasi ke akun Arsip Log: 1. Di akun Aplikasi, buka CloudWatch konsol. 2. Pada panel navigasi kiri, pilih Grup log. 3. Pilih setiap grup log (`/aws/rds`,`/aws/eks`,`/aws/lambda`) dan pilih tab Filter langganan. Anda akan melihat filter langganan aktif yang mengarah ke ARN tujuan, berdasarkan nama yang Anda tentukan dalam file konfigurasi Terraform. 4. Pilih filter langganan apa pun untuk memverifikasi konfigurasi dan statusnya.	DevOps insinyur
Verifikasi aliran Firehose.	Untuk memverifikasi bahwa Firehose melakukan streaming	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di log aplikasi proses akun Arsip Log berhasil: 1. Di akun Arsip Log, buka konsol Firehose. 2. Di panel navigasi kiri, pilih Firehose stream. 3. Pilih aliran Firehose dan verifikasi hal berikut: • Tujuan menunjukkan bucket S3 yang benar. • Tab Monitoring menunjukkan metrik pengiriman yang berhasil. • Stempel waktu pengirima n baru-baru ini adalah saat ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi bucket S3 terpusat.	Untuk memverifikasi bahwa bucket S3 terpusat menerima dan mengatur log dengan benar: 1. Di akun Arsip Log, buka konsol Amazon S3. 2. Pilih setiap bucket logging pusat. 3. Arahkan melalui struktur folder: AWSLogs/AccountID/Region/Service. Anda akan melihat file log yang diatur oleh timestamp () YYYY/MM/DD/HH. 4. Pilih file log terbaru dan verifikasi format dan integritas datanya.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi antrian SQS.	Untuk memverifikasi bahwa antrian SQS menerima pemberitahuan untuk file log baru: 1. Di akun Arsip Log, buka konsol Amazon SQS. 2. Di panel navigasi kiri, pilih Antrian. 3. Pilih setiap antrian yang dikonfigurasi dan pilih Kirim dan terima pesan. Anda akan melihat pesan yang berisi pemberitahuan acara S3 untuk file log baru. 4. Pilih pesan apa pun untuk memverifikasi bahwa pesan tersebut berisi informasi objek S3 yang benar.	DevOps insinyur

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 1 - Nonaktifkan file konfigurasi Terraform dari AFT.	Saat Anda menghapus file konfigurasi Terraform dan mendorong perubahan, AFT secara otomatis memulai proses penghapusan sumber daya.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	1. Arahkan ke `aft-account-customizations` repositori.
	2. Pergi ke `terraform` direktori.
	3. Hapus file-file berikut:
	• `modules`direktori
	• `iam.tf`
	• `versions.tf`
	• `variables.tf`
	• `outputs.tf`
	• `terraform.tfvars`
	4. Hapus isi `main.tf` file.
	5. Dorong perubahan Anda ke repositori:
	`# Stage all changes`
	`$ git add *`
	`# Commit cleanup changes`
	`$ git commit -m "Remove AFT customizations"`
	`# Push to repository`
	`$ git push origin main`
	Note
	Jika Anda menggunakan cabang yang berbeda	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(sepertidev), ganti `main` dengan nama cabang Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 2 — Bersihkan sumber daya Terraform secara manual.	Jika Anda tidak menggunakan AFT atau Anda ingin membersihkan sumber daya secara manual, gunakan perintah Terraform berikut dari folder `Application_account` dan `Log_archive_account` :
	1. Inisialisasi konfigurasi Terraform:
	`$ terraform init`
	Perintah ini menginisialisasi Terraform dan memastikan akses ke status saat ini.
	2. Pratinjau perubahan pembersihan:
	`$ terraform destroy`
	Perintah ini mengevaluasi sumber daya mana yang akan dihancurkan dan membandingkan keadaan yang diinginkan dengan keadaan infrastruktur Anda saat ini.
	3. Jalankan pembersihan. Saat Anda diminta, ketik ya untuk mengonfirmasi
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan menjalankan rencana penghancuran.	

Pemecahan masalah

Isu	Solusi
Tujuan CloudWatch Log tidak dibuat atau tidak aktif.	Validasi berikut ini: 1. Di akun Arsip Log, verifikasi bahwa kebijakan tujuan mencakup: • Prinsipal akun sumber yang benar. • Tindakan yang benar (`logs:PutSubscriptionFilter`). • ARN tujuan yang valid. 2. Konfirmasikan bahwa aliran Firehose ada dan aktif. 3. Verifikasi bahwa peran IAM yang dilampirkan ke tujuan memiliki izin untuk Firehose.
Filter langganan gagal atau macet dalam status tertunda.	Periksa hal-hal berikut: 1. Di akun Aplikasi, verifikasi bahwa peran IAM memiliki: • Izin untuk meneleponPutSubscriptionFilter. • Hubungan kepercayaan dengan CloudWatch Log. 2. Konfirmasikan bahwa ARN tujuan sudah benar. 3. Periksa CloudWatch Log untuk pesan kesalahan tertentu.

Isu	Solusi
Aliran pengiriman Firehose tidak menunjukkan catatan yang masuk.	Verifikasi hal berikut: 1. Konfirmasikan bahwa peran Firehose IAM memiliki: • Izin untuk menulis ke Amazon S3. • Akses ke AWS KMS kunci jika enkripsi diaktifkan. 2. Tinjau CloudWatch metrik untuk: • `IncomingRecords` • `DeliveryToS3.Records` 3. Verifikasi pengaturan buffer dan konfigurasi pengiriman.

Sumber daya terkait

- Pengaturan infrastruktur Terraform (dokumentasi Terraform)
- Menyebarkan AWS Control Tower Account Factory untuk Terraform (AFT) (dokumentasi)AWS Control Tower
- Tutorial IAM: Delegasikan akses di seluruh Akun AWS menggunakan peran IAM () IAMdocumentation

Siapkan end-to-end enkripsi untuk aplikasi di Amazon EKS menggunakan cert-manager dan Let's Encrypt

Mahendra Revanasiddappa dan Vasanth Jeyaraj, Amazon Web Services

Ringkasan

Menerapkan end-to-end enkripsi dapat menjadi rumit dan Anda perlu mengelola sertifikat untuk setiap aset dalam arsitektur layanan mikro Anda. Meskipun Anda dapat menghentikan koneksi Transport Layer Security (TLS) di tepi jaringan Amazon Web Services (AWS) dengan Network Load Balancer atau Amazon API Gateway, beberapa organisasi memerlukan enkripsi. end-to-end

Pola ini menggunakan NGINX Ingress Controller untuk masuk. Ini karena ketika Anda membuat ingress Kubernetes, resource ingress menggunakan Network Load Balancer. Network Load Balancer tidak mengizinkan pengunggahan sertifikat klien. Oleh karena itu, Anda tidak dapat mencapai TLS timbal balik dengan masuknya Kubernetes.

Pola ini ditujukan untuk organisasi yang membutuhkan otentikasi timbal balik antara semua layanan mikro dalam aplikasi mereka. Mutual TLS mengurangi beban mempertahankan nama pengguna atau kata sandi dan juga dapat menggunakan kerangka keamanan turnkey. Pendekatan pola ini kompatibel jika organisasi Anda memiliki sejumlah besar perangkat yang terhubung atau harus mematuhi pedoman keamanan yang ketat.

Pola ini membantu meningkatkan postur keamanan organisasi Anda dengan menerapkan end-to-end enkripsi untuk aplikasi yang berjalan di Amazon Elastic Kubernetes Service (Amazon EKS). Pola ini menyediakan contoh aplikasi dan kode dalam GitHub End-to-end enkripsi di repositori Amazon EKS untuk menunjukkan bagaimana layanan mikro berjalan dengan enkripsi end-to-end di Amazon EKS. Pendekatan pola menggunakan cert-manager, sebuah add-on untuk Kubernetes, dengan Let's Encrypt sebagai certificate authority (CA). Let's Encrypt adalah solusi hemat biaya untuk mengelola sertifikat dan menyediakan sertifikat gratis yang berlaku selama 90 hari. Cert-manager mengotomatiskan penyediaan sesuai permintaan dan pemutaran sertifikat saat layanan mikro baru diterapkan di Amazon EKS.

Audiens yang dituju

Pola ini direkomendasikan untuk pengguna yang memiliki pengalaman dengan Kubernetes, TLS, Amazon Route 53, dan Domain Name System (DNS).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Sebuah klaster Amazon EKS yang sudah ada.
- AWS Command Line Interface (AWS CLI) versi 1.7 atau yang lebih baru, diinstal dan dikonfigurasi di macOS, Linux, atau Windows.
- Utilitas baris `kubectl` perintah, diinstal dan dikonfigurasi untuk mengakses cluster Amazon EKS. Untuk informasi selengkapnya tentang ini, lihat Menginstal kubectl di dokumentasi Amazon EKS.
- Nama DNS yang ada untuk menguji aplikasi. Untuk informasi selengkapnya tentang hal ini, lihat Mendaftarkan nama domain menggunakan Amazon Route 53 dalam dokumentasi Amazon Route 53.
- Versi Helm terbaru, diinstal pada mesin lokal Anda. Untuk informasi selengkapnya tentang ini, lihat Menggunakan Helm dengan Amazon EKS di dokumentasi Amazon EKS dan repositori GitHub Helm.
- GitHub End-to-end Enkripsi di repositori Amazon EKS, dikloning ke mesin lokal Anda.
- Ganti nilai berikut di `trustpolicy.json` file `policy.json` dan dari GitHub End-to-end enkripsi kloning di repositori Amazon EKS:
  - `<account number>`— Ganti dengan ID akun AWS untuk akun tempat Anda ingin menerapkan solusinya.
  - `<zone id>`— Ganti dengan ID zona Route 53 nama domain.
  - `<node_group_role>`— Ganti dengan nama peran AWS Identity and Access Management (IAM) yang terkait dengan node Amazon EKS.
  - `<namespace>`— Ganti dengan namespace Kubernetes di mana Anda menggunakan NGINX Ingress Controller dan aplikasi sampel.
  - `<application-domain-name>`— Ganti dengan nama domain DNS dari Route 53.

Batasan

- Pola ini tidak menjelaskan cara memutar sertifikat dan hanya menunjukkan cara menggunakan sertifikat dengan layanan mikro di Amazon EKS.

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Diagram menunjukkan alur kerja berikut:

1. Klien mengirimkan permintaan untuk mengakses aplikasi ke nama DNS.
2. Catatan Route 53 adalah CNAME ke Network Load Balancer.
3. Network Load Balancer meneruskan permintaan ke NGINX Ingress Controller yang dikonfigurasi dengan pendengar TLS. Komunikasi antara NGINX Ingress Controller dan Network Load Balancer mengikuti protokol HTTPS.
4. NGINX Ingress Controller melakukan routing berbasis jalur berdasarkan permintaan klien ke layanan aplikasi.
5. Layanan aplikasi meneruskan permintaan ke pod aplikasi. Aplikasi ini dirancang untuk menggunakan sertifikat yang sama dengan memanggil rahasia.
6. Pod menjalankan aplikasi sampel menggunakan sertifikat cert-manager. Komunikasi antara NGINX Ingress Controller dan pod menggunakan HTTPS.

Catatan: Cert-manager berjalan di namespace sendiri. Ini menggunakan peran klaster Kubernete s untuk menyediakan sertifikat sebagai rahasia di ruang nama tertentu. Anda dapat melampirkan namespace tersebut ke pod aplikasi dan NGINX Ingress Controller.

Alat

Layanan AWS

- Amazon Elastic Kubernetes Service (Amazon EKS) adalah layanan terkelola yang dapat Anda gunakan untuk menjalankan Kubernetes di AWS tanpa perlu menginstal, mengoperasikan, dan memelihara control plane atau node Kubernetes Anda sendiri.
- Elastic Load Balancing secara otomatis mendistribusikan lalu lintas masuk Anda ke beberapa target, kontainer, dan alamat IP.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.

Alat-alat lainnya

- cert-manager adalah add-on untuk Kubernetes yang meminta sertifikat, mendistribusikannya ke kontainer Kubernetes, dan mengotomatiskan perpanjangan sertifikat.
- NGINX Ingress Controller adalah solusi manajemen lalu lintas untuk aplikasi cloud-native di Kubernetes dan lingkungan kontainer.

Epik

Buat dan konfigurasikan zona yang dihosting publik dengan Route 53

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat zona yang dihosting publik di Route 53.	Masuk ke AWS Management Console, buka konsol Amazon Route 53, pilih Zona yang dihosting, lalu pilih Buat zona yang dihosting. Buat zona yang dihosting publik dan catat ID zona. Untuk informasi selengkapnya tentang hal ini, lihat Membuat zona yang dihosting publik di dokumentasi Amazon Route 53. **Note** ACME DNS01 menggunakan penyedia DNS untuk memposting tantangan bagi manajer sertifikat untuk mengeluarkan sertifikat. Tantangan ini meminta Anda	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk membuktikan bahwa Anda mengontrol DNS untuk nama domain Anda dengan menempatkan nilai tertentu dalam catatan TXT di bawah nama domain tersebut. Setelah Let's Encrypt memberikan token kepada klien ACME Anda, klien Anda membuat catatan TXT yang berasal dari token itu dan kunci akun Anda, dan itu menempatkan catatan itu di. `_acme-challenge.<YOURDOMAIN>` Kemudian Let's Encrypt menanyakan DNS untuk catatan itu. Jika menemukan kecocokan, Anda dapat melanjutkan untuk mengeluarkan sertifikat.	

Konfigurasikan peran IAM untuk memungkinkan pengelola sertifikat mengakses zona yang dihosting publik

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM untuk cert-manager.	Kebijakan IAM diperlukan untuk memberikan izin kepada manajer sertifikat untuk memvalidasi bahwa Anda memiliki domain Route 53. `policy.json` Contoh kebijakan IAM disediakan di `1-IAMRole` direktori dalam GitHub End-to-end enkripsi kloning di repositori Amazon EKS. Masukkan perintah berikut di AWS CLI untuk membuat kebijakan IAM. `aws iam create-policy \ --policy-name PolicyForCertManager \ --policy-document file://policy.json`	AWS DevOps
Buat peran IAM untuk cert-manager.	Setelah membuat kebijakan IAM, Anda harus membuat peran IAM. `trustpolicy.json` Contoh peran IAM disediakan di `1-IAMRole` direktori. Masukkan perintah berikut di AWS CLI untuk membuat peran IAM.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws iam create-role \ --role-name RoleForCertManager \ --assume-role-policy-document file://trustpolicy.json`	
Lampirkan kebijakan pada peran tersebut.	Masukkan perintah berikut di AWS CLI untuk melampirkan kebijakan IAM ke peran IAM. Ganti `AWS_ACCOUNT_ID` dengan ID akun AWS Anda. `aws iam attach-role-policy \ --policy-arn arn:aws:iam::AWS_ACCOUNT_ID:policy/PolicyForCertManager \ --role-name RoleForCertManager`	AWS DevOps

Siapkan NGINX Ingress Controller di Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan NGINX Ingress Controller.	Instal versi terbaru `nginx-ingress` menggunakan Helm. Anda dapat memodifikasi `nginx-ingress` konfigurasi sesuai dengan kebutuhan Anda sebelum menerapkannya. Pola ini menggunakan Network Load Balancer beranotasi dan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menghadap internal dan yang tersedia di direktori. `5-Nginx-Ingress-Controller` Instal NGINX Ingress Controller dengan menjalankan perintah Helm berikut dari direktori. `5-Nginx-Ingress-Controller` `helm install test-nginx nginx-stable/nginx-ingress -f 5-Nginx-Ingress-Controller/values_internal_nlb.yaml`	
Verifikasi bahwa NGINX Ingress Controller diinstal.	Masukkan `helm list` perintah. Output harus menunjukkan bahwa NGINX Ingress Controller diinstal.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat catatan Route 53 A.	Catatan A menunjuk ke Network Load Balancer yang dibuat oleh NGINX Ingress Controller. 1. Dapatkan nama DNS dari Network Load Balancer. Untuk petunjuk, lihat Mendapatkan nama DNS untuk penyeimbang beban ELB. 2. Di konsol Amazon Route 53, pilih Zona yang Dihosting. 3. Pilih zona yang dihosting publik tempat Anda ingin membuat rekaman, lalu pilih Buat catatan. 4. Masukkan nama untuk catatan. 5. Dalam tipe Rekam, pilih A - Rute lalu lintas ke IPv4 dan beberapa sumber daya AWS. 6. Aktifkan Alias. 7. Dalam Rute lalu lintas ke, lakukan hal berikut: a. Pilih Alias ke Network Load Balancer. b. Pilih Wilayah AWS tempat Network Load Balancer digunakan.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	c. Masukkan nama DNS Network Load Balancer. 8. Pilih Create records (Buat catatan).	

Siapkan NGINX di VirtualServer Amazon EKS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan NGINX VirtualServer.	VirtualServer Sumber daya NGINX adalah konfigurasi load balancing yang merupakan alternatif dari sumber daya ingress. Konfigurasi untuk membuat VirtualServer sumber daya NGINX tersedia dalam `nginx_virtualserver.yaml` file di direktori. `6-Nginx-Virtual-Server` Masukkan perintah berikut `kubectl` untuk membuat sumber daya NGINX VirtualServer . `kubectl apply -f nginx_virtualserver.yaml` **Important** Pastikan Anda memperbarui nama domain aplikasi, rahasia sertifikat, dan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nama layanan aplikasi dalam `nginx_virtualserver.yaml` file.	
Verifikasi bahwa NGINX VirtualServer dibuat.	Masukkan perintah berikut `kubectl` untuk memverifikasi bahwa VirtualServer sumber daya NGINX berhasil dibuat. `kubectl get virtualserver` **Note** Verifikasi bahwa `Host` kolom tersebut cocok dengan nama domain aplikasi Anda.	AWS DevOps
Menyebarkan server web NGINX dengan TLS diaktifkan.	Pola ini menggunakan server web NGINX dengan TLS diaktifkan sebagai aplikasi untuk menguji enkripsi. end-to-end File konfigurasi yang diperlukan untuk menyebarkan aplikasi pengujian tersedia di `demo-webserver` direktori. Masukkan perintah berikut `kubectl` untuk menyebarkan aplikasi uji. `kubectl apply -f nginx-tls-ap.yaml`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa sumber daya aplikasi pengujian dibuat.	Masukkan perintah berikut `kubectl` untuk memverifikasi bahwa sumber daya yang diperlukan dibuat untuk aplikasi pengujian: • `kubectl get deployments` **Note** Validasi `Ready` kolom dan `Available` kolom. • `kubectl get pods | grep -i example-deploy` **Note** Pod harus dalam `running` keadaan. • `kubectl get configmap` • `kubectl get svc`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi aplikasi.	1. Masukkan perintah berikut dengan mengganti `<application-domain-name>` dengan nama DNS Route53 yang Anda buat sebelumnya. `curl --verbose https://<application-domain-name>` 2. Verifikasi bahwa Anda dapat mengakses aplikasi.	AWS DevOps

Sumber daya terkait

Sumber daya AWS

- Membuat rekaman dengan menggunakan konsol Amazon Route 53 (dokumentasi Amazon Route 53)
- Menggunakan Network Load Balancer dengan pengontrol ingress NGINX di Amazon EKS (postingan blog AWS)

Sumber daya lainnya

- Route 53 (dokumentasi pengelola sertifikat)
- Mengkonfigurasi Penyedia Tantangan DNS01 (dokumentasi manajer sertifikat)
- Mari mengenkripsi tantangan DNS (Let's Encrypt documentation)

Sederhanakan penerapan aplikasi multi-tenant Amazon EKS dengan menggunakan Flux

Nadeem Rahaman, Aditya Ambati, Aniket Dekate, dan Shrikant Patil, Amazon Web Services

Ringkasan

Banyak perusahaan yang menawarkan produk dan layanan adalah industri yang diatur data yang diperlukan untuk mempertahankan hambatan data antara fungsi bisnis internal mereka. Pola ini menjelaskan bagaimana Anda dapat menggunakan fitur multi-tenancy di Amazon Elastic Kubernetes Service (Amazon EKS) untuk membangun platform data yang mencapai isolasi logis dan fisik antara penyewa atau pengguna yang berbagi satu kluster Amazon EKS. Pola memberikan isolasi melalui pendekatan berikut:

- Isolasi namespace Kubernetes
- Kontrol akses berbasis peran (RBAC)
- Kebijakan jaringan
- Kuota sumber daya
- AWS Identity and Access Management (IAM) peran untuk akun layanan (IRSA)

Selain itu, solusi ini menggunakan Flux untuk menjaga konfigurasi penyewa tetap tidak berubah saat Anda menerapkan aplikasi. Anda dapat menerapkan aplikasi penyewa Anda dengan menentukan repositori penyewa yang berisi file Flux dalam konfigurasi Anda. `kustomization.yaml`

Pola ini mengimplementasikan yang berikut:

- AWS CodeCommit Repositori, AWS CodeBuild proyek, dan AWS CodePipeline pipeline, yang dibuat dengan menerapkan skrip Terraform secara manual.
- Jaringan dan komponen komputasi yang diperlukan untuk hosting penyewa. Ini dibuat melalui CodePipeline dan CodeBuild dengan menggunakan Terraform.
- Ruang nama penyewa, kebijakan jaringan, dan kuota sumber daya, yang dikonfigurasi melalui bagan Helm.
- Aplikasi milik penyewa yang berbeda, digunakan dengan menggunakan Flux.

Kami menyarankan Anda merencanakan dan membangun arsitektur Anda sendiri dengan hati-hati untuk multi-tenancy berdasarkan persyaratan unik dan pertimbangan keamanan Anda. Pola ini memberikan titik awal untuk implementasi Anda.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Command Line Interface (AWS CLI) versi 2.11.4 atau yang lebih baru, diinstal dan dikonfigurasi
- Terraform versi 0.12 atau yang lebih baru diinstal pada mesin lokal Anda
- Terraform AWS Provider versi 3.0.0 atau yang lebih baru
- Penyedia Kubernetes versi 2.10 atau yang lebih baru
- Helm Provider versi 2.8.0 atau yang lebih baru
- Kubectl Provider versi 1.14 atau yang lebih baru

Batasan

- Ketergantungan pada penerapan manual Terraform: Penyiapan awal alur kerja, termasuk membuat CodeCommit repositori, CodeBuild proyek, dan CodePipeline saluran pipa, bergantung pada penerapan Terraform manual. Ini memperkenalkan batasan potensial dalam hal otomatisasi dan skalabilitas, karena memerlukan intervensi manual untuk perubahan infrastruktur.
- CodeCommit ketergantungan repositori: Alur kerja bergantung pada CodeCommit repositori sebagai solusi manajemen kode sumber dan digabungkan dengan erat. Layanan AWS

Arsitektur

Arsitektur target

Pola ini menyebarkan tiga modul untuk membangun infrastruktur pipeline, jaringan, dan komputasi untuk platform data, seperti yang diilustrasikan dalam diagram berikut.

Arsitektur pipa:

Arsitektur jaringan:

Arsitektur komputasi:

Alat

Layanan AWS

- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Transit Gatewayadalah hub pusat yang menghubungkan virtual private cloud (VPCs) dan jaringan lokal.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat-alat lainnya

- Kebijakan Jaringan Cilium mendukung kebijakan jaringan Kubernetes L3 dan L4. Mereka dapat diperluas dengan kebijakan L7 untuk menyediakan keamanan tingkat API untuk HTTP, Kafka, dan gRPC, dan protokol serupa lainnya.
- Flux adalah alat pengiriman berkelanjutan (CD) berbasis Git yang mengotomatiskan penerapan aplikasi di Kubernetes.
- Helm adalah pengelola paket open source untuk Kubernetes yang membantu Anda menginstal dan mengelola aplikasi di klaster Kubernetes Anda.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di repositori Solusi GitHub Terraform Multi-Tenancy EKS.

Praktik terbaik

Untuk pedoman dan praktik terbaik untuk menggunakan implementasi ini, lihat berikut ini:

- Praktik terbaik multi-tenancy Amazon EKS
- Dokumentasi fluks

Epik

Buat pipeline untuk tahap build, test, dan deploy Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori proyek.	Kloning repositori Solusi Terraform Multi-Tenancy GitHub EKS dengan menjalankan perintah berikut di jendela terminal: `git clone https://github.com/aws-samples/aws-eks-multitenancy-deployment.git`	AWS DevOps
Bootstrap bucket Terraform S3 dan Amazon DynamoDB.	1. Di `bootstrap` folder, buka `bootstrap.sh` file dan perbarui nilai variabel untuk nama bucket S3, nama tabel DynamoDB, dan: Wilayah AWS `S3_BUCKET_NAME="<S3_BUCKET_NAME>"`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` DYNAMODB_TABLE_NA ME="<DYNAMODB_NAME >" REGION="<AWS_REGI ON>" ``` 2. Jalankan `bootstrap.sh` skrip. Skrip membutuhk an AWS CLI, yang Anda instal sebagai bagian dari prasyarat. ``` cd bootstrap ./bootstrap.sh ```	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui `locals.tf` file `run.sh` dan.	1. Setelah proses bootstrap selesai dengan sukses, salin bucket S3 dan nama tabel DynamoDB dari bagian skrip: `variables bootstrap.sh`
	`# Variables S3_BUCKET_NAME="< S3_BUCKET_NAME>" DYNAMODB_TABLE_NAME ="<DYNAMODB_NAME"`
	2. Tempelkan nilai-nilai tersebut ke `run.sh` skrip, yang ada di direktori root proyek:
	`BACKEND_BUCKET_ID= "<SAME_NAME_AS_S3_ BUCKET_NAME>" DYNAMODB_ID=" <SAME_NAME_AS_DYNA MODB_NAME>"`
	3. Unggah kode proyek ke CodeCommit repositori. Anda dapat secara otomatis membuat repositori ini melalui Terraform dengan menyetel variabel berikut ke `true` dalam file: `demo/ pipeline/locals.tf`
	`create_new_repo = true`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Perbarui `locals.tf` file sesuai dengan kebutuhan Anda untuk membuat sumber daya pipa.	

Menyebarkan modul pipa.	Untuk membuat sumber daya pipeline, jalankan perintah Terraform berikut secara manual. Tidak ada orkestrasi untuk menjalankan perintah ini secara otomatis.
	`./run.sh -m pipeline -e demo -r <AWS_REGION> -t init`
	`./run.sh -m pipeline -e demo -r <AWS_REGION> -t plan`
	`./run.sh -m pipeline -e demo -r <AWS_REGION> -t apply`
		AWS DevOps

Buat infrastruktur jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai pipa.	1. Dalam `templates` folder, pastikan bahwa `buildspec` file memiliki variabel berikut diatur ke`network`: `TF_MODULE_TO_BUILD: "network"`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di CodePipeline konsol, pada halaman detail pipeline, mulai pipeline dengan memilih Rilis perubahan. Setelah proses pertama ini, pipeline dimulai secara otomatis setiap kali Anda melakukan perubahan ke cabang utama CodeCommit repositori. Pipa meliputi tahapan berikut: • `validate`menginisialisasi Terraform, menjalankan pemindaian keamanan Terraform dengan menggunakan alat checkov dan tfsec, dan mengungga h laporan pemindaian ke bucket S3. • `plan` menunjukkan paket Terraform dan mengunggah paket ke bucket S3. • `apply`menerapkan output paket Terraform dari bucket S3 dan membuat sumber daya. AWS • `destroy`menghapus AWS sumber daya yang dibuat selama `apply` tahap. Untuk mengaktifkan tahap	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	opsional ini, atur variabel berikut ke `true` dalam `demo/pipeline/locals.tf` file: `enable_destroy_stage = true`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi sumber daya yang dibuat melalui modul jaringan.	Konfirmasikan bahwa AWS sumber daya berikut dibuat setelah pipeline berhasil diterapkan: • VPC jalan keluar dengan tiga subnet publik dan tiga subnet pribadi, gateway internet, dan gateway NAT. • VPC Amazon EKS dengan tiga subnet pribadi. • Penyewa 1 dan Penyewa 2 VPCs dengan masing-ma sing tiga subnet pribadi. • Gateway transit dengan semua lampiran VPC dan rute ke setiap subnet pribadi. • Rute gateway transit statis untuk VPC jalan keluar Amazon EKS dengan blok CIDR tujuan. `0.0.0.0/0` Ini diperlukan untuk memungkinkan semua memiliki VPCs akses internet keluar melalui VPC jalan keluar Amazon EKS.	AWS DevOps

Buat infrastruktur komputasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui `locals.tf` untuk mengaktifkan akses CodeBuild proyek ke VPC.	Untuk menyebarkan add-on untuk kluster pribadi Amazon EKS, CodeBuild proyek harus dilampirkan ke VPC Amazon EKS.
	1. Di `demo/pipeline` folder, buka `locals.tf` file, dan atur `vpc_enabled` variabel ke`true`.
	2. Jalankan `run.sh` skrip untuk menerapkan perubahan pada modul pipeline:
	`demo/pipeline/locals.tf`
	`./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd init`
	`./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd plan`
	`./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd apply`
		AWS DevOps

Perbarui `buildspec` file untuk membangun modul komputasi.	Di `templates` folder, di semua file `buildspec` YAMM, atur nilai `TF_MODULE_TO_BUILD` variabel dari `network` ke`compute`:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`TF_MODULE_TO_BUILD: "compute"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui `values` file untuk bagan Helm manajemen penyewa.	1. Buka `values.yaml` file di lokasi berikut:
	`cd cfg-terraform/demo/compute/cfg-tenant-mgmt`
	File terlihat seperti ini:
	`---`
	`global:`
	`  clusterRoles:`
	`  operator: platform-tenant`
	`  flux: flux-tenant-applier`
	`  flux:`
	`  tenantCloneBaseUrl: ${TEANT_BASE_URL}`
	`  repoSecret: ${TENANT_REPO_SECRET}`
	`tenants:`
	`  tenant-1:`
	`  quotas:`
	`  limits:`
	`  cpu: 1`
	`  memory: 1Gi`
	`  flux:`
	`  path: overlays/tenant-1`
	`  tenant-2:`
	`  quotas:`
	`  limits:`
	`  cpu: 1`
	`  memory: 2Gi`
	`  flux:`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>path: overlays/
	tenant-2</pre>
	2. Di `tenants` bagian `global` dan, perbarui konfigurasi berdasarkan kebutuhan Anda:
	• `tenantCloneBaseUrl` — Jalur ke repositori yang menampung kode untuk semua penyewa (kami menggunakan repositori Git yang sama untuk semua penyewa)
	• `repoSecret` - Rahasia Kubernetes yang menyimpan kunci SSH dan host yang dikenal untuk mengautentikasi ke repositori Git penyewa global
	• `quotas`— Kubernetes kuota sumber daya yang ingin Anda terapkan untuk setiap penyewa
	• `flux path`— Jalur ke file YAMM aplikasi penyewa di repositori penyewa global	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi sumber daya komputasi.	Setelah Anda memperbarui file di langkah sebelumnya, CodePipeline mulai secara otomatis. Konfirmasikan bahwa itu membuat AWS sumber daya berikut untuk infrastruktur komputasi: • Cluster Amazon EKS dengan titik akhir pribadi • Node pekerja Amazon EKS • Pengaya Amazon EKS: rahasia eksternal,`aws-loadbalancer-controller` , dan `metrics-server` • GitOps modul, bagan Helm Flux, bagan Helm Cilium, dan bagan helm manajemen penyewa	AWS DevOps

Periksa manajemen penyewa dan sumber daya lainnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi sumber daya manajemen penyewa di Kubernetes.	Jalankan perintah berikut untuk memeriksa apakah sumber daya manajemen penyewa berhasil dibuat dengan bantuan Helm. 1. Namespace penyewa telah dibuat, seperti	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	yang ditentukan dalam: `values.yaml`
	`kubectl get ns -A`
	2. Kuota ditetapkan untuk setiap namespace penyewa, seperti yang ditentukan dalam: `values.yaml`
	`kubectl get quota --namespace=<tenant_namespace>`
	3. Detail kuota benar untuk setiap namespace penyewa:
	`kubectl describe quota cpu-memory-resource-quota-limit -n <tenant_namespace>`
	4. Kebijakan Jaringan Cilium diterapkan ke setiap namespace penyewa:
	`kubectl get CiliumNetworkPolicy -A`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi penerapan aplikasi penyewa.	Jalankan perintah berikut untuk memverifikasi bahwa aplikasi penyewa telah digunakan.
	1. Flux dapat terhubung ke CodeCommit repositor i yang ditentukan dalam modul: GitOps
	`kubectl get gitreposi tories -A`
	2. Pengontrol kustomisasi Flux telah menerapkan file YAMAL di repositori: CodeCommit
	`kubectl get kustomiza tions -A`
	3. Semua sumber daya aplikasi digunakan di ruang nama penyewa mereka:
	`kubectl get all -n <tenant_namespace>`
	4. Ingress telah dibuat untuk setiap penyewa:
	`kubectl get ingress -n <tenant_namespace>`	

Pemecahan masalah

Isu	Solusi

Anda menemukan pesan galat yang mirip dengan berikut ini:
 	
`Failed to checkout and determine revision: unable to clone unknown error: You have successfully authenticated over SSH. You can use Git to interact with AWS CodeCommit.`
	Ikuti langkah-langkah ini untuk memecahkan masalah:
	
	1. Verifikasi repositori aplikasi penyewa: Repositori kosong atau salah konfigurasi mungkin menyebabkan kesalahan. Pastikan repositori aplikasi penyewa berisi kode yang diperlukan.
	2. Menerapkan ulang `tenant_mgmt` modul: Dalam file konfigurasi `tenant_mgmt` modul, cari `app` bloknya, lalu atur `deploy` parameternya ke: `0`
	
	`deploy = 0`
	
	Setelah Anda menjalankan `apply` perintah Terraform, ubah nilai `deploy` parameter kembali ke: 1
	
	`deploy = 1`
	
	3. Periksa kembali status: Setelah Anda menjalankan langkah-langkah sebelumnya, gunakan perintah berikut untuk memeriksa apakah masalah tetap ada:
	
	`kubectl get gitrepositories -A`
	
	Jika terus berlanjut, pertimbangkan untuk menyelam lebih dalam ke log Flux untuk detail lebih lanjut atau lihat panduan pemecahan masalah umum Flux.

Sumber daya terkait

- Cetak Biru Amazon EKS untuk Terraform
- Panduan Praktik Terbaik Amazon EKS, bagian Multi-penyewaan
- Situs Flux
- Situs Helm

Informasi tambahan

Berikut adalah contoh struktur repositori untuk menerapkan aplikasi penyewa:

```
applications
sample_tenant_app
### README.md
### base
#   ### configmap.yaml
#   ### deployment.yaml
#   ### ingress.yaml
#   ### kustomization.yaml
#   ### service.yaml
### overlays
    ### tenant-1
    #   ### configmap.yaml
    #   ### deployment.yaml
    #   ### kustomization.yaml
    ### tenant-2
        ### configmap.yaml
        ### kustomization.yaml
```

Merampingkan pengembangan dan penerapan bot Amazon Lex dengan menggunakan alur kerja otomatis

Balaji Panneerselvam, Attila Dancso, Pavan Dusanapudi, Anand Jumnani, dan James O'Hara, Amazon Web Services

Ringkasan

Mengembangkan dan menerapkan bot percakapan Amazon Lex dapat menjadi tantangan saat Anda mencoba mengelola beberapa fitur, pengembang, dan lingkungan. Alur kerja otomatis menggunakan prinsip infrastruktur sebagai kode (IAc) dapat membantu merampingkan proses. Pola ini dapat membantu meningkatkan produktivitas pengembang Amazon Lex dan memungkinkan manajemen siklus hidup bot yang efisien dengan cara berikut:

- Aktifkan pengembangan beberapa fitur secara bersamaan - Dengan alur kerja otomatis, pengembang dapat mengerjakan berbagai fitur secara paralel di cabang terpisah. Perubahan kemudian dapat digabungkan dan digunakan tanpa memblokir pekerjaan lain.
- Gunakan UI konsol Amazon Lex - Pengembang dapat menggunakan konsol Amazon Lex yang mudah digunakan untuk membuat dan menguji bot. Bot kemudian dijelaskan dalam kode infrastruktur untuk penerapan.
- Promosikan bot di seluruh lingkungan - Alur kerja mengotomatiskan mempromosikan versi bot dari lingkungan yang lebih rendah seperti pengembangan dan pengujian hingga produksi. Pendekatan ini mengurangi risiko dan overhead promosi manual.
- Pertahankan kontrol versi - Mengelola definisi bot di Git, bukan hanya melalui layanan Amazon Lex memberi Anda kontrol versi dan jejak audit. Perubahan dilacak ke pengembang individu, tidak seperti ketika hanya menggunakan Konsol Manajemen AWS atau APIs untuk memodifikasi bot yang disimpan di dalamnya. AWS

Dengan mengotomatiskan proses rilis bot Amazon Lex, tim dapat memberikan fitur lebih cepat dengan mengurangi risiko dan usaha. Bot tetap di bawah kontrol versi daripada diisolasi di konsol Amazon Lex.

Prasyarat dan batasan

Prasyarat

- Alur kerja melibatkan beberapa Akun AWS untuk lingkungan yang berbeda (pengembangan, produksi, dan DevOps), yang memerlukan manajemen akun dan konfigurasi akses lintas akun.

- Python 3.9 tersedia di lingkungan penerapan atau pipeline Anda.
- Git diinstal dan dikonfigurasi pada workstation lokal untuk kontrol sumber.
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi untuk mengautentikasi dengan menggunakan baris perintah atau Python.

Keterbatasan

- Akses repositori — Alur kerja mengasumsikan bahwa pipeline continuous integration and continuous delivery (CI/CD) memiliki izin yang diperlukan untuk melakukan perubahan pada repositori kode sumber.
- Versi bot awal — Perkakas mengharuskan versi awal bot digunakan dengan menggunakan AWS CloudFormation templat. Anda harus membuat iterasi pertama bot dan mengkomitmennya ke repo sebelum alur kerja otomatis dapat mengambil alih.
- Konflik gabungan — Meskipun alur kerja bertujuan untuk memungkinkan pengembangan bersamaan, masih ada kemungkinan konflik gabungan saat mengintegrasikan perubahan dari cabang yang berbeda. Menyelesaikan konflik dalam konfigurasi bot mungkin memerlukan intervensi manual.

Versi produk

- Python 3.9 atau lebih tinggi
- AWS CDK v2 2.124.0 atau lebih tinggi
- AWS SDK for Python (Boto3)1.28 atau di atas

Arsitektur

Diagram berikut menampilkan arsitektur tingkat tinggi dan komponen kunci dari solusi.

Komponen utama meliputi:

- Lex bot repo — Sebuah repositori Git yang menyimpan definisi IAc untuk bot Amazon Lex.
- DevOps— Akun AWS Didedikasikan untuk menampung CI/CD jaringan pipa dan sumber daya terkait untuk proses pengembangan dan penyebaran.

- Pipelines — AWS CodePipeline Contoh yang mengotomatiskan berbagai tahap pengembangan bot dan siklus hidup penerapan, seperti membuat bot baru, mengekspor definisi bot, mengimpor definisi bot, dan menghapus bot.
- Bot tiket dan bot utama — Sumber daya bot Amazon Lex, di mana bot tiket adalah bot khusus fitur yang dikembangkan oleh tim atau pengembang individu dan bot utama adalah bot dasar yang mengintegrasikan semua fitur.

Diagram arsitektur menggambarkan alur kerja berikut:

1. Bot utama dasar - Titik awal alur kerja adalah untuk mendasarkan bot utama di lingkungan pengembangan (Dev). Bot utama berfungsi sebagai fondasi untuk pengembangan masa depan dan penambahan fitur.
2. Buat bot tiket — Ketika fitur atau perubahan baru diperlukan, bot tiket dibuat. Bot tiket pada dasarnya adalah salinan atau cabang dari bot utama yang dapat dikerjakan pengembang tanpa memengaruhi versi utama.
3. Bot tiket ekspor - Setelah bekerja pada bot tiket selesai, itu diekspor dari layanan Amazon Lex. Kemudian, cabang yang berisi bot tiket didasarkan ulang dari cabang utama. Langkah ini memastikan bahwa setiap perubahan yang dilakukan pada bot utama saat bot tiket sedang dalam pengembangan dimasukkan, mengurangi potensi konflik.
4. Impor bot tiket berbasis ulang dan validasi - Bot tiket berbasis ulang diimpor kembali ke lingkungan pengembangan dan divalidasi untuk memastikannya berfungsi dengan benar dengan perubahan terbaru dari cabang utama. Jika validasi berhasil, permintaan tarik (PR) dibuat untuk menggabungkan perubahan bot tiket ke cabang utama.
5. Hapus bot tiket — Setelah perubahan berhasil digabungkan ke cabang utama, bot tiket tidak lagi diperlukan. Bot tiket dapat dihapus untuk menjaga lingkungan tetap bersih dan mudah dikelola.
6. Terapkan bot utama ke lingkungan pengembangan dan uji — Bot utama yang diperbarui, sekarang termasuk fitur atau perubahan baru, diterapkan ke lingkungan pengembangan. Di sini, ia menjalani pengujian menyeluruh untuk memastikan semua fungsi berfungsi seperti yang diharapkan.
7. Menyebarkan bot utama ke lingkungan produksi - Setelah pengujian di lingkungan pengembangan selesai dan berhasil, bot utama dikerahkan ke lingkungan produksi. Langkah ini adalah tahap akhir dari alur kerja, di mana fitur-fitur baru menjadi tersedia untuk pengguna akhir.

Otomatisasi dan skala

Alur kerja otomatis memungkinkan pengembang untuk mengerjakan berbagai fitur secara paralel, masing-masing di cabang terpisah. Ini memfasilitasi pengembangan bersamaan, memungkinkan tim untuk berkolaborasi secara efektif dan memberikan fitur lebih cepat. Dengan cabang yang terisolasi satu sama lain, perubahan dapat digabungkan dan digunakan tanpa memblokir atau mengganggu pekerjaan lain yang sedang berlangsung.

Alur kerja mengotomatiskan penerapan dan promosi versi bot di berbagai lingkungan, seperti pengembangan, pengujian, dan produksi.

Menyimpan definisi bot dalam sistem kontrol versi seperti Git menyediakan jejak audit yang komprehensif dan memungkinkan kolaborasi yang efisien. Perubahan dilacak ke pengembang individu, memastikan transparansi dan akuntabilitas di seluruh siklus pengembangan. Pendekatan ini juga memfasilitasi tinjauan kode, memungkinkan tim untuk mengidentifikasi dan mengatasi masalah sebelum menerapkan ke produksi.

Dengan menggunakan AWS CodePipeline dan lainnya Layanan AWS, alur kerja otomatis dapat diskalakan untuk mengakomodasi peningkatan beban kerja dan ukuran tim.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak open-source untuk mendefinisikan AWS Cloud infrastruktur dalam kode dengan menggunakan bahasa pemrograman yang sudah dikenal dan menyediakannya. CloudFormation Implementasi sampel dalam pola ini menggunakan Python.
- AWS CDK Command Line Interface (AWS CDK CLI) - AWS CDK Toolkit adalah alat utama untuk berinteraksi dengan aplikasi Anda. AWS CDK Ini mengeksekusi aplikasi Anda, menginterogasi model aplikasi yang Anda tentukan, dan menghasilkan serta menerapkan CloudFormation template yang dihasilkan oleh CDK.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS Pola ini digunakan CloudFormation untuk menyebarkan konfigurasi bot Amazon Lex dan sumber daya terkait menggunakan infrastruktur sebagai kode.
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan. Pola ini digunakan CodeBuild untuk membangun dan mengemas artefak penyebaran.

- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus. Pola ini digunakan CodePipeline untuk mengatur pipa pengiriman kontinu.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Lex V2 adalah Layanan AWS untuk membangun antarmuka percakapan (bot) untuk aplikasi menggunakan suara dan teks.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS

Alat lainnya

- Git adalah sistem kontrol versi terdistribusi open source.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub management-framework-sample-for-amazon-lex. Kode repo berisi folder dan file berikut:

- `prerequisite`folder - Berisi definisi CloudFormation tumpukan (menggunakan AWS CDK) untuk menyiapkan sumber daya dan lingkungan yang diperlukan.
- `prerequisite/lexmgmtworkflow`folder — Direktori utama untuk proyek Alur Kerja Manajemen Lex, termasuk definisi tumpukan dan kode Python.
- `prerequisite/tests`— Berisi unit test.
- `src`folder — Direktori kode sumber, termasuk pembungkus dan utilitas manajemen bot Amazon Lex.
- `src/dialogue_lambda`— Direktori kode sumber dari fungsi Lambda kait dialog yang mencegat dan memproses input pengguna selama percakapan dengan bot Amazon Lex.

Praktik terbaik

- Pemisahan kekhawatiran
  - Mempertahankan pemisahan tanggung jawab yang jelas antara lingkungan DevOps, pengembangan, dan produksi.
  - Gunakan terpisah Akun AWS untuk setiap lingkungan untuk menegakkan batasan isolasi dan keamanan yang tepat.
  - Gunakan peran lintas akun dan prinsip akses paling tidak memiliki hak istimewa untuk memastikan akses terkontrol antar lingkungan.
- Infrastruktur sebagai kode
  - Tinjau dan perbarui kode infrastruktur secara teratur agar selaras dengan praktik terbaik dan persyaratan yang terus berkembang.
  - Menetapkan strategi percabangan dan penggabungan yang jelas untuk repositori kode sumber
- Pengujian dan validasi
  - Menerapkan pengujian otomatis pada berbagai tahap pipeline untuk menangkap masalah di awal siklus pengembangan.
  - Gunakan konsol Amazon Lex atau kerangka kerja pengujian otomatis untuk memvalidasi konfigurasi dan fungsionalitas bot sebelum mempromosikan ke lingkungan yang lebih tinggi.
  - Pertimbangkan untuk menerapkan gerbang persetujuan manual untuk penyebaran ke produksi atau lingkungan kritis.
- Pemantauan dan pencatatan
  - Siapkan mekanisme pemantauan dan pencatatan untuk pipeline, penerapan, dan interaksi bot.
  - Pantau peristiwa pipeline, status penerapan, dan metrik kinerja bot untuk mengidentifikasi dan mengatasi masalah dengan segera.
  - Gunakan layanan AWS seperti Amazon CloudWatch AWS CloudTrail, dan AWS X-Ray untuk pencatatan dan pemantauan terpusat.
  - Secara teratur meninjau dan menganalisis kinerja, efisiensi, dan efektivitas alur kerja otomatis.
- Keamanan dan kepatuhan
  - Terapkan praktik pengkodean yang aman dan ikuti praktik terbaik AWS keamanan untuk pengembangan dan penerapan bot Amazon Lex.
  - Tinjau dan perbarui peran, kebijakan, dan izin IAM secara berkala agar selaras dengan prinsip hak istimewa paling sedikit.

- Pertimbangkan untuk mengintegrasikan pemindaian keamanan dan pemeriksaan kepatuhan ke dalam jaringan pipa.

Epik

Mengatur IAc untuk manajemen bot Amazon Lex

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan lingkungan CDK lokal.	1. Untuk mengkloning repo pola ini dan menavigasi ke `prerequisite` direktori, jalankan perintah berikut:
	`git clone https://github.com/aws-samples/management-framework-sample-for-amazon-lex.git`
	`cd management-framework-sample-for-amazon-lex`
	2. Untuk menginstal dan mengaktifkan lingkungan virtual Python, jalankan perintah berikut yang menginstal dependensi CDK secara lokal di folder proyek, bukan secara global:
	`pip install virtualenv`
	`python<version> -m venv .venv`
	`source .venv/bin/activate`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`python -m pip install -r requirements.txt`	

Buat peran lintas akun di `devops` lingkungan.	`devops`Akun bertanggung jawab untuk hosting dan mengelola CI/CD jaringan pipa. Untuk mengaktifkan CI/CD pipeline berintera ksi dengan `dev` dan `prod` lingkungan, jalankan perintah berikut untuk membuat peran lintas akun di `devops` akun.
	`cdk bootstrap --profile =devops`
	
	`cdk deploy LexMgmtDe vopsRoleStack -c dev-account-id=222 2222222222 -c prod- account-id=33333333333 3 --profile=devops`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat peran lintas akun di `dev` lingkungan.	Buat peran IAM di `dev` akun dengan izin yang diperlukan untuk memungkinkan `devops` akun mengambil peran ini. CI/CD Pipeline menggunakan peran ini untuk melakukan tindakan di `dev` akun, seperti menerapkan dan mengelola sumber daya bot Amazon Lex.
	
	Untuk membuat peran IAM, jalankan perintah berikut:
	
	`cdk bootstrap --profile =dev`
	
	`cdk deploy LexMgmtCr ossaccountRoleStac k -c devops-account- id=1111111111111 -- profile=dev`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat peran lintas akun di `prod` lingkungan.	Buat peran IAM di `prod` akun dengan izin yang diperlukan untuk memungkinkan `devops` akun mengambil peran ini. CI/CD Pipeline menggunakan peran ini untuk melakukan tindakan di `prod` akun, seperti menerapkan dan mengelola sumber daya bot Amazon Lex.
	`cdk bootstrap --profile =prod`
	
	`cdk deploy LexMgmtCrossaccountRoleStack -c devops-account-id=111111111111 --profile=prod`
		AWS DevOps

Buat jaringan pipa di `devops` lingkungan.	Untuk mengelola alur kerja pengembangan untuk bot Amazon Lex, jalankan perintah berikut untuk menyiapkan pipeline di lingkungan. `devops` `cdk deploy LexMgmtWorkflowStack -c devops-account-id=111111111111 -c dev-account-id=2222222222222 -c prod-account-id=333333333333 --profile =devops`	AWS DevOps

Tetapkan baseline untuk bot utama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan versi awal bot utama.	Untuk menentukan versi awal bot utama, picu `BaselineBotPipeline` pipeline. Pipeline menyebarkan definisi bot dasar yang didefinisikan dalam CloudFormation template, mengekspor definisi bot utama sebagai file.json . dan menyimpan kode bot utama dalam sistem kontrol versi.	AWS DevOps

Menerapkan alur kerja pengembangan fitur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bot tiket untuk mengembangkan dan menguji fitur.	`TicketBot` adalah instance bot baru yang diimpor dari definisi bot utama yang ada di cabang fitur. Pendekatan ini memastikan bahwa bot baru memiliki semua fungsionalitas dan konfigurasi saat ini dari bot utama. Untuk menentukan versi awal bot tiket, picu `CreateTicketBotPipeline` pipeline. Pipeline membuat cabang fitur baru di sistem kontrol versi dan membuat instance bot	Lex Bot Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tiket baru berdasarkan bot utama.	
Kembangkan dan uji fitur bot tiket.	Untuk mengembangkan dan menguji fitur, masuk ke Konsol Manajemen AWS dan buka konsol Amazon Lex di https://console.aws.amazon.com/lex/. Untuk informasi selengkapnya, lihat Menguji bot menggunakan konsol di dokumentasi Amazon Lex. Dengan `TicketBot` instance, Anda sekarang dapat menambahkan, memodifikasi, atau memperluas fungsionalitas bot untuk mengimplementasikan fitur baru. Misalnya, Anda dapat membuat atau memodifikasi maksud, ucapan, slot, dan alur dialog. Untuk informasi selengkapnya, lihat Menambahkan maksud dalam dokumentasi Amazon Lex.	Lex Bot Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekspor definisi bot tiket.	Definisi bot yang diekspor pada dasarnya adalah representasi dari konfigurasi dan fungsionalitas bot dalam format JSON. Untuk mengekspor definisi bot tiket, picu `ExportTicketBotPipeline` pipeline. Pipeline mengekspor definisi bot tiket sebagai file.json dan menyimpan kode bot tiket di cabang fitur di sistem kontrol versi.	Lex Bot Pengembang
Rebase cabang fitur dari cabang utama terbaru.	Selama pengembangan fitur baru, cabang utama mungkin telah menerima perubahan lain dari pengembang atau tim yang berbeda. Untuk memasukkan perubahan ini ke dalam cabang fitur, lakukan `rebase` operasi Git. Operasi ini pada dasarnya memutar ulang komit dari cabang fitur di atas komit terbaru dari cabang utama, memastikan bahwa cabang fitur mencakup semua perubahan terbaru	Lex Bot Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Impor dan validasi bot tiket berbasis ulang.	Setelah Anda rebase cabang fitur, Anda harus mengimpornya ke instance bot tiket. Impor ini memperbarui bot tiket yang ada dengan perubahan terbaru dari cabang berbasis ulang. Untuk mengimpor bot tiket berbasis ulang, picu `ImportTicketBotPipeline` pipeline. Pipeline mengimpor file definisi.json ticket bot di cabang fitur di sistem kontrol versi ke dalam instance. `TicketBot`	Lex Bot Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi definisi bot berbasis ulang.	Setelah Anda mengimpor definisi bot berbasis ulang, sangat penting untuk memvalidasi fungsinya . Anda ingin memastikan bahwa fitur baru berfungsi seperti yang diharapkan dan tidak bertentangan dengan fungsionalitas yang ada. Validasi ini biasanya melibatkan pengujian bot dengan berbagai skenario masukan, memeriksa tanggapan, dan memverifikasi bahwa bot berperilaku sebagaimana dimaksud. Anda dapat melakukan validasi dengan salah satu cara berikut: • Uji bot secara manual dengan menggunakan konsol Amazon Lex. • Gunakan pendekatan otomatis dengan menggunakan kerangka kerja pengujian dan alat yang dapat mensimulasikan interaksi pengguna dan menegaskan respons yang diharapkan.	Lex Bot Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gabungkan cabang fitur ke cabang utama.	Setelah Anda mengembangkan dan menguji fitur baru dalam `TicketBot` instance terisolasi, lakukan hal berikut: 1. Komit perubahan ke cabang fitur yang sesuai dalam sistem kontrol versi. 2. Untuk menggabungkan cabang fitur ke cabang utama, buat permintaan tarik (PR). PR ini berfungsi sebagai permintaan untuk meninjau dan memasukkan perubahan ke dalam basis kode utama.	Lex Bot Developer, Adminstrator Repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus cabang fitur dan bot tiket.	Setelah cabang fitur berhasil digabungkan ke cabang utama, hapus cabang fitur dan bot tiket dari repo kode sumber. Untuk menghapus cabang fitur dan bot tiket, picu `DeleteTicketBotPipeline` pipeline. Pipeline menghapus sumber daya bot sementara yang dibuat selama proses pengembangan (misalnya, bot tiket). Tindakan ini membantu menjaga repo bersih dan mencegah kebingungan atau konflik dengan cabang fitur future.	Lex Bot Pengembang

Pertahankan bot utama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Impor definisi bot utama terbaru ke `dev` lingkungan.	Untuk mengimpor definisi bot utama terbaru di cabang utama ke `dev` lingkungan, picu `DeployBotDevPipeline` pipeline. Pipeline juga membuat tag git pada persetujuan.	AWS DevOps
Impor definisi bot utama terbaru ke `prod` lingkungan.	Untuk mengimpor definisi bot terbaru di cabang	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	utama ke `prod` lingkungan, berikan referensi tag dari tugas sebelumnya sebagai parameter dan picu `DeployBotProdPipeline` pipeline. Pipeline mengimpor definisi bot terbaru dari tag tertentu ke `prod` lingkungan.	

Pemecahan masalah

Isu	Solusi
Saat Anda menerapkan bot Amazon Lex ke berbagai Akun AWS, layanan perkakas harus memiliki izin yang diperlukan untuk mengakses sumber daya di akun tersebut.	Untuk memberikan akses lintas akun, gunakan peran dan kebijakan IAM. Buat peran IAM di akun target dan lampirkan kebijakan ke peran yang memberikan izin yang diperlukan. Kemudian, asumsikan peran ini dari akun tempat bot Amazon Lex digunakan. Untuk informasi selengkapnya, lihat izin IAM yang diperlukan untuk mengimpor dan izin IAM diperlukan untuk mengekspor bot di Lex V2 dalam dokumentasi Amazon Lex.

Sumber daya terkait

- Mengimpor bot di Amazon Lex V2
- Mulai pipa di CodePipeline
- Bekerja dengan bot Amazon Lex V2

Berlangganan beberapa titik akhir email ke topik SNS dengan menggunakan sumber daya khusus

Ricardo Morais, Amazon Web Services

Ringkasan

Catatan, Agustus 2022: AWS CloudFormation sekarang mendukung langganan beberapa sumber daya melalui AWS::SNS::Topicobjek dan atribut Langganannya.

Pola ini menjelaskan cara berlangganan beberapa alamat email untuk menerima notifikasi dari topik Amazon Simple Notification Service (Amazon SNS). Ini menggunakan fungsi AWS Lambda sebagai sumber daya khusus dalam template CloudFormation AWS. Fungsi Lambda dikaitkan dengan parameter input yang menentukan titik akhir email untuk topik SNS.

Saat ini, Anda dapat menggunakan objek CloudFormation template AWS AWS::SNS::Topicdan AWS::SNS::Subscriptionberlangganan titik akhir tunggal ke topik SNS. Untuk berlangganan beberapa titik akhir, Anda harus memanggil objek beberapa kali. Dengan menggunakan fungsi Lambda sebagai sumber daya khusus, Anda dapat berlangganan beberapa titik akhir melalui parameter input. Anda dapat menggunakan fungsi Lambda ini sebagai sumber daya khusus di template CloudFormation AWS apa pun.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Profil AWS yang dikonfigurasi di lingkungan lokal Anda dengan kunci akses dan kunci rahasia.
- Izin untuk hal berikut:
  - Peran dan kebijakan AWS Identity and Access Management (IAM)
  - Fungsi AWS Lambda
  - Amazon Simple Storage Service (Amazon S3) untuk mengunggah fungsi Lambda
  - Topik dan kebijakan Amazon SNS
  - AWS CloudFormation tumpukan

Batasan

- Kode ini mendukung workstation Linux dan macOS.

Versi produk

- AWS Command Line Interface (AWS CLI) versi 2 atau yang lebih baru.

Arsitektur

Tumpukan teknologi target

- AWS CloudFormation
- Amazon SNS
- AWS Lambda

Alat

Alat

- AWS CLI versi 2

Kode

Lampiran mencakup file-file berikut:

- Fungsi Lambda: `lambda_function.py`
- CloudFormation Templat AWS: `template.yaml`
- Dua file parameter untuk menangani beberapa atau satu langganan titik akhir email: `parameters-multiple-values.json` (digunakan sebagai default) dan `parameters-one-value.json`

Untuk menyebarkan tumpukan, Anda dapat menggunakan salah satu file parameter. Untuk menentukan beberapa titik akhir email:

```
./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION>
```

Untuk menentukan titik akhir email tunggal:

```
./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION> -f parameters-one-value.json
```

Epik

Opsi 1 - Menyebarkan topik SNS dengan satu langganan email

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan titik akhir email untuk langganan topik SNS.	Edit file `parameters-one-value.json` (terlampir), dan ubah nilai `pSNSNotificationsEmail` parameter untuk mencerminkan alamat email yang ingin Anda gunakan, seperti `someone@example.com` .	
Terapkan CloudFormation tumpukan AWS yang membuat sumber daya dan langganan.	Jalankan perintah deploy.sh dengan nama profil AWS, Wilayah AWS, dan `parameters-one-value.json` file. `./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION> -f parameters-one-value.json`	Peran IAM dengan izin yang tepat

Opsi 2 - Menyebarkan topik SNS dengan dua atau lebih langganan email

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan titik akhir email untuk langganan topik SNS.	Edit file `parameters-multiple-values.json` (terlampir), dan ubah nilai `pSNSNotificationsE`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`mail` parameter untuk mencerminkan alamat email yang ingin Anda gunakan, dipisahkan dengan koma, sebagai berikut:`someone1@example.com, someone2@example.com` .	
Terapkan CloudFormation tumpukan AWS yang membuat sumber daya dan langganan.	Jalankan perintah deploy.sh dengan nama profil AWS dan Wilayah AWS Anda. Anda tidak perlu menentukan `parameters-multiple-values.json` file karena digunakan secara default. `./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION>`	Peran IAM dengan izin yang tepat

Opsi 3 - Menyebarkan topik SNS melalui template AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat sebuah topik SNS.	Buat topik SNS melalui CloudFormation template AWS, tanpa menentukan titik akhir langganan di objek template. `AWS::SNS::Topic` Anda dapat menggunakan `template.yaml` lampiran sebagai titik awal.	Peran IAM dengan izin yang tepat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan topik SNS.	Buat kebijakan topik SNS di CloudFormation template AWS.	Peran IAM dengan izin yang tepat
Berlangganan daftar titik akhir email ke topik SNS.	Berdasarkan daftar titik akhir email (satu atau lebih), berlangganan titik akhir ke topik SNS yang Anda buat.	Peran IAM dengan izin yang tepat

Sumber daya terkait

Referensi

- Sumber daya CloudFormation khusus AWS (dokumentasi AWS)
- Pembuatan sumber daya CloudFormation khusus AWS dengan Python, AWS Lambda, dan crhelper (posting blog)

Alat yang dibutuhkan

- AWS CLI versi 2

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Koordinasikan ketergantungan sumber daya dan eksekusi tugas dengan menggunakan konstruksi AWS Fargate WaitCondition kait

Penggemar Stan, Amazon Web Services

Ringkasan

Pola ini menjelaskan paket WaitCondition hook (`waitcondition-hook-for-aws-fargate-task`) npm, yang merupakan solusi cloud-native yang dirancang untuk mengatur AWS Fargatetugas di kluster Amazon Elastic Container Service (Amazon ECS).

WaitCondition Hook adalah AWS Cloud Development Kit (AWS CDK) konstruksi yang secara khusus dirancang untuk integrasi dengan. AWS CloudFormation WaitCondition Hook menyediakan kemampuan kunci berikut:

- Bertindak sebagai mekanisme kondisi tunggu, menjeda eksekusi CloudFormation tumpukan hingga tugas Fargate tertentu selesai, yang membantu penerapan dan penyediaan sumber daya yang teratur.
- Mendukung TypeScript dan Python, membuatnya ideal untuk AWS CDK proyek.
- Memungkinkan pengembang dan arsitek untuk mengatur penerapan dengan mengoordinasikan penyelesaian tugas dan manajemen sumber daya untuk aplikasi kontainer di. AWS
- Memungkinkan menjalankan tugas Fargate dengan satu atau beberapa kontainer yang disematkan dalam CloudFormation siklus hidup. dan dapat menangani kegagalan tugas dan memutar kembali tumpukan setelah kegagalan tugas. CloudFormation
- Memberikan fleksibilitas untuk menambahkan dependensi antara sumber daya dan hasil eksekusi tugas Fargate, memungkinkan tugas khusus atau memanggil titik akhir lainnya. Misalnya, Anda dapat menjeda CloudFormation tumpukan dan menunggu migrasi database (dilakukan oleh tugas Fargate) dan menyediakan sumber daya lain yang mungkin bergantung pada keberhasilan migrasi database.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Cloud Development Kit (AWS CDK) Command Line Interface (CLI) diinstal pada workstation lokal. Untuk informasi lebih lanjut, lihat referensi AWS CDK CLI dalam dokumentasi. AWS CDK

- Node package manager (npm), diinstal pada workstation lokal dan dikonfigurasi untuk in.AWS CDK TypeScript Untuk informasi selengkapnya, lihat Downloading and installing Node.js and npm dalam dokumentasi npm.
- Benang dipasang di workstation lokal. Untuk informasi selengkapnya, lihat Instalasi di dokumentasi Yarn.

Batasan

- Solusi ini diterapkan ke satu Akun AWS.
- Kode pengembalian yang diharapkan dari wadah adalah `0` untuk sukses. Kode pengembalian lainnya menunjukkan kegagalan, dan CloudFormation tumpukan akan berputar kembali.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan arsitektur konstruksi.

Diagram menunjukkan alur kerja dari`waitcondition-hook-for-aws-fargate-task`:

1. `WaitCondition`dan `WaitConditionHandler` disediakan untuk mendengarkan respons dari fungsi. AWS Lambda
2. Tergantung pada hasil tugas, baik `CallbackFunction` atau `ErrorHandlerFunction` dipicu oleh penyelesaian tugas Fargate.
3. Fungsi Lambda mengirimkan sinyal SUCCED atau FAILURE ke. `WaitConditionHandler`
4. `WaitConditionHandler`terus menyediakan sumber daya jika hasil eksekusi tugas Fargate berhasil, atau memutar kembali tumpukan jika tugas gagal.

Diagram berikut menunjukkan contoh alur kerja untuk melakukan migrasi database.

Contoh alur kerja menggunakan `waitcondition-hook-for-aws-fargate-task` konstruksi untuk melakukan migrasi database, sebagai berikut:

1. Instans Amazon Relational Database Service (Amazon RDS) disediakan.
2. `waitcondition-hook-for-aws-fargate-task`Konstruksi menjalankan tugas migrasi database dan menjeda tumpukan sebagai instance Amazon Elastic Compute Cloud (Amazon). EC2
3. Jika tugas migrasi selesai dengan sukses, ia akan mengirimkan sinyal Success ke CloudFormation. Jika tidak, ia mengirimkan sinyal Gagal ke CloudFormation dan memutar kembali tumpukan.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda mendefinisikan infrastruktur cloud dalam kode dan menyediakannya CloudFormation.
- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.
- AWS Fargatemembantu Anda menjalankan kontainer tanpa perlu mengelola server atau EC2 instans Amazon. Ini digunakan bersama dengan Amazon ECS.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.

- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat-alat lainnya

- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.
- Yarn adalah manajer paket open source yang dapat Anda gunakan untuk mengelola dependensi dalam JavaScript proyek. Yarn dapat membantu Anda menginstal, memperbarui, mengonfigurasi, dan menghapus dependensi paket.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub waitcondition-hook-for-aws-fargate-task.

Praktik terbaik

- Saat membuat AWS CDK aplikasi, ikuti Praktik terbaik untuk mengembangkan dan menerapkan infrastruktur cloud dengan dokumentasi AWS CDK di AWS CDK v2.
- Untuk AWS Fargate tugas tersebut, ikuti Praktik terbaik untuk gambar kontainer Amazon ECS di dokumentasi Amazon ECS.

Epik

Mengatur AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS CDK.	Untuk menginstal AWS CDK pada mesin lokal Anda atau lingkungan lain, jalankan perintah berikut:	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`npm install -g aws-cdk@latest`	
Bootstrap AWS CDK.	Bootstrapping adalah proses mempersiapkan lingkungan untuk penyebaran. Untuk bootstrap AWS CDK toolkit Anda untuk target Akun AWS dan Wilayah AWS, jalankan perintah berikut: `cdk bootstrap aws://ACCOUNT-NUMBER-1/REGION-1` Perintah ini menciptakan CloudFormation tumpukan bernama`CDKToolkit` .	Arsitek awan

Jalankan WaitCondition hook untuk konstruksi AWS Fargate tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat proyek CDK.	Buat proyek CDK menggunakan bahasa yang Anda sukai. Pola ini menggunakan TypeScript. Untuk membuat proyek CDK menggunakan TypeScript, jalankan perintah berikut: `cdk init app –language typescript`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal paket .	Jalankan `npm install` di jalur root proyek CDK Anda. Setelah perpustakaan CDK diinstal, jalankan perintah berikut untuk menginsta l`waitcondition-hook-for-aws-fargate-task` : `yarn add waitcondition-hook-for-aws-fargate-task`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bangun aplikasi CDK dan komponen Amazon ECS Anda.	Bangun proyek CDK Anda. Diperlukan sumber daya definisi tugas Amazon ECS. Untuk informasi tentang membuat definisi tugas, lihat definisi tugas Amazon ECS dalam dokumentasi Amazon ECS. Contoh berikut menggunakan konstruksi ini: `import * as cdk from 'aws-cdk-lib'; import { Vpc } from 'aws-cdk-lib/aws-ec2'; import * as ecr from 'aws-cdk-lib/aws-ecr'; import * as ecs from 'aws-cdk-lib/aws-ecs'; import { Construct } from 'constructs'; import { FargateRunner } from 'waitcondition-hook-for-aws-fargate-task'; import { Queue } from 'aws-cdk-lib/aws-sqs';`  `export class FargateRunnerStack extends cdk.Stack { constructor(scope: Construct, id: string,`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	props?: cdk.StackProps) {
	super(scope, id, props);
	// Define the VPC
	const vpc = new Vpc(this, 'MyVpc')
	// Define the Fargate Task
	const taskDefinition = new ecs.FargateTaskDefinition(this, 'MyTask', {});
	// Import exiting ecr repo
	const repo = ecr.Repository.fromRepositoryName(this, 'MyRepo', 'RepoName');
	// Add a container to the task
	taskDefinition.addContainer('MyContainer', {
	image: ecs.ContainerImage.fromEcrRepository(repo),
	});
	// Create the Fargate runner
	const myFargateRunner = new FargateRunner(this, 'MyRunner', {
	fargateTaskDef: taskDefinition,
	timeout: `${60 * 5}`,
	vpc: vpc,
	});	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`// Create the SQS queue const myQueue = new Queue(this, 'MyQueue', {}); // Add dependency myQueue.node.addDependency(myFargateRunner); } }`	

Synth dan luncurkan aplikasi CDK.	1. Untuk menghasilkan aset dan CloudFormation template, jalankan perintah berikut di jalur root CDK Anda:
	`cdk synth`
	2. Setelah `synth` perintah berhasil, jalankan perintah berikut untuk menyebarkan sumber daya:
	`cdk deploy`
	
	`waitcondition-hook-for-aws-fargate-task` Konstruk menjalankan tugas Fargate.
		Arsitek awan

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Untuk membersihkan sumber daya yang disediakan dari langkah sebelumnya, jalankan perintah berikut: `cdk destroy`	Arsitek awan

Pemecahan masalah

Isu	Solusi
Kegagalan CloudFormation tumpukan umum	Untuk membantu memecahkan masalah kegagalan CloudFormation tumpukan umum, tambahkan `--no-rollback` tanda seperti yang ditunjukkan pada contoh berikut: `cdk deploy --no-rollback` Perintah ini akan menjeda CloudFormation tumpukan dari memutar kembali, yang memberi Anda sumber daya untuk memecahkan masalah. Untuk informasi selengkapnya, lihat Memilih cara menangani kegagalan saat menyediakan sumber daya dalam dokumenta si. CloudFormation
AWS Step Functions kegagalan	Mesin AWS Step Functions negara mungkin gagal mengeksekusi karena alasan yang berbeda. Dengan `–disable-rollback` dikonfigurasi, gunakan langkah-langkah berikut untuk memecahkan masalah:

Isu	Solusi
	1. Masuk ke Konsol Manajemen AWS, masukkan Step Functions di kolom Pencarian, lalu pilih layanan Step Functions. 2. Di panel navigasi kiri, pilih mesin State dan kemudian pilih mesin status yang disediakan oleh tumpukan. CloudFormation 3. Dalam Eksekusi, pilih nama eksekusi yang gagal secara tak terduga. 4. Dalam tampilan Acara, pilih langkah gagal. Untuk informasi selengkapnya, lihat Memecahkan masalah di Step Functions dan Melihat detail eksekusi di konsol Step Functions dalam dokumentasi. AWS Step Functions

Isu	Solusi
AWS Lambda kegagalan fungsi	Konstruksi ini menyediakan dua fungsi `CallbackFunction` Lambda: dan. `ErrorhandlerFunction` Mereka bisa gagal karena berbagai alasan seperti pengecualian yang tidak tertangani. Gunakan langkah-langkah berikut untuk memecahkan masalah: 1. Masuk ke Konsol Manajemen AWS, masukkan CloudWatchdi kolom Pencarian, lalu pilih CloudWatch layanan. 2. Pada panel navigasi kiri, pilih Grup log. 3. Di bidang Pencarian, masukkan nama fungsi Lambda. 4. Pilih nama grup log yang terkait dengan fungsi Lambda. 5. Untuk pergi ke hasil eksekusi fungsi Lambda, pilih aliran log terbaru. Untuk informasi selengkapnya, lihat Memecahkan masalah di Lambda dalam dokumentasi. AWS Lambda

Sumber daya terkait

AWS dokumentasi

- AWS CDK Membangun Referensi API
- Memulai dengan AWS CDK
- Pelajari cara membuat dan menggunakan sumber daya Amazon ECS
- Pelajari cara memulai dengan Step Functions
- Apa itu AWS CDK?

Sumber daya lainnya

- Waitcondition Hook untuk AWS Fargate tugas (npm)
- waitcondition-hook-for-aws-fargate-task 1.0.6 (pypi.org)

Gunakan repositori sumber Git pihak ketiga di AWS CodePipeline

Kirankumar Chandrashekar, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan AWS CodePipeline dengan repositori sumber Git pihak ketiga.

AWS CodePipeline adalah layanan pengiriman berkelanjutan yang mengotomatiskan tugas untuk membangun, menguji, dan menerapkan perangkat lunak Anda. Layanan saat ini mendukung repositori Git yang dikelola oleh CodeCommit, GitHub AWS, dan Atlassian Bitbucket. Namun, beberapa perusahaan menggunakan repositori Git pihak ketiga yang terintegrasi dengan layanan single sign-on (SSO) dan Microsoft Active Directory untuk otentikasi. Anda dapat menggunakan repositori Git pihak ketiga ini sebagai sumber untuk CodePipeline dengan membuat tindakan kustom dan webhook.

Webhook adalah pemberitahuan HTTP yang mendeteksi peristiwa di alat lain, seperti GitHub repositori, dan menghubungkan peristiwa eksternal tersebut ke pipeline. Saat Anda membuat webhook di CodePipeline, layanan mengembalikan URL yang dapat Anda gunakan di webhook repositori Git Anda. Jika Anda mendorong kode ke cabang tertentu dari repositori Git, webhook Git memulai CodePipeline webhook melalui URL ini, dan menetapkan tahap sumber pipeline ke In Progress. Ketika pipeline dalam keadaan ini, pekerja pekerjaan melakukan polling CodePipeline untuk pekerjaan khusus, menjalankan pekerjaan, dan mengirimkan status keberhasilan atau kegagalan ke CodePipeline. Dalam hal ini, karena pipeline berada dalam tahap sumber, job worker mendapatkan konten repositori Git, membuat zip konten, dan mengunggahnya ke bucket Amazon Simple Storage Service (Amazon S3) tempat artefak untuk pipeline disimpan, menggunakan kunci objek yang disediakan oleh job yang disurvei. Anda juga dapat mengaitkan transisi untuk tindakan kustom dengan peristiwa di Amazon CloudWatch, dan memulai pekerja pekerjaan berdasarkan peristiwa tersebut. Penyiapan ini memungkinkan Anda untuk menggunakan repositori Git pihak ketiga yang tidak didukung oleh layanan sebagai sumber. CodePipeline

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sebuah repositori Git yang mendukung webhooks dan dapat terhubung ke URL CodePipeline webhook melalui internet

- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi untuk bekerja dengan akun AWS

Arsitektur

Polanya melibatkan langkah-langkah ini:

1. Pengguna melakukan kode ke repositori Git.
2. Webhook Git disebut.
3. CodePipeline Webhook disebut.
4. Pipeline diatur ke In Progress, dan tahap sumber diatur ke status In Progress.
5. Tindakan tahap sumber memulai aturan CloudWatch Peristiwa, yang menunjukkan bahwa itu dimulai.
6. CloudWatch Acara ini memulai fungsi Lambda.
7. Fungsi Lambda mendapatkan detail pekerjaan tindakan kustom.
8. Fungsi Lambda memulai CodeBuild AWS dan meneruskan semua informasi terkait pekerjaan.
9. CodeBuild mendapatkan kunci SSH publik atau kredensyal pengguna untuk akses HTTPS Git dari Secrets Manager.
10. CodeBuild mengkloning repositori Git untuk cabang tertentu.
11. CodeBuild zip arsip dan mengunggahnya ke bucket S3 yang berfungsi sebagai toko artefak. CodePipeline

Alat

- AWS CodePipeline — AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dikelola sepenuhnya yang membantu Anda mengotomatiskan saluran pipa rilis untuk pembaruan aplikasi dan infrastruktur yang cepat dan andal. CodePipeline mengotomatiskan fase build, test, dan deployment dari proses rilis Anda untuk setiap perubahan kode, berdasarkan model rilis yang Anda tentukan. Ini memungkinkan Anda untuk dengan cepat dan andal memberikan fitur dan pembaruan. Anda dapat mengintegrasikan AWS CodePipeline dengan layanan pihak ketiga seperti GitHub atau dengan plugin kustom Anda sendiri.

- AWS Lambda — AWS Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. Dengan Lambda, Anda dapat menjalankan kode untuk hampir semua jenis aplikasi atau layanan backend tanpa administrasi yang diperlukan. Anda mengunggah kode Anda dan Lambda menangani semua yang diperlukan untuk menjalankan dan menskalakan kode Anda dengan ketersediaan tinggi. Anda dapat mengatur kode untuk memulai secara otomatis dari layanan AWS lain atau memanggilnya langsung dari web atau aplikasi seluler apa pun.
- AWS CodeBuild — AWS CodeBuild adalah layanan integrasi berkelanjutan yang dikelola sepenuhnya yang mengkompilasi kode sumber, menjalankan pengujian, dan menghasilkan paket perangkat lunak yang siap digunakan. Dengan CodeBuild, Anda tidak perlu menyediakan, mengelola, dan menskalakan server build Anda sendiri. CodeBuild menskalakan terus menerus dan memproses beberapa build secara bersamaan, sehingga build Anda tidak dibiarkan menunggu dalam antrian. Anda dapat memulai dengan cepat menggunakan lingkungan build yang dikemas sebelumnya, atau Anda dapat membuat lingkungan build khusus yang menggunakan alat build Anda sendiri.
- AWS Secrets Manager — AWS Secrets Manager membantu Anda melindungi rahasia yang diperlukan untuk mengakses aplikasi, layanan, dan sumber daya TI Anda. Layanan ini memungkinkan Anda untuk memutar, mengelola, dan mengambil kredensyal database, kunci API, dan rahasia lainnya sepanjang siklus hidupnya. Pengguna dan aplikasi mengambil rahasia dengan memanggil Secrets Manager APIs, tanpa harus hardcode informasi sensitif dalam teks biasa. Secrets Manager menawarkan rotasi rahasia dengan integrasi bawaan untuk Amazon Relational Database Service (Amazon RDS), Amazon Redshift, dan Amazon DocumentDB. Layanan ini dapat diperluas untuk mendukung jenis rahasia lainnya, termasuk kunci API dan OAuth token. Selain itu, Secrets Manager memungkinkan Anda mengontrol akses ke rahasia dengan menggunakan izin berbutir halus, dan mengaudit rotasi rahasia secara terpusat untuk sumber daya di AWS Cloud, layanan pihak ketiga, dan lingkungan lokal.
- Amazon CloudWatch - Amazon CloudWatch adalah layanan pemantauan dan observasi yang dibangun untuk DevOps insinyur, pengembang, insinyur keandalan situs (SREs), dan manajer TI. CloudWatch memberi Anda data dan wawasan yang dapat ditindaklanjuti untuk memantau aplikasi Anda, menanggapi perubahan kinerja di seluruh sistem, mengoptimalkan pemanfaatan sumber daya, dan mendapatkan pandangan terpadu tentang kesehatan operasional. CloudWatch mengumpulkan data pemantauan dan operasional dalam bentuk log, metrik, dan peristiwa, memberi Anda tampilan terpadu tentang sumber daya, aplikasi, dan layanan AWS yang berjalan di AWS dan server lokal. Anda dapat menggunakannya CloudWatch untuk mendeteksi perilaku anomali di lingkungan Anda, menyetel alarm, memvisualisasikan log dan metrik secara berdampingan, mengambil tindakan otomatis, memecahkan masalah, dan menemukan wawasan agar aplikasi Anda berjalan lancar.

- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang memungkinkan Anda menyimpan dan melindungi sejumlah data untuk berbagai kasus penggunaan, seperti situs web, aplikasi seluler, pencadangan dan pemulihan, arsip, aplikasi perusahaan, perangkat IoT, dan analitik data besar. Amazon S3 menyediakan fitur easy-to-use manajemen untuk membantu Anda mengatur data dan mengonfigurasi kontrol akses yang disetel dengan baik untuk memenuhi persyaratan bisnis, organisasi, dan kepatuhan spesifik Anda.

Epik

Buat tindakan kustom di CodePipeline

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tindakan kustom menggunakan AWS CLI atau AWS. CloudFormation	Langkah ini melibatkan pembuatan tindakan sumber khusus yang dapat digunakan dalam tahap sumber pipeline di akun AWS Anda di wilayah tertentu. Anda harus menggunakan AWS CLI atau AWS CloudFormation (bukan konsol) untuk membuat tindakan sumber kustom. Untuk informasi selengkapnya tentang perintah dan langkah yang dijelaskan dalam epos ini dan epos lainnya, lihat bagian “Sumber daya terkait” di akhir pola ini. Di AWS CLI, gunakan perintah. create-custom-action-type Gunakan --configuration-properties untuk menyediakan semua parameter yang diperlukan agar pekerja pekerjaan dapat	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	diproses saat melakukan polling CodePipeline untuk suatu pekerjaan. Pastikan untuk mencatat nilai yang diberikan ke opsi --provider dan --action-version, sehingga Anda dapat menggunakan nilai yang sama saat membuat pipeline dengan tahap sumber kustom ini. Anda juga dapat membuat tindakan sumber kustom di AWS CloudForm ation dengan menggunakan jenis sumber daya AWS::Code Pipeline::CustomActionType.	

Mengatur otentikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat key pair SSH.	Buat key pair Secure Shell (SSH). Untuk instruksi, lihat GitHub dokumentasi.	Sistem/insinyur DevOps
Buat rahasia di AWS Secrets Manager.	Salin konten kunci pribadi dari key pair SSH dan buat rahasia di AWS Secrets Manager. Rahasia ini digunakan untuk otentikasi saat mengakses repositori Git.	AWS Umum
Tambahkan kunci publik ke repositori Git.	Tambahkan kunci publik dari key pair SSH ke pengatura n akun repositori Git, untuk	Sistem/insinyur DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	otentikasi terhadap kunci pribadi.	

Buat pipeline dan webhook

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pipeline yang menyertakan tindakan sumber kustom.	Buat pipeline di CodePipeline. Saat Anda mengonfigurasi tahap sumber, pilih tindakan sumber kustom yang Anda buat sebelumnya. Anda dapat melakukan ini di CodePipeline konsol AWS atau di AWS CLI. CodePipeline meminta Anda untuk properti konfigurasi yang Anda tetapkan pada tindakan kustom. Informasi ini diperlukan bagi pekerja kerja untuk memproses pekerjaan untuk tindakan kustom. Ikuti wizard dan buat tahap selanjutnya untuk pipeline.	AWS Umum
Buat CodePipeline webhook.	Buat webhook untuk pipeline yang Anda buat dengan aksi sumber kustom. Anda harus menggunakan AWS CLI atau AWS CloudFormation (bukan konsol) untuk membuat webhook. Di AWS CLI, jalankan perintah put-webhook dan berikan nilai yang sesuai untuk opsi webhook. Catat URL webhook	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang dikembalikan perintah. Jika Anda menggunakan AWS CloudFormation untuk membuat webhook, gunakan jenis AWS::CodePipeline: :Webhook sumber daya. Pastikan untuk menampilkan URL webhook dari sumber daya yang dibuat, dan catat itu.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi dan proyek Lambda. CodeBuild	Pada langkah ini, Anda menggunakan Lambda dan CodeBuild untuk membuat pekerja pekerjaan yang akan melakukan polling CodePipeline untuk permintaa n pekerjaan untuk tindakan kustom, menjalankan pekerjaan, dan mengembal ikan hasil status ke. CodePipel ine Buat fungsi Lambda yang diprakarsai oleh aturan Amazon CloudWatch Events saat tahap tindakan sumber kustom dari pipeline bertransi si ke “Sedang Berlangsung.” Ketika fungsi Lambda dimulai, itu harus mendapatkan detail pekerjaan tindakan khusus dengan melakukan polling untuk pekerjaan. Anda dapat menggunakan PollForJobs API untuk mengembalikan informasi ini. Setelah informasi pekerjaan yang disurvei diperoleh, fungsi Lambda harus mengembalikan pengakuan, dan kemudian memproses informasi dengan data yang diperolehnya dari properti konfigurasi untuk tindakan kustom. Ketika pekerja siap untuk berbicara dengan repositori Git, Anda	AWS Umum, pengembang kode

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dapat memulai sebuah CodeBuild proyek, karena akan lebih mudah untuk menangani tugas Git dengan menggunakan klien SSH.	

Buat acara di CloudWatch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan CloudWatch Acara.	Buat aturan CloudWatch Peristiwa yang memulai fungsi Lambda sebagai target setiap kali tahap tindakan kustom pipeline bertransisi ke “Sedang Berlangsung.”	AWS Umum

Sumber daya terkait

Membuat tindakan kustom di CodePipeline

- Buat dan tambahkan tindakan kustom di CodePipeline
- AWS::CodePipeline::CustomActionType sumber daya

Menyiapkan otentikasi

- Membuat dan Mengelola Rahasia dengan AWS Secrets Manager

Membuat pipeline dan webhook

- Buat Pipeline di CodePipeline
- referensi perintah put-webhook
- AWS::CodePipeline::Webhook sumber daya

- PollForJobs Referensi API
- Buat dan Tambahkan Tindakan Kustom di CodePipeline
- Buat proyek build di AWS CodeBuild

Membuat acara

- Mendeteksi dan bereaksi terhadap perubahan status pipeline dengan Amazon CloudWatch Events

Referensi tambahan

- Bekerja dengan jaringan pipa di CodePipeline
- Panduan pengembang AWS Lambda

Buat CI/CD pipeline untuk memvalidasi konfigurasi Terraform dengan menggunakan AWS CodePipeline

Aromal Raj Jayarajan dan Vijesh Vijayakumaran Nair, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menguji konfigurasi HashiCorp Terraform dengan menggunakan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang digunakan oleh AWS. CodePipeline

Terraform adalah aplikasi antarmuka baris perintah yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud. Solusi yang disediakan dalam pola ini membuat CI/CD pipeline yang membantu Anda memvalidasi integritas konfigurasi Terraform Anda dengan menjalankan lima tahap: CodePipeline

1. `"checkout"`menarik konfigurasi Terraform yang Anda uji dari repositori AWS. CodeCommit
2. `"validate"`menjalankan infrastruktur sebagai alat validasi kode (IAc), termasuk tfsec,, dan checkov. TFLint Tahap ini juga menjalankan perintah validasi Terraform IAc berikut: dan. `terraform validate` `terraform fmt`
3. `"plan"`menunjukkan perubahan apa yang akan diterapkan pada infrastruktur jika konfigurasi Terraform diterapkan.
4. `"apply"`menggunakan rencana yang dihasilkan untuk menyediakan infrastruktur yang diperlukan di lingkungan pengujian.
5. `"destroy"`menghapus infrastruktur pengujian yang dibuat selama `"apply"` tahap.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Git, diinstal dan dikonfigurasi pada mesin lokal Anda
- Terraform, diinstal dan dikonfigurasi pada mesin lokal Anda

Batasan

- Pendekatan pola ini menerapkan AWS CodePipeline ke dalam satu akun AWS dan Wilayah AWS saja. Perubahan konfigurasi diperlukan untuk penerapan multi-akun dan Multi-wilayah.
- Peran AWS Identity and Access Management (IAM) yang diberikan pola ini (codepipeline_iam_role) mengikuti prinsip hak istimewa terkecil. Izin peran IAM ini harus diperbarui berdasarkan sumber daya spesifik yang perlu dibuat pipeline Anda.

Versi produk

- AWS CLI versi 2.9.15 atau yang lebih baru
- Terraform versi 1.3.7 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- AWS CodePipeline
- AWS CodeBuild
- AWS CodeCommit
- AWS IAM
- Amazon Simple Storage Service (Amazon S3)
- AWS Key Management Service (AWS KMS)
- Terraform

Arsitektur target

Diagram berikut menunjukkan contoh alur kerja CI/CD pipeline untuk menguji konfigurasi Terraform di. CodePipeline

Diagram menunjukkan alur kerja berikut:

1. Di CodePipeline, pengguna AWS memulai tindakan yang diusulkan dalam paket Terraform dengan menjalankan `terraform apply` perintah di AWS CLI.
2. AWS CodePipeline mengasumsikan peran layanan IAM yang mencakup kebijakan yang diperlukan untuk mengakses CodeCommit, AWS KMS CodeBuild, dan Amazon S3.

3. CodePipeline menjalankan tahap `"checkout"` pipeline untuk menarik konfigurasi Terraform dari CodeCommit repositori AWS untuk pengujian.
4. CodePipeline menjalankan `"validate"` tahapan untuk menguji konfigurasi Terraform dengan menjalankan alat validasi IAc dan menjalankan perintah validasi Terraform IAc dalam sebuah proyek. CodeBuild
5. CodePipeline menjalankan `"plan"` panggung untuk membuat rencana dalam CodeBuild proyek berdasarkan konfigurasi Terraform. Pengguna AWS dapat meninjau paket ini sebelum perubahan diterapkan ke lingkungan pengujian.
6. Code Pipeline menjalankan `"apply"` tahapan untuk mengimplementasikan rencana dengan menggunakan CodeBuild proyek untuk menyediakan infrastruktur yang diperlukan di lingkungan pengujian.
7. CodePipeline menjalankan `"destroy"` panggung, yang digunakan CodeBuild untuk menghapus infrastruktur pengujian yang dibuat selama `"apply"` tahap.
8. Bucket Amazon S3 menyimpan artefak pipeline, yang dienkripsi dan didekripsi dengan menggunakan kunci yang dikelola pelanggan AWS KMS.

Alat

Alat

Layanan AWS

- AWS CodePipeline membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS CodeBuild adalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Layanan lainnya

- HashiCorp Terraform adalah aplikasi antarmuka baris perintah yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.

Kode

Kode untuk pola ini tersedia di GitHub aws-codepipeline-terraform-cicdsamplesrepositori. Repositori berisi konfigurasi Terraform yang diperlukan untuk membuat arsitektur target yang diuraikan dalam pola ini.

Epik

Menyediakan komponen solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Kloning GitHub aws-codepipeline-terraform-cicdsamplesrepositori dengan menjalankan perintah berikut di jendela terminal: `git clone https://github.com/aws-samples/aws-codepipeline-terraform-cicd-samples.git` Untuk informasi selengkapnya, lihat Mengkloning repositori dalam dokumentasi. GitHub	DevOps insinyur
Buat file definisi variabel Terraform.	Buat `terraform.tfvars` file berdasarkan persyaratan kasus penggunaan Anda.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda dapat memperbarui variabel dalam `examples/terraform.tfvars` file yang ada di repositori kloning. Untuk informasi selengkap nya, lihat Menetapkan nilai ke variabel modul root dalam dokumentasi Terraform. **Note** `Readme.md` File repositori mencakup informasi lebih lanjut tentang variabel yang diperlukan.	
Konfigurasikan AWS sebagai penyedia Terraform.	1. Dalam editor kode, buka file repositori kloning. `main.tf` 2. Tambahkan konfigurasi yang diperlukan untuk membangun konektivitas ke akun AWS target. Untuk informasi selengkap nya, lihat penyedia AWS di dokumentasi Terraform.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui konfigurasi penyedia Terraform untuk membuat bucket replikasi Amazon S3.	1. Buka `S3` direktori repositori dengan menjalankan perintah berikut: `cd ./modules/s3` 2. Perbarui konfigurasi penyedia Terraform untuk membuat bucket replikasi Amazon S3 dengan memperbarui nilai `region` dalam file. `tf` Pastikan Anda memasukkan Wilayah yang Anda inginkan Amazon S3 untuk mereplikasi objek. 3. (Opsional) Secara default, Terraform menggunakan file status lokal untuk manajemen negara. Jika Anda ingin menambahkan Amazon S3 sebagai backend jarak jauh, Anda harus memperbarui konfigurasi Terraform. Untuk informasi selengkapnya, lihat Konfigurasi backend di dokumentasi Terraform. **Note** Replikasi mengaktifkan penyalinan objek	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	otomatis dan asinkron di seluruh bucket Amazon S3.	
Inisialisasi konfigurasi Terraform.	Untuk menginisialisasi direktori kerja Anda yang berisi file konfigurasi Terraform, jalankan perintah berikut di folder root repositori kloning: `terraform init`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat paket Terraform.	Untuk membuat paket Terraform, jalankan perintah berikut di folder root repositori kloning: `terraform plan --var-file=terraform.tfvars -out=tfplan` **Note** Terraform mengevaluasi file konfigurasi untuk menentukan status target untuk sumber daya yang dideklarasikan. Kemudian membandingkan status target dengan keadaan saat ini dan membuat rencana.	DevOps insinyur
Verifikasi paket Terraform.	Tinjau paket Terraform dan konfirmasikan bahwa paket tersebut mengonfigurasi arsitektur yang diperlukan di akun AWS target Anda.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan solusinya.	1. Untuk menerapkan paket Terraform, jalankan perintah berikut di folder root repositori kloning: `terraform apply "tfplan"` 2. Masukkan ya untuk mengonfirmasi bahwa Anda ingin menyebarkan sumber daya. **Note** Terraform membuat, memperbarui, atau menghancurkan infrastruktur untuk mencapai status target yang dinyatakan dalam file konfigurasi.	DevOps insinyur

Validasi konfigurasi Terraform dengan menjalankan pipeline

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan repositori kode sumber.	1. Dari keluaran Terraform, dapatkan detail repositori sumber untuk repositori yang berisi konfigurasi Terraform yang ingin Anda validasi.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Masuk ke AWS Management Console. Kemudian, buka CodeCommit konsol. 3. Buat cabang baru di repositori sumber bernama. `main` Untuk petunjuk, lihat Membuat cabang di AWS CodeCommit dalam CodeCommit dokumentasi. 4. `main`Kloning cabang repositori sumber ke workstation lokal Anda. Untuk petunjuknya, lihat Langkah penyiapan untuk koneksi HTTPS ke CodeCommit repositori AWS di Windows dengan pembantu kredensyal AWS CLI dalam dokumentasi. CodeCommit 5. Salin **`templates`** folder dari GitHub aws-codepipeline-terraform-cicdsamplesrepositori dengan menjalankan perintah berikut: `cp -r templates $YOUR_CODECOMMIT_REPO_ROOT`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note `templates` Folder berisi file spesifikasi build dan skrip validasi untuk direktori root repositori sumber. 6. Tambahkan konfigurasi Terraform IAc yang diperlukan ke folder root repositori sumber. 7. Tambahkan detail untuk backend jarak jauh dalam konfigurasi Terraform proyek Anda. Untuk informasi lebih lanjut, lihat S3 di dokumentasi Terraform. 8. (Opsional) Perbarui variabel dalam `templates` folder untuk mengaktifkan atau menonaktifkan pemindaian yang telah dikonfigurasi sebelumnya, versi perubahan alat, dan untuk menentukan direktori Anda dalam file skrip khusus. Untuk informasi selengkapnya, lihat bagian Informasi tambahan dari pola ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	9. Dorong perubahan ke `main` cabang repositori sumber.	
Validasi tahapan pipa.	1. Masuk ke AWS Management Console dan buka CodePipeline konsol. 2. Pada output yang dihasilkan dari `terraform apply "tfplan"` perintah di bagian Epic sebelumnya, temukan nama yang dihasilkan CodePipeline. 3. Buka pipeline di CodePipeline konsol dan pilih Rilis perubahan. 4. Tinjau setiap tahap pipa dan konfirmasikan itu berfungsi seperti yang diharapkan. Untuk informasi selengkapnya, lihat Melihat detail pipeline dan histori (konsol) di Panduan CodePipeline Pengguna AWS. **Important** Ketika perubahan dilakukan ke cabang utama repositori sumber, pipa uji diaktifkan secara otomatis.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi keluaran laporan.	1. Di CodePipeline konsol, di panel navigasi kiri, pilih Build. Kemudian, pilih Laporkan riwayat. 2. Tinjau laporan pemindaian tfsec dan checkov yang dihasilkan oleh pipeline. Laporan ini dapat membantu Anda mengidentifikasi masalah melalui visualisasi dan representasi grafis. **Note** `<project_name>-validate` CodeBuild Proyek ini menghasilkan laporan kerentanan untuk kode Anda selama `"validate"` tahap.	DevOps insinyur

Bersihkan sumber daya Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan pipa dan sumber daya terkait.	Untuk menghapus sumber daya pengujian dari akun AWS Anda, jalankan perintah berikut di folder root repositori kloning:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`terraform destroy --var-file=terraform.tfvars`	

Pemecahan masalah

Isu	Solusi
Anda menerima AccessDenied kesalahan selama `"apply"` tahap.	1. Tinjau log eksekusi CodeBuild proyek yang terkait dengan `"apply"` tahapan untuk mengidentifikasi izin IAM yang hilang. Untuk informasi selengkapnya, lihat Melihat detail build di AWS CodeBuild di Panduan CodeBuild Pengguna AWS. 2. Dalam editor kode, buka folder repositori kloning. `modules` Kemudian, navigasikan ke `iam-role` folder dan buka `main.tf` file yang ada di folder itu. 3. Dalam `codepipeline_policy` pernyataan tersebut, tambahkan kebijakan IAM yang diperlukan untuk menyediakan sumber daya di akun AWS Anda.

Sumber daya terkait

- Blok modul (dokumentasi Terraform)
- Cara menggunakan CI/CD untuk menerapkan dan mengonfigurasi layanan keamanan AWS dengan Terraform (posting blog AWS)
- Menggunakan peran terkait layanan (dokumentasi IAM)
- buat-pipa (dokumentasi AWS CLI)
- Konfigurasikan enkripsi sisi server untuk artefak yang disimpan di Amazon S3 untuk (dokumentasi AWS) CodePipeline CodePipeline
- Kuota untuk AWS CodeBuild (CodeBuild dokumentasi AWS)

- Perlindungan data di AWS CodePipeline (CodePipeline dokumentasi AWS)

Informasi tambahan

Modul Terraform kustom

Berikut ini adalah daftar modul Terraform kustom yang digunakan dalam pola ini:

- `codebuild_terraform`menciptakan CodeBuild proyek yang membentuk setiap tahap pipa.
- `codecommit_infrastructure_source_repo`menangkap dan membuat CodeCommit repositori sumber.
- `codepipeline_iam_role`membuat peran IAM yang diperlukan untuk pipa.
- `codepipeline_kms`membuat kunci AWS KMS yang diperlukan untuk enkripsi dan dekripsi objek Amazon S3.
- `codepipeline_terraform`membuat pipa uji untuk CodeCommit repositori sumber.
- `s3_artifacts_bucket`membuat bucket Amazon S3 untuk mengelola artefak pipeline.

Membangun file spesifikasi

Berikut ini adalah daftar file spesifikasi build (buildspec) yang digunakan pola ini untuk menjalankan setiap tahap pipeline:

- `buildspec_validate.yml`menjalankan `"validate"` panggung.
- `buildspec_plan.yml`menjalankan `"plan"` panggung.
- `buildspec_apply.yml`menjalankan `"apply"` panggung.
- `buildspec_destroy.yml`menjalankan `"destroy"` panggung.

Membangun variabel file spesifikasi

Setiap file buildspec menggunakan variabel berikut untuk mengaktifkan pengaturan khusus build yang berbeda:

Variabel	Nilai default	Deskripsi
`CODE_SRC_DIR`	`"."`	Mendefinisikan direktori sumber CodeCommit

TF_VERSION	“1.3.7"	Mendefinisikan versi Terraform untuk lingkungan build

buildspec_validate.ymlFile ini juga mendukung variabel berikut untuk mengaktifkan pengaturan khusus build yang berbeda:

Variabel	Nilai default	Deskripsi
SCRIPT_DIR	“. /template/skrip”	Mendefinisikan direktori skrip
ENVIRONMENT	“pengembang”	Mendefinisikan nama lingkungan
SKIPVALIDATIONFAILURE	“Y”	Melewatkan validasi pada kegagalan
ENABLE_TFVALIDATE	“Y”	Mengaktifkan validasi Terraform
ENABLE_TFFORMAT	“Y”	Mengaktifkan format Terraform
ENABLE_TFCHECKOV	“Y”	Mengaktifkan pemindaian checkov
ENABLE_TFSEC	“Y”	Mengaktifkan pemindaian tfsec
TFSEC_VERSION	“v1.28.1"	Mendefinisikan versi tfsec

Lebih banyak pola

- Akses aplikasi kontainer secara pribadi di Amazon EKS menggunakan AWS PrivateLink dan Network Load Balancer
- Kaitkan AWS CodeCommit repositori di satu dengan Akun AWS Amazon SageMaker AI Studio Classic di akun lain
- Mengotomatiskan pembuatan akun dengan menggunakan Akselerator Zona Landing di AWS
- Otomatis menambahkan atau memperbarui entri registri Windows menggunakan AWS Systems Manager
- Mengotomatiskan pencadangan untuk Amazon RDS untuk instans DB Amazon RDS for PostgreSQL dengan menggunakan AWS Batch
- Otomatiskan penerapan aplikasi bersarang menggunakan AWS SAM
- Otomatiskan penerapan Node Termination Handler di Amazon EKS dengan menggunakan pipeline CI/CD
- Otomatiskan konfigurasi RabbitMQ di Amazon MQ
- Otomatiskan replikasi instans Amazon RDS di seluruh Akun AWS
- Secara otomatis membangun dan menyebarkan aplikasi Java ke Amazon EKS menggunakan pipeline CI/CD
- Secara otomatis menghasilkan model PynamoDB dan fungsi CRUD untuk Amazon DynamoDB dengan menggunakan aplikasi Python
- Secara otomatis memvalidasi dan menerapkan kebijakan dan peran IAM dengan menggunakan CodePipeline, IAM Access Analyzer, dan makro AWS CloudFormation
- Cadangkan server Sun SPARC di emulator Stromasys Charon-SSP di AWS Cloud
- Membangun pipeline data untuk menyerap, mengubah, dan menganalisis data Google Analytics menggunakan AWS DataOps Development Kit
- Bangun PAC Server Perusahaan Fokus Mikro dengan Amazon EC2 Auto Scaling dan Systems Manager
- Buat pipeline untuk gambar kontainer yang diperkeras menggunakan EC2 Image Builder dan Terraform
- Membangun MLOps alur kerja dengan menggunakan Amazon SageMaker AI dan Azure DevOps
- Memusatkan resolusi DNS dengan menggunakan Microsoft Active AWS Managed Microsoft AD Directory dan lokal

- Bersihkan AWS Account Factory untuk sumber daya Terraform (AFT) dengan aman setelah kehilangan file status
- Konfigurasikan logging untuk aplikasi.NET di Amazon CloudWatch Logs dengan menggunakan NLog
- Salin gambar wadah Amazon ECR di seluruh Akun AWS dan Wilayah AWS
- Buat image container Docker kustom untuk SageMaker dan gunakan untuk pelatihan model di AWS Step Functions
- Buat pipeline di Wilayah AWS yang tidak mendukung AWS CodePipeline
- Buat alarm untuk metrik kustom menggunakan deteksi anomali Amazon CloudWatch
- Sesuaikan nama peran default dengan menggunakan AWS CDK aspek dan lubang keluar
- Menerapkan pipeline yang secara bersamaan mendeteksi masalah keamanan di beberapa kiriman kode
- Menerapkan dan mengelola data lake tanpa server di AWS Cloud dengan menggunakan infrastruktur sebagai kode
- Menerapkan aplikasi kontainer saat AWS IoT Greengrass V2 berjalan sebagai wadah Docker
- Menerapkan resource dan paket Kubernetes menggunakan Amazon EKS dan repositori bagan Helm di Amazon S3
- Menerapkan aplikasi multi-stack menggunakan AWS CDK dengan TypeScript
- Menyebarkan validasi keamanan pengkodean real-time dengan menggunakan server MCP dengan Kiro dan asisten pengkodean lainnya
- Menerapkan instance cluster failover SQL Server di Amazon EC2 dan Amazon FSx dengan menggunakan Terraform
- Terapkan Otomatisasi Keamanan untuk AWS WAF solusi dengan menggunakan Terraform
- Kembangkan asisten berbasis obrolan AI generatif tingkat lanjut dengan menggunakan RAG dan prompt ReAct
- Aktifkan Amazon GuardDuty secara kondisional dengan menggunakan templat AWS CloudFormation
- Siapkan penskalaan otomatis berbasis peristiwa di Amazon EKS dengan menggunakan Amazon EKS Pod Identity dan KEDA
- Buat rekomendasi yang dipersonalisasi dan diberi peringkat ulang menggunakan Amazon Personalize
- Dapatkan notifikasi Amazon SNS saat status kunci kunci AWS KMS berubah

- Mengatur set izin untuk beberapa akun dengan menggunakan Account Factory untuk Terraform
- Identifikasi gambar kontainer duplikat secara otomatis saat bermigrasi ke repositori Amazon ECR
- Menerapkan versi API berbasis jalur dengan menggunakan domain khusus di Amazon API Gateway
- Tingkatkan kinerja operasional dengan mengaktifkan Amazon DevOps Guru di beberapa Wilayah AWS, akun, dan OUs dengan AWS CDK
- Instal Agen SSM di node pekerja Amazon EKS dengan menggunakan Kubernetes DaemonSet
- Integrasikan Stonebranch Universal Controller dengan AWS Mainframe Modernisasi
- Modernisasi mainframe: aktif AWS dengan Rocket DevOps Software Enterprise Suite
- Kelola set AWS IAM Identity Center izin sebagai kode dengan menggunakan AWS CodePipeline
- Kelola set AWS izin secara dinamis dengan menggunakan Terraform
- Mengelola aplikasi kontainer lokal dengan menyiapkan Amazon ECS Anywhere dengan AWS CDK
- Mengelola AWS Organizations kebijakan sebagai kode dengan menggunakan AWS CodePipeline dan Amazon Bedrock
- Migrasikan data DNS secara massal ke zona host pribadi Amazon Route 53
- Migrasi aplikasi yang dihosting IIS ke Amazon dengan menggunakan appcmd.exe EC2
- Pantau penggunaan Gambar Mesin Amazon bersama di beberapa Akun AWS
- Mengatur pipeline ETL dengan validasi, transformasi, dan partisi menggunakan AWS Step Functions
- Mengotomatiskan blue/green penyebaran database global Amazon Aurora dengan menggunakan prinsip IAc
- Pertahankan ruang IP yang dapat dirutekan dalam desain VPC multi-akun untuk subnet non-beban kerja
- Menyediakan produk Terraform AWS Service Catalog dengan menggunakan repositori kode
- Jalankan tugas AWS Systems Manager Automation secara sinkron dari AWS Step Functions
- Siapkan CI/CD pipeline untuk beban kerja hybrid di Amazon ECS Anywhere dengan menggunakan AWS CDK dan GitLab
- Siapkan CI/CD pipeline untuk migrasi database dengan menggunakan Terraform
- Menyiapkan infrastruktur Multi-AZ untuk SQL Server Always On FCI dengan menggunakan Amazon FSx
- Siapkan bot UiPath RPA secara otomatis di Amazon EC2 dengan menggunakan AWS CloudFormation

- Sederhanakan otentikasi aplikasi dengan TLS timbal balik di Amazon ECS dengan menggunakan Application Load Balancer
- Orientasi penyewa dalam arsitektur SaaS untuk model silo menggunakan C# dan AWS CDK
- Gunakan Terraform untuk mengaktifkan Amazon secara otomatis GuardDuty untuk organisasi
- Gunakan agen Amazon Bedrock untuk mengotomatiskan pembuatan kontrol entri akses di Amazon EKS melalui petunjuk berbasis teks
- Validasi Account Factory untuk kode Terraform (AFT) secara lokal
- Visualisasikan hasil AI/ML model menggunakan Flask dan AWS Elastic Beanstalk

Infrastruktur

Topik

- Akses host bastion menggunakan Session Manager dan Amazon EC2 Instance Connect
- Memusatkan resolusi DNS dengan menggunakan Microsoft Active AWS Managed Microsoft AD Directory dan lokal
- Memusatkan pemantauan dengan menggunakan Amazon CloudWatch Observability Access Manager
- Periksa EC2 instance untuk tag wajib saat peluncuran
- Bersihkan AWS Account Factory untuk sumber daya Terraform (AFT) dengan aman setelah kehilangan file status
- Buat pipeline di Wilayah AWS yang tidak mendukung AWS CodePipeline
- Sesuaikan nama peran default dengan menggunakan AWS CDK aspek dan lubang keluar
- Terapkan cluster Cassandra di Amazon EC2 dengan statis IPs pribadi untuk menghindari penyeimbangan kembali
- Memperluas VRFs ke AWS dengan menggunakan AWS Transit Gateway Connect
- Dapatkan notifikasi Amazon SNS saat status kunci kunci AWS KMS berubah
- Pertahankan ruang IP yang dapat dirutekan dalam desain VPC multi-akun untuk subnet non-beban kerja
- Menyediakan produk Terraform AWS Service Catalog dengan menggunakan repositori kode
- Daftarkan beberapa Akun AWS dengan satu alamat email dengan menggunakan Amazon SES
- Menyiapkan resolusi DNS untuk jaringan hybrid di lingkungan AWS akun tunggal
- Siapkan bot UiPath RPA secara otomatis di Amazon EC2 dengan menggunakan AWS CloudFormation
- Siapkan PeopleSoft arsitektur yang sangat tersedia di AWS
- Siapkan pemulihan bencana untuk Oracle JD Edwards dengan EnterpriseOne AWS Elastic Disaster Recovery
- Mengatur deteksi CloudFormation drift di organisasi multi-wilayah dan multi-akun
- Berhasil mengimpor bucket S3 sebagai tumpukan AWS CloudFormation
- Sinkronisasi data antara sistem file Amazon EFS di Wilayah AWS yang berbeda dengan menggunakan AWS DataSync
- Uji AWS infrastruktur dengan menggunakan LocalStack dan Tes Terraform

- Tingkatkan cluster SAP Pacemaker dari ke ENSA1 ENSA2
- Gunakan Availability Zone yang konsisten VPCs di berbagai akun AWS
- Gunakan pengguna IDs dalam kebijakan IAM untuk kontrol akses dan otomatisasi
- Validasi Account Factory untuk kode Terraform (AFT) secara lokal
- Lebih banyak pola

Akses host bastion menggunakan Session Manager dan Amazon EC2 Instance Connect

Piotr Chotkowski dan Witold Kowalik, Amazon Web Services

Ringkasan

Sebuah bastion host, kadang-kadang disebut jump box, adalah server yang menyediakan satu titik akses dari jaringan eksternal ke sumber daya yang terletak di jaringan pribadi. Server yang terpapar ke jaringan publik eksternal, seperti internet, menimbulkan risiko keamanan potensial untuk akses yang tidak sah. Sangat penting untuk mengamankan dan mengontrol akses ke server ini.

Pola ini menjelaskan cara Anda dapat menggunakan Session Manager dan Amazon EC2 Instance Connect untuk terhubung dengan aman ke host bastion Amazon Elastic Compute Cloud EC2 (Amazon) yang digunakan di host Anda. Akun AWS Session Manager adalah kemampuan AWS Systems Manager. Manfaat dari pola ini meliputi:

- Host benteng yang dikerahkan tidak memiliki port terbuka dan masuk yang terpapar ke internet publik. Ini mengurangi permukaan serangan potensial.
- Anda tidak perlu menyimpan dan memelihara kunci Secure Shell (SSH) jangka panjang di dalam. Akun AWS Sebagai gantinya, setiap pengguna menghasilkan key pair SSH baru setiap kali mereka terhubung ke host bastion. AWS Identity and Access Management (IAM) kebijakan yang dilampirkan pada AWS kredensyal pengguna mengontrol akses ke host bastion.

Audiens yang dituju

Pola ini ditujukan untuk pembaca yang memiliki pengalaman dengan pemahaman dasar tentang Amazon EC2, Amazon Virtual Private Cloud (Amazon VPC), dan Hashicorp Terraform.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi
- Plugin Session Manager untuk AWS CLI, diinstal
- Terraform CLI, diinstal

- Penyimpanan untuk status Terraform, seperti bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dan tabel Amazon DynamoDB yang berfungsi sebagai backend jarak jauh untuk menyimpan status Terraform. Untuk informasi selengkapnya tentang penggunaan backend jarak jauh untuk status Terraform, lihat Backend Amazon S3 (dokumentasi Terraform). Untuk contoh kode yang menyiapkan manajemen status jarak jauh dengan backend Amazon S3, lihat remote-state-s3-backend (Terraform Registry). Perhatikan persyaratan berikut:
  - Bucket Amazon S3 dan tabel DynamoDB harus sama. Wilayah AWS
  - Saat membuat tabel DynamoDB, kunci partisi `LockID` harus (peka huruf besar/kecil), dan tipe kunci partisi harus. `String` Semua pengaturan tabel lainnya harus pada nilai default mereka. Untuk informasi selengkapnya, lihat Tentang kunci utama dan Membuat tabel di dokumentasi DynamoDB.
- Klien SSH, diinstal

Batasan

- Pola ini dimaksudkan sebagai bukti konsep (PoC) atau sebagai dasar untuk pengembangan lebih lanjut. Seharusnya tidak digunakan dalam bentuknya saat ini di lingkungan produksi. Sebelum penerapan, sesuaikan kode sampel di repositori untuk memenuhi persyaratan dan kasus penggunaan Anda.
- Pola ini mengasumsikan bahwa target bastion host menggunakan Amazon Linux 2 sebagai sistem operasinya. Meskipun dimungkinkan untuk menggunakan Amazon Machine Images (AMIs) lainnya, sistem operasi lain berada di luar cakupan untuk pola ini.

> **Note**
>
> Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs.

- Dalam pola ini, host benteng terletak di subnet pribadi tanpa gateway NAT dan gateway internet. Desain ini mengisolasi EC2 instance Amazon dari internet publik. Anda dapat menambahkan konfigurasi jaringan tertentu yang memungkinkannya berkomunikasi dengan internet. Untuk informasi selengkapnya, lihat Connect virtual private cloud (VPC) Anda ke jaringan lain dalam dokumentasi Amazon VPC. Demikian pula, mengikuti prinsip hak istimewa terkecil, host bastion tidak memiliki akses ke sumber daya lain di Anda Akun AWS kecuali Anda secara eksplisit memberikan izin. Untuk informasi selengkapnya, lihat Kebijakan berbasis sumber daya dalam dokumentasi IAM.

Versi produk

- AWS CLI versi 2
- Terraform versi 1.3.9

Arsitektur

Tumpukan teknologi target

- VPC dengan subnet pribadi tunggal
- Antarmuka berikut titik akhir VPC:
  - `amazonaws.<region>.ssm`— Titik akhir untuk AWS Systems Manager layanan.
  - `amazonaws.<region>.ec2messages`— Systems Manager menggunakan endpoint ini untuk melakukan panggilan dari Agen SSM ke layanan Systems Manager.
  - `amazonaws.<region>.ssmmessages`— Session Manager menggunakan endpoint ini untuk terhubung ke EC2 instans Amazon Anda melalui saluran data yang aman.
- EC2 Instans `t3.nano` Amazon yang menjalankan Amazon Linux 2
- Peran IAM dan profil contoh
- Grup keamanan Amazon VPC dan aturan grup keamanan untuk titik akhir dan instans Amazon EC2

Arsitektur target

Diagram menunjukkan proses berikut:

1. Pengguna mengasumsikan peran IAM yang memiliki izin untuk melakukan hal berikut:
   - Mengautentikasi, mengotorisasi, dan menghubungkan ke instans Amazon EC2
   - Memulai sesi dengan Manajer Sesi
2. Pengguna memulai sesi SSH melalui Session Manager.
3. Session Manager mengautentikasi pengguna, memverifikasi izin dalam kebijakan IAM terkait, memeriksa pengaturan konfigurasi, dan mengirim pesan ke Agen SSM untuk membuka koneksi dua arah.
4. Pengguna mendorong kunci publik SSH ke host benteng melalui metadata Amazon. EC2 Ini harus dilakukan sebelum setiap koneksi. Kunci publik SSH tetap tersedia selama 60 detik.

5. Host benteng berkomunikasi dengan titik akhir VPC antarmuka untuk Systems Manager dan Amazon. EC2
6. Pengguna mengakses host bastion melalui Session Manager dengan menggunakan saluran komunikasi dua arah terenkripsi TLS 1.2.

Otomatisasi dan skala

Opsi berikut tersedia untuk mengotomatiskan penerapan atau untuk menskalakan arsitektur ini:

- Anda dapat menerapkan arsitektur melalui pipeline continuous integration dan continuous delivery (CI/CD).
- Anda dapat memodifikasi kode untuk mengubah jenis instance dari host bastion.
- Anda dapat memodifikasi kode untuk menyebarkan beberapa host bastion. Dalam `bastion-host/main.tf` file, di blok `aws_instance` sumber daya, tambahkan `count` meta-argumen. Untuk informasi selengkapnya, lihat dokumentasi Terraform.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar. Pola ini menggunakan Session Manager, kemampuan Systems Manager.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang

akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud. Pola ini menggunakan Terraform CLI.

Repositori kode

Kode untuk pola ini tersedia di host GitHub Access a bastion dengan menggunakan Session Manager dan Amazon EC2 Instance Connect repositori.

Praktik terbaik

- Sebaiknya gunakan alat pemindaian kode otomatis untuk meningkatkan keamanan dan kualitas kode. Pola ini dipindai dengan menggunakan Checkov, alat analisis kode statis untuk IAc. Minimal, kami menyarankan Anda melakukan pemeriksaan validasi dan pemformatan dasar dengan menggunakan perintah `terraform validate` dan `terraform fmt -check -recursive` Terraform.
- Ini adalah praktik yang baik untuk menambahkan tes otomatis untuk IAc. Untuk informasi lebih lanjut tentang berbagai pendekatan untuk menguji kode Terraform, lihat Menguji HashiCorp Terraform (posting blog Terraform).
- Selama penerapan, Terraform menggunakan instans pengganti EC2 Amazon setiap kali versi baru AMI Amazon Linux 2 terdeteksi. Ini menyebarkan versi baru dari sistem operasi, termasuk tambalan dan peningkatan. Jika jadwal penerapan jarang terjadi, ini dapat menimbulkan risiko keamanan karena instance tidak memiliki tambalan terbaru. Penting untuk sering memperbarui dan menerapkan tambalan keamanan ke instans Amazon EC2 yang diterapkan. Untuk informasi selengkapnya, lihat Memperbarui manajemen di Amazon EC2.
- Karena pola ini adalah bukti konsep, ia menggunakan kebijakan AWS terkelola, sepertiAmazonSSMManagedInstanceCore. AWS kebijakan terkelola mencakup kasus penggunaan umum tetapi tidak memberikan izin hak istimewa paling sedikit. Jika diperlukan untuk kasus penggunaan Anda, sebaiknya Anda membuat kebijakan khusus yang memberikan izin hak istimewa paling sedikit untuk sumber daya yang digunakan dalam arsitektur ini. Untuk informasi selengkapnya, lihat Memulai kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit.

- Gunakan kata sandi untuk melindungi akses ke kunci SSH dan menyimpan kunci di lokasi yang aman.
- Siapkan logging dan monitoring untuk host bastion. Pencatatan dan pemantauan adalah bagian penting dari pemeliharaan sistem, baik dari perspektif operasional maupun keamanan. Ada beberapa cara untuk memantau koneksi dan aktivitas di host bastion Anda. Untuk informasi selengkapnya, lihat topik berikut dalam dokumentasi Systems Manager:
  - Pemantauan AWS Systems Manager
  - Penebangan dan pemantauan di AWS Systems Manager
  - Aktivitas sesi audit
  - Aktivitas sesi logging

Epik

Menyebarkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode.	1. Di antarmuka baris perintah, ubah direktori kerja Anda ke lokasi tempat Anda ingin menyimpan file sampel. 2. Masukkan perintah berikut. `git clone https://github.com/aws-samples/secured-bastion-host-terraform.git`	DevOps insinyur, Pengembang
Inisialisasi direktori kerja Terraform.	Langkah ini diperlukan hanya untuk penerapan pertama. Jika Anda menerapkan kembali pola, lompat ke langkah berikutnya.	DevOps insinyur, Pengembang, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Di direktori root repositori kloning, masukkan perintah berikut, di mana: • `$S3_STATE_BUCKET` adalah nama bucket Amazon S3 yang berisi status Terraform • `$PATH_TO_STATE_FILE` adalah kunci ke file status Terraform, seperti `infra/bastion-host/tetfstate` • `$AWS_REGION` adalah Wilayah tempat bucket Amazon S3 digunakan `terraform init \ -backend-config="bucket=$S3_STATE_BUCKET" \ -backend-config="key=$PATH_TO_STATE_FILE" \ -backend-config="region=$AWS_REGION` **Note** Atau, Anda dapat membuka file config.tf dan, di `terraform` bagian tersebut,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memberikan nilai-nilai ini secara manual.	
Menyebarkan sumber daya.	1. Di direktori root repositori kloning, masukkan perintah berikut. `terraform apply -var-file="dev.tfvars"` 2. Tinjau daftar semua perubahan yang akan diterapkan ke Anda Akun AWS, dan kemudian konfirmasikan penerapan. 3. Tunggu hingga semua sumber daya dikerahkan.	DevOps insinyur, Pengembang, Terraform

Mengatur lingkungan lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan koneksi SSH.	Perbarui file konfigurasi SSH untuk memungkinkan koneksi SSH melalui Session Manager. Untuk petunjuk, lihat Mengizinkan koneksi SSH untuk Pengelola Sesi. Hal ini memungkinkan pengguna yang berwenang untuk memasukkan perintah proxy yang memulai sesi Session Manager dan mentransfer	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	semua data melalui koneksi dua arah.	
Hasilkan kunci SSH.	Masukkan perintah berikut untuk menghasilkan private dan public SSH key pair lokal. Anda menggunakan key pair ini untuk terhubung ke host bastion. `ssh-keygen -t rsa -f my_key`	DevOps insinyur, Pengembang

Connect ke host bastion dengan menggunakan Session Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan ID instance.	1. Untuk terhubung ke host bastion yang digunakan , Anda memerlukan ID instans Amazon EC2 . Lakukan salah satu hal berikut untuk menemukan ID: • Buka EC2 konsol Amazon. Di panel navigasi, pilih Instans. Temukan instance host bastion. • Di AWS CLI, masukkan perintah berikut. `aws ec2 describe-instances`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk memfilter hasil, masukkan perintah berikut, di mana tag `$BASTION_HOST_TAG` yang Anda tetapkan ke host bastion. Nilai default untuk tag ini adalah `sandbox-dev-bastion-host` . `aws ec2 describe-instances \ --filters "Name=tag:Name,Values=$BASTION_HOST_TAG" \ --output text \ --query 'Reservations[*].Instances[*].InstanceId' \ --output text` 2. Salin ID EC2 instance Amazon. Anda menggunakan ID ini nanti.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim kunci publik SSH.	**Note** Di bagian ini, Anda mengunggah kunci publik ke metadata instance dari host bastion. Setelah kunci diunggah, Anda memiliki 60 detik untuk memulai koneksi dengan host bastion. Setelah 60 detik, kunci publik dihapus. Untuk informasi selengkap nya, lihat bagian Pemecahan Masalah dari pola ini. Selesaika n langkah selanjutn ya dengan cepat untuk mencegah kunci dihapus sebelum Anda terhubung ke host bastion. 1. Kirim kunci SSH ke host bastion menggunakan Amazon Instance EC2 Connect. Masukkan perintah berikut, di mana: • `$INSTANCE_ID` adalah ID dari EC2 instance Amazon	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `$PUBLIC_KEY_FILE` adalah jalur ke file kunci publik Anda, seperti `my_key.pub` **Important** Pastikan untuk menggunakan kunci publik dan bukan kunci pribadi. `aws ec2-instance-connect send-ssh-public-key \ --instance-id $INSTANCE_ID \ --instance-os-user ec2-user \ --ssh-public-key file://$PUBLIC_KEY_FILE` 2. Tunggu hingga Anda menerima pesan yang menunjukkan bahwa kunci telah berhasil diunggah. Lanjutkan ke langkah berikutnya segera.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke host bastion.	1. Masukkan perintah berikut untuk terhubung ke host bastion melalui Session Manager, di mana: • `$PRIVATE_KEY_FILE` adalah jalan menuju kunci pribadi Anda, seperti `my_key` • `$INSTANCE_ID` adalah ID dari EC2 instance Amazon `ssh -i $PRIVATE_KEY_FILE ec2-user@$INSTANCE_ID` 2. Konfirmasikan koneksi dengan memasukkan`yes`. Ini membuka koneksi SSH dengan menggunakan Session Manager. **Note** Ada opsi lain untuk membuka koneksi SSH dengan host bastion. Untuk informasi lebih lanjut, lihat Pendekata n alternatif untuk membuat koneksi SSH dengan host bastion	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di bagian Informasi tambahan dari pola ini.	

(Opsional) Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya yang digunakan.	1. Untuk menghapus semua sumber daya yang digunakan, jalankan perintah berikut dari direktori root repositori kloning. `terraform destroy -var-file="dev.tfvars"` 2. Konfirmasikan penghapusan sumber daya.	DevOps insinyur, Pengembang, Terraform

Pemecahan masalah

Isu	Solusi
`TargetNotConnected` kesalahan saat mencoba terhubung ke host bastion	1. Reboot host bastion sesuai dengan instruksi di Reboot instance Anda di EC2 dokumentasi Amazon. 2. Setelah instance berhasil di-boot ulang, kirim ulang kunci publik ke host bastion dan coba ulang koneksi.
`Permission denied` kesalahan saat mencoba terhubung ke host bastion	Setelah kunci publik diunggah ke host bastion, Anda hanya memiliki 60 detik untuk memulai

Isu	Solusi
	koneksi. Setelah 60 detik, kunci secara otomatis dihapus, dan Anda tidak dapat menggunakannya untuk terhubung ke instance. Jika ini terjadi, Anda dapat mengulangi langkah untuk mengirim ulang kunci ke instance.

Sumber daya terkait

AWS dokumentasi

- AWS Systems Manager Session Manager(Dokumentasi Systems Manager)
- Instal plugin Session Manager untuk dokumentasi AWS CLI(Systems Manager)
- Mengizinkan koneksi SSH untuk Session Manager (dokumentasi Systems Manager)
- Tentang menggunakan EC2 Instance Connect (EC2 dokumentasi Amazon)
- Connect menggunakan EC2 Instance Connect (EC2 dokumentasi Amazon)
- Manajemen identitas dan akses untuk Amazon EC2 (EC2 dokumentasi Amazon)
- Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon (dokumentasi IAM)
- Praktik terbaik keamanan dalam IAM (dokumentasi IAM)
- Kontrol lalu lintas ke sumber daya menggunakan grup keamanan (dokumentasi Amazon VPC)

Sumber daya lainnya

- Halaman web Pengembang Terraform
- Perintah: memvalidasi (dokumentasi Terraform)
- Perintah: fmt (dokumentasi Terraform)
- Menguji HashiCorp Terraform (HashiCorp posting blog)
- Halaman web Checkov

Informasi tambahan

Pendekatan alternatif untuk membangun koneksi SSH dengan host bastion

Penerusan port

Anda dapat menggunakan `-D 8888` opsi untuk membuka koneksi SSH dengan penerusan port dinamis. Untuk informasi lebih lanjut, lihat petunjuk di explainshell.com. Berikut ini adalah contoh perintah untuk membuka koneksi SSH dengan menggunakan port forwarding.

```
ssh -i $PRIVATE_KEY_FILE -D 8888 ec2-user@$INSTANCE_ID
```

Ini adalah jenis koneksi membuka proxy SOCKS yang dapat meneruskan lalu lintas dari browser lokal Anda melalui host bastion. Jika Anda menggunakan Linux atau macOS, untuk melihat semua opsi, masukkan. `man ssh` Ini menampilkan manual referensi SSH.

Menggunakan skrip yang disediakan

Alih-alih menjalankan langkah-langkah yang dijelaskan di Connect to the bastion host secara manual dengan menggunakan Session Manager di bagian Epics, Anda dapat menggunakan skrip connect.sh yang disertakan dalam repositori kode. Skrip ini menghasilkan key pair SSH, mendorong kunci publik ke EC2 instance Amazon, dan memulai koneksi dengan host bastion. Saat Anda menjalankan skrip, Anda meneruskan tag dan nama kunci sebagai argumen. Berikut ini adalah contoh dari perintah untuk menjalankan script.

```
./connect.sh sandbox-dev-bastion-host my_key
```

Memusatkan resolusi DNS dengan menggunakan Microsoft Active AWS Managed Microsoft AD Directory dan lokal

Brian Westmoreland, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memusatkan resolusi DNS dalam lingkungan AWS multi-akun dengan menggunakan AWS Directory Service for Microsoft Active Directory ()AWS Managed Microsoft AD dan Amazon Route 53. Dalam pola ini namespace AWS DNS adalah subdomain dari namespace DNS lokal. Pola ini juga memberikan panduan tentang cara mengonfigurasi server DNS lokal untuk meneruskan kueri AWS saat solusi DNS lokal menggunakan Microsoft Active Directory.

Prasyarat dan batasan

Prasyarat

- Lingkungan AWS multi-akun yang disiapkan dengan menggunakan AWS Organizations.
- Konektivitas jaringan dibangun antara Akun AWS.
- Konektivitas jaringan yang dibangun antara AWS dan lingkungan lokal (dengan menggunakan AWS Direct Connect atau jenis koneksi VPN apa pun).
- AWS Command Line Interface (AWS CLI) dikonfigurasi pada workstation lokal.
- AWS Resource Access Manager (AWS RAM) digunakan untuk membagikan aturan Route 53 antar akun. Oleh karena itu, berbagi harus diaktifkan di dalam AWS Organizations lingkungan, seperti yang dijelaskan di bagian Epik.

Batasan

- AWS Managed Microsoft AD Standard Edition memiliki batas 5 saham.
- AWS Managed Microsoft AD Enterprise Edition memiliki batas 125 saham.
- Solusi dalam pola ini terbatas pada berbagi dukungan Wilayah AWS itu AWS RAM.

Versi produk

- Microsoft Active Directory berjalan pada Windows Server 2008, 2012, 2012 R2, atau 2016.

Arsitektur

Arsitektur target

Dalam desain ini, AWS Managed Microsoft AD dipasang di layanan bersama Akun AWS. Meskipun ini bukan persyaratan, pola ini mengasumsikan konfigurasi ini. Jika Anda mengonfigurasi AWS Managed Microsoft AD yang berbeda Akun AWS, Anda mungkin harus memodifikasi langkah-langkah di bagian Epik yang sesuai.

Desain ini menggunakan Resolver Route 53 untuk mendukung resolusi nama melalui penggunaan aturan Route 53. Jika solusi DNS lokal menggunakan Microsoft DNS, membuat aturan penerusan bersyarat untuk AWS namespace (), yang merupakan subdomain dari namespace DNS perusahaan `aws.company.com` (), tidaklah mudah. `company.com` Jika Anda mencoba membuat forwarder bersyarat tradisional, itu akan menghasilkan kesalahan. Ini karena Microsoft Active Directory sudah dianggap otoritatif untuk setiap subdomain. `company.com` Untuk mengatasi kesalahan ini, Anda harus terlebih dahulu membuat delegasi untuk `aws.company.com` mendelegasikan otoritas namespace itu. Anda kemudian dapat membuat forwarder bersyarat.

Virtual private cloud (VPC) untuk setiap akun spoke dapat memiliki namespace DNS uniknya sendiri berdasarkan namespace root. AWS Dalam desain ini, setiap akun spoke menambahkan singkatan dari nama akun ke namespace AWS dasar. Setelah zona host pribadi di akun spoke telah dibuat, zona dikaitkan dengan VPC lokal di akun spoke serta dengan VPC di akun jaringan pusat. AWS Ini memungkinkan akun AWS jaringan pusat untuk menjawab pertanyaan DNS yang terkait dengan akun spoke. Dengan cara ini, baik Route 53 dan AWS Managed Microsoft AD bekerja sama untuk berbagi tanggung jawab mengelola AWS namespace ()`aws.company.com`.

Otomatisasi dan skala

Desain ini menggunakan titik akhir Route 53 Resolver untuk menskalakan kueri DNS antara AWS dan lingkungan lokal Anda. Setiap titik akhir Route 53 Resolver terdiri dari beberapa antarmuka jaringan elastis (tersebar di beberapa Availability Zone), dan setiap antarmuka jaringan dapat menangani hingga 10.000 kueri per detik. Route 53 Resolver mendukung hingga 6 alamat IP per titik akhir, jadi secara keseluruhan desain ini mendukung hingga 60.000 kueri DNS per detik yang tersebar di beberapa Availability Zone untuk ketersediaan tinggi.

Selain itu, pola ini secara otomatis memperhitungkan pertumbuhan masa depan di dalamnya AWS. Aturan penerusan DNS yang dikonfigurasi di tempat tidak harus dimodifikasi untuk mendukung zona host pribadi baru VPCs dan terkait yang ditambahkan. AWS

Alat

Layanan AWS

- AWS Directory Service for Microsoft Active Directorymemungkinkan beban kerja dan sumber daya yang sadar direktori Anda AWS untuk menggunakan Microsoft Active Directory di file. AWS Cloud
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Resource Access Manager (AWS RAM) membantu Anda berbagi sumber daya dengan aman Akun AWS untuk mengurangi overhead operasional dan memberikan visibilitas dan auditabilitas.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.

Alat

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda. Dalam pola ini, AWS CLI digunakan untuk mengkonfigurasi otorisasi Route 53.

Epik

Membuat dan berbagi AWS Managed Microsoft AD direktori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan AWS Managed Microsoft AD.	1. Buat dan konfigurasikan direktori baru. Untuk langkah-langkah rinci, lihat Membuat Anda AWS Managed Microsoft AD di Panduan AWS Directory Service Administrasi. 2. Catat alamat IP pengontro l AWS Managed Microsoft AD domain. Ini akan direferensikan pada langkah selanjutnya.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bagikan direktori.	Setelah direktori dibangun, bagikan dengan yang lain Akun AWS di AWS organisasi. Untuk petunjuk, lihat Bagikan direktori Anda di Panduan AWS Directory Service Administrasi. **Note** AWS Managed Microsoft AD Standard Edition memiliki batas 5 saham. Enterprise Edition memiliki batas 125 saham.	Administrator AWS

Konfigurasikan Rute 53

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Resolver Rute 53.	Resolver Route 53 memfasili tasi resolusi kueri DNS antara AWS dan pusat data lokal. 1. Instal Resolver Route 53 dengan mengikuti i nstructio ns di Panduan Pengemban g Route 53. 2. Konfigurasikan Resolver Route 53 dalam subnet pribadi di setidaknya dua Availability Zone dalam VPC AWS akun jaringan	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pusat untuk ketersediaan tinggi. **Note** Meskipun menggunakan VPC akun AWS jaringan pusat bukanlah persyaratan, langkah-langkah yang tersisa mengasumsikan konfigurasi ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat aturan Route 53.	Kasus penggunaan spesifik Anda mungkin memerluka n sejumlah besar aturan Route 53, tetapi Anda harus mengonfigurasi aturan berikut sebagai dasar:
	• Aturan keluar untuk namespace lokal (`company.com`) dengan menggunakan akun jaringan pusat Outbound Route 53 Resolvers. Alamat IP target adalah server DNS lokal.
	  • Kaitkan aturan ini dengan akun jaringan pusat VPC.
	• Aturan keluar untuk AWS namespace (`aws.compa ny.com`) dengan menggunakan akun jaringan pusat Outbound Route 53 Resolvers. Alamat IP target adalah akun jaringan pusat inbound Route 53 Resolver IP address.
	  • Jangan kaitkan aturan ini dengan akun AWS jaringan pusat VPC (yang menampung Resolver Route 53).
	• Aturan keluar kedua untuk AWS namespace (`aws.company.com`) yang menunjuk ke
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengontrol AWS Managed Microsoft AD domain (gunakan IPs dari epik sebelumnya). • Kaitkan aturan ini dengan akun AWS jaringan pusat VPC (yang menampung Resolver Route 53). • Jangan berbagi atau mengaitkan aturan ini dengan yang lain Akun AWS. Untuk informasi selengkap nya, lihat Mengelola aturan penerusan di Panduan Pengembang Route 53.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Profil Route 53.	Profil Route 53 digunakan untuk membagikan aturan dengan akun spoke. 1. Buat Profil Route 53 baru di akun jaringan pusat dengan mengikuti petunjuk di Panduan Pengembang Route 53. 2. Tambahkan aturan untuk namespace lokal (`company.com`) ke Profil. 3. Tambahkan aturan pertama untuk AWS namespace (`aws.company.com`), yang menargetkan alamat IP resolver masuk Route 53, ke Profil. 4. Bagikan Profil Route 53 dengan AWS organisasi. 5. Terima pembagian sumber daya Profil Route 53 di setiap akun spoke. 6. Kaitkan Profil Route 53 dengan setiap akun bicara VPC.	Administrator AWS

Konfigurasikan DNS Direktori Aktif lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat delegasi.	Gunakan snap-in (`dnsmgmt.msc`) Microsoft DNS untuk	Active Directory

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membuat delegasi baru untuk `company.com` namespace dalam Active Directory. Nama domain yang didelegasikan harus`aws`. Ini membuat nama domain yang sepenuhnya memenuhi syarat (FQDN) dari delegasi. `aws.company.com` Gunakan alamat IP pengontrol AWS Managed Microsoft AD domain untuk nilai IP server nama, dan gunakan `server.aws.company.com` untuk nama tersebut. (Delegasi ini hanya untuk redundansi, karena forwarder bersyarat akan dibuat untuk namespace ini yang lebih diutamakan daripada delegasi.)	
Buat forwarder bersyarat.	Gunakan snap-in (`dnsmgmt.msc`) Microsoft DNS untuk membuat forwarder bersyarat baru untuk. `aws.company.com` Gunakan alamat IP dari AWS Inbound Route 53 Resolvers di DNS pusat Akun AWS untuk target forwarder bersyarat.	Active Directory

Buat Route 53 zona host pribadi untuk spoke Akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat zona host pribadi Route 53.	Buat zona host pribadi Route 53 di setiap akun spoke. Kaitkan zona host pribadi ini dengan VPC akun spoke. Untuk langkah-langkah mendetail, lihat Membuat zona yang dihosting pribadi di Panduan Pengembang Route 53.	Administrator AWS
Buat otorisasi.	Gunakan AWS CLI untuk membuat otorisasi untuk akun AWS jaringan pusat VPC. Jalankan perintah ini dari konteks setiap pembicaraan Akun AWS: `aws route53 create-vpc-association-authorization --hosted-zone-id <hosted-zone-id> \ --vpc VPCRegion=<region>,VPCId=<vpc-id>` di mana: • `<hosted-zone-id>` adalah zona host pribadi Route 53 di akun spoke. • `<region>` dan `<vpc-id>` merupakan ID VPC Wilayah	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS dan VPC dari akun AWS jaringan pusat VPC.	
Buat asosiasi.	Buat asosiasi zona host pribadi Route 53 untuk VPC akun AWS jaringan pusat dengan menggunakan. AWS CLI Jalankan perintah ini dari konteks akun AWS jaringan pusat: `aws route53 associate -vpc-with-hosted-z one --hosted-zone-id <hosted-zone-id> \ --vpc VPCRegion =<region>,VPCId=<vpc-id>` di mana: • `<hosted-zone-id>`adalah zona host pribadi Route 53 di akun spoke. • `<region>`dan `<vpc-id>` merupakan ID VPC Wilayah AWS dan dari akun AWS jaringan pusat.	Administrator AWS

Sumber daya terkait

- Sederhanakan manajemen DNS di lingkungan multi-akun dengan Route 53 Resolver (posting blog)AWS
- Membuat AWS Managed Microsoft AD (AWS Directory Service dokumentasi) Anda
- Berbagi AWS Managed Microsoft AD direktori (AWS Directory Service dokumentasi)

- Apa itu Amazon Route 53 Resolver? (Dokumentasi Amazon Route 53)
- Membuat zona host pribadi (dokumentasi Amazon Route 53)
- Apa itu Profil Amazon Route 53? (Dokumentasi Amazon Route 53)

Memusatkan pemantauan dengan menggunakan Amazon CloudWatch Observability Access Manager

Anand Krishna Varanasi, JAGDISH KOMAKALA, Ashish Kumar, Jimmy Morgan, Sarat Chandra Pothula, Vivek Thangamuthu, dan Balaji Vedagiri, Amazon Web Services

Ringkasan

Observabilitas sangat penting untuk pemantauan, pemahaman, dan pemecahan masalah aplikasi. Aplikasi yang menjangkau beberapa akun, seperti implementasi dengan AWS Control Tower atau landing zone, menghasilkan sejumlah besar log dan data pelacakan. Untuk memecahkan masalah dengan cepat atau memahami analisis pengguna atau analitik bisnis, Anda memerlukan platform pengamatan umum di semua akun. Amazon CloudWatch Observability Access Manager memberi Anda akses ke, dan kontrol atas, beberapa log akun dari lokasi pusat.

Anda dapat menggunakan Pengelola Akses Observabilitas untuk melihat dan mengelola log data observabilitas yang dihasilkan oleh akun sumber. Akun sumber adalah individu Akun AWS yang menghasilkan data observabilitas untuk sumber daya mereka. Data observabilitas dibagi antara akun sumber dan akun pemantauan. Data observabilitas bersama dapat mencakup metrik di Amazon CloudWatch, log di Amazon CloudWatch Logs, dan trace in. AWS X-Ray Untuk informasi selengkapnya, lihat dokumentasi Observability Access Manager.

Pola ini untuk pengguna yang memiliki aplikasi atau infrastruktur yang berjalan di banyak Akun AWS dan membutuhkan tempat umum untuk melihat log. Ini menjelaskan bagaimana Anda dapat mengatur Observability Access Manager dengan menggunakan Terraform, untuk memantau status dan kesehatan aplikasi atau infrastruktur ini. Anda dapat menginstal solusi ini dengan berbagai cara:

- Sebagai modul Terraform mandiri yang Anda atur secara manual
- Dengan menggunakan pipeline continuous integration dan continuous delivery (CI/CD)
- Dengan mengintegrasikan dengan solusi lain seperti AWS Control Tower Account Factory for Terraform (AFT)

Instruksi di bagian Epik mencakup implementasi manual. Untuk langkah-langkah instalasi AFT, lihat file README untuk repositori GitHub Observability Access Manager.

Prasyarat dan batasan

Prasyarat

- Terraform diinstal atau direferensikan di sistem Anda atau di saluran pipa otomatis. (Kami menyarankan Anda menggunakan versi terbaru.)
- Akun yang dapat Anda gunakan sebagai akun pemantauan pusat. Akun lain membuat tautan ke akun pemantauan pusat untuk melihat log.
- (Opsional) Sebuah repositori kode sumber seperti GitHub,, Atlassian Bitbucket AWS CodeCommit, atau sistem serupa. Repositori kode sumber tidak diperlukan jika Anda menggunakan pipeline otomatis CI/CD .
- (Opsional) Izin untuk membuat permintaan tarik (PRs) untuk peninjauan kode dan kolaborasi kode di GitHub.

Batasan

Observability Access Manager memiliki kuota layanan berikut, yang tidak dapat diubah. Pertimbangkan kuota ini sebelum Anda menerapkan fitur ini. Untuk informasi selengkapnya, lihat kuota CloudWatch layanan dalam CloudWatch dokumentasi.

- Tautan akun sumber: Anda dapat menautkan setiap akun sumber ke maksimal lima akun pemantauan.
- Tenggelam: Anda dapat membuat beberapa sink untuk akun, tetapi hanya satu wastafel per yang Wilayah AWS diizinkan.

Selain itu:

- Wastafel dan tautan harus dibuat dalam hal yang sama Wilayah AWS; mereka tidak bisa lintas wilayah.

Pemantauan Lintas Wilayah dan lintas akun

Untuk pemantauan lintas wilayah dan lintas akun, Anda dapat memilih salah satu opsi ini:

- Buat CloudWatch dasbor lintas akun dan lintas wilayah untuk alarm dan metrik. Opsi ini tidak mendukung log dan jejak.
- Menerapkan logging terpusat dengan menggunakan Amazon OpenSearch Service.
- Buat satu sink per Wilayah dari semua akun penyewa, dorong metrik ke akun pemantauan terpusat (seperti yang dijelaskan dalam pola ini), lalu gunakan aliran CloudWatch metrik untuk

mengirim data ke tujuan eksternal umum atau ke produk pemantauan pihak ketiga seperti Datadog, Dynatrace, Sumo Logic, Splunk, atau New Relic.

Arsitektur

Komponen

CloudWatch Observability Access Manager terdiri dari dua komponen utama yang memungkinkan observabilitas lintas akun:

- Wastafel menyediakan kemampuan untuk akun sumber untuk mengirim data observabilitas ke akun pemantauan pusat. Wastafel pada dasarnya menyediakan persimpangan gateway untuk akun sumber untuk terhubung. Hanya ada satu gateway atau koneksi sink, dan beberapa akun dapat terhubung dengannya.
- Setiap akun sumber memiliki tautan ke persimpangan gateway sink, dan data observabilitas dikirim melalui tautan ini. Anda harus membuat wastafel sebelum membuat tautan dari setiap akun sumber.

Arsitektur

Diagram berikut menggambarkan Observability Access Manager dan komponen-komponennya.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Alat

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.
- AWS Control Tower Account Factory for Terraform (AFT) menyiapkan pipeline Terraform untuk membantu Anda menyediakan dan menyesuaikan akun. AWS Control Tower Anda dapat menggunakan AFT secara opsional untuk menyiapkan Observability Access Manager dalam skala besar di beberapa akun.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Observability Access Manager.

Praktik terbaik

- Di AWS Control Tower lingkungan, tandai akun logging sebagai akun pemantauan pusat (sink).
- Jika Anda memiliki beberapa organisasi dengan beberapa akun AWS Organizations, sebaiknya sertakan organisasi, bukan akun individual dalam kebijakan konfigurasi. Jika Anda memiliki sejumlah kecil akun atau jika akun tersebut bukan bagian dari organisasi dalam kebijakan konfigurasi sink, Anda dapat memutuskan untuk menyertakan akun individual sebagai gantinya.

Epik

Siapkan modul wastafel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Kloning repositori GitHub Observability Access Manager: `git clone https://github.com/aws-samples/cloudwatch-observability-access-manager-terraform`	AWS DevOps, administrator Cloud, administrator AWS
Tentukan nilai properti untuk modul wastafel.	Dalam `main.tf` file (dalam `deployments/aft-ac`	AWS DevOps, administrator Cloud, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`count-customizations/LOGGING/terraform/` folder repositori), tentukan nilai untuk properti berikut: • `sink_name` : Nama CloudWatch wastafel. • `allowed_oam_resource_types` : Pengelola Akses Observabilitas saat ini mendukung CloudWatch metrik, grup log, dan AWS X-Ray jejak. • `allowed_source_accounts` : Akun sumber yang diizinkan untuk mengirim log ke akun CloudWatch wastafel pusat. • `allowed_source_organizations` : AWS Control Tower Organisasi sumber yang diizinkan mengirim log ke akun CloudWatch wastafel pusat. Untuk informasi lebih lanjut, lihat AWS::Oam::Sinkdi CloudFormation dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Pasang modul wastafel.	Ekspor kredensil Akun AWS yang telah Anda pilih sebagai akun pemantauan, dan instal modul sink Observability Access Manager:
	`Terraform Init`
	`Terrafom Plan`
	`Terraform Apply`
		AWS DevOps, administrator Cloud, administrator AWS

Siapkan modul tautan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tentukan nilai properti untuk modul tautan.	Dalam `main.tf` file (dalam `deployments/aft-account-customizations/LOGGING/terraform/` folder repositori), tentukan nilai untuk properti berikut:
	• `account_label` : Gunakan salah satu nilai berikut:
	  • `$AccountName` : Nama akun.
	  • `$AccountEmail` Alamat email yang unik secara global, yang mencakup domain email (misalnya ,`hello@example.com`)
		AWS DevOps, Administrator cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `$AccountEmailNoDomain` Alamat email tanpa nama domain • `allowed_oam_resource_types` : Pengelola Akses Observabilitas saat ini mendukung CloudWatch metrik, grup log, dan AWS X-Ray jejak. Untuk informasi lebih lanjut, lihat AWS::Oam::Linkdi CloudFormation dokumentasi.	

Instal modul tautan untuk akun individual.	Ekspor kredensil akun individual dan instal modul tautan Observability Access Manager:
	
	`Terraform Plan`
	`Terraform Apply`
	
	Anda dapat mengatur modul tautan satu per satu untuk setiap akun, atau menggunakan AFT untuk menginstal modul ini secara otomatis di sejumlah besar akun.
		AWS DevOps, Administrator cloud, arsitek Cloud

Menyetujui koneksi sink-to-link

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa pesan status.	1. Masuk ke akun pemantaua n. 2. Buka konsol CloudWatch . 3. Pada panel navigasi kiri, pilih Pengaturan. Di sebelah kanan, Anda akan melihat pesan status Akun pemantauan diaktifkan dengan tanda centang hijau. Ini berarti bahwa akun pemantauan memiliki wastafel Manajer Akses Observabilitas yang akan terhubung ke tautan akun lain.	
Menyetujui link-to-sink koneksi.	1. Pilih opsi Sumber Daya untuk menautkan akun di bawah pesan status. Informasi tersebut mengonfirmasi bahwa ini adalah akun pemantauan, mencantumkan data yang dibagikan dari akun sumber penyewa (Log, Metrik, Jejak), dan menampilk an label akun sebagai $. AccountName Layar ini menyediakan dua opsi untuk menautkan akun penyewa ke akun pemantauan: persetuju	AWS DevOps, Administrator cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an tingkat organisasi atau persetujuan tingkat akun. Untuk setiap opsi, Anda dapat memilih untuk mengunduh CloudForm ation templat untuk persetujuan atau menyetuju i setiap akun satu per satu. 2. Untuk mempermudah, pilih Akun Apa saja yang akan disetujui di setiap tingkat akun. Opsi ini menyediak an tautan persetujuan untuk akun tersebut. 3. Pilih Salin URL untuk menyalin tautan. 4. Masuk ke setiap akun sumber. 5. Di jendela browser, tempel tautan, dan pilih Setujui tautan sambungkan ke sink. 6. Ulangi untuk akun sumber tambahan. Untuk informasi selengkap nya, lihat Menautkan akun pemantauan dengan akun sumber dalam CloudWatch dokumentasi.	

Verifikasi data observabilitas lintas akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat data lintas akun.	1. Masuk ke akun pemantaua n pusat. 2. Buka konsol CloudWatch . 3. Di panel navigasi kiri, pilih opsi untuk melihat log, metrik, dan jejak lintas akun.	AWS DevOps, Administrator cloud, arsitek Cloud

(Opsional) Aktifkan akun sumber untuk mempercayai akun pemantauan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat metrik, dasbor, log, widget, dan alarm dari akun lain.	Sebagai fitur tambahan, Anda dapat berbagi CloudWatch metrik, dasbor, log, widget, dan alarm dengan akun lain. Setiap akun menggunak an peran IAM yang disebut CloudWatch- CrossAcco untSharingRole untuk mendapatkan akses ke data ini. Akun sumber yang memiliki hubungan kepercayaan dengan akun pemantauan pusat dapat mengambil peran ini dan melihat data dari akun pemantauan. CloudWatch menyediakan contoh CloudFormation skrip	AWS DevOps, Administrator cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk membuat peran. Pilih Kelola peran di IAM dan jalankan skrip ini di akun tempat Anda ingin melihat data. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": [ "arn:aws:iam::XXXXXXXXXX:root", "arn:aws:iam::XXXXXXXXXX:root", "arn:aws:iam::XXXXXXXXXX:root", "arn:aws:iam::XXXXXXXXXX:root" ] }, "Action": "sts:AssumeRole" } ] }` Untuk informasi selengkapnya, lihat Mengaktifkan fungsionalitas lintas akun	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CloudWatch dalam dokumentasi. CloudWatch	

(Opsional) Lihat lintas akun Lintas wilayah dari akun pemantauan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan akses lintas akun, lintas wilayah.	Di akun pemantauan pusat, Anda dapat menambahkan pemilih akun untuk dengan mudah beralih antar akun dan melihat datanya tanpa harus mengautentikasi. 1. Masuk ke akun pemantauan pusat. 2. Buka konsol CloudWatch . 3. Pada panel navigasi kiri, pilih Pengaturan. 4. Di bagian Lihat lintas wilayah lintas akun, pilih Konfigurasi. 5. Pilih Aktifkan, lalu pilih kotak centang Tampilkan pemilih di konsol. 6. Pilih salah satu opsi ini: • Input Id Akun: Opsi ini meminta Anda untuk memasukkan ID akun secara manual kapan pun Anda ingin mengubah akun untuk melihat data lintas akun.	AWS DevOps, Administrator cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Pemilih akun AWS Organization: Jika Anda telah terintegrasi CloudWatch AWS Organizations, opsi ini menyediakan pemilih tarik-turun dengan daftar lengkap akun di organisasi. • Pemilih akun khusus: Opsi ini memungkinkan Anda memasukkan daftar akun secara manual IDs untuk mengisi pemilih. 7. Pilih Simpan perubahan. Untuk informasi selengkapnya, lihat CloudWatch Konsol lintas wilayah lintas akun di dokumentasi. CloudWatch	

Sumber daya terkait

- CloudWatch observabilitas lintas akun (dokumentasi Amazon CloudWatch)
- Referensi API Manajer Akses CloudWatch Observabilitas Amazon (CloudWatch dokumentasi Amazon)
- Sumber daya: aws_oam_sink (dokumentasi Terraform)
- Sumber Data: aws_oam_link (Dokumentasi Terraform)
- CloudWatchObservabilityAccessManager(AWS Dokumentasi Boto3)

Periksa EC2 instance untuk tag wajib saat peluncuran

Susanne Kangnoh dan Archit Mathur, Amazon Web Services

Ringkasan

Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di Amazon Web Services (AWS) Cloud. Menggunakan Amazon EC2 menghilangkan kebutuhan Anda untuk berinvestasi dalam perangkat keras di muka, sehingga Anda dapat mengembangkan dan menerapkan aplikasi lebih cepat.

Anda dapat menggunakan penandaan untuk mengkategorikan sumber daya AWS Anda dengan berbagai cara. EC2 penandaan instance berguna ketika Anda memiliki banyak sumber daya di akun Anda dan Anda ingin mengidentifikasi sumber daya tertentu dengan cepat berdasarkan tag. Anda dapat menetapkan metadata kustom ke EC2 instance Anda dengan menggunakan tag. Tag terdiri dari kunci dan nilai yang ditentukan pengguna. Kami menyarankan Anda membuat kumpulan tag yang konsisten untuk memenuhi persyaratan organisasi Anda.

Pola ini menyediakan CloudFormation template AWS untuk membantu Anda memantau EC2 instance untuk tag tertentu. Template membuat CloudWatch acara Amazon Events yang mengawasi AWS CloudTrail TagResourceatau UntagResourceperistiwa, untuk mendeteksi penandaan EC2 instans baru atau penghapusan tag. Jika tag yang telah ditentukan tidak ada, tag tersebut akan memanggil fungsi AWS Lambda, yang mengirimkan pesan pelanggaran ke alamat email yang Anda berikan, dengan menggunakan Amazon Simple Notification Service (Amazon SNS).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Bucket Amazon Simple Storage Service (Amazon S3) untuk mengunggah kode Lambda yang disediakan.
- Alamat email tempat Anda ingin menerima pemberitahuan pelanggaran.

Batasan

- Solusi ini mendukung CloudTrail TagResourceatau UntagResourceacara. Itu tidak membuat pemberitahuan untuk acara lainnya.

- Solusi ini hanya memeriksa kunci tag. Itu tidak memantau nilai-nilai kunci.

Arsitektur

Arsitektur alur kerja

Otomatisasi dan skala

- Anda dapat menggunakan CloudFormation template AWS beberapa kali untuk Wilayah dan akun AWS yang berbeda. Anda perlu menjalankan template hanya sekali di setiap Wilayah atau akun.

Alat

Layanan AWS

- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang aman dan dapat diubah ukurannya di cloud. Ini dirancang untuk membuat komputasi awan skala web lebih mudah bagi pengembang.
- AWS CloudTrail — CloudTrail adalah layanan AWS yang membantu Anda dalam tata kelola, kepatuhan, serta audit operasional dan risiko akun AWS Anda. Tindakan yang diambil oleh pengguna, peran, atau layanan AWS dicatat sebagai peristiwa di CloudTrail.
- Amazon CloudWatch Events — Amazon CloudWatch Events menghadirkan aliran peristiwa sistem yang mendekati real-time yang menjelaskan perubahan dalam sumber daya AWS. CloudWatch Peristiwa menjadi sadar akan perubahan operasional saat terjadi dan mengambil tindakan korektif seperlunya, dengan mengirim pesan untuk merespons lingkungan, mengaktifkan fungsi, membuat perubahan, dan menangkap informasi negara.
- AWS Lambda — Lambda adalah layanan komputasi yang mendukung kode berjalan tanpa perlu menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel yang dapat digunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan danau data.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) adalah layanan web yang memungkinkan aplikasi, pengguna akhir, dan perangkat untuk langsung mengirim dan menerima pemberitahuan dari cloud.

Kode

Pola ini mencakup lampiran dengan dua file:

- `index.zip`adalah file terkompresi yang menyertakan kode Lambda untuk pola ini.
- `ec2-require-tags.yaml`adalah CloudFormation template yang menyebarkan kode Lambda.

Lihat bagian Epics untuk informasi tentang cara menggunakan file-file ini.

Epik

Terapkan kode Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode ke bucket S3.	Buat bucket S3 baru atau gunakan bucket S3 yang sudah ada untuk mengungga h `index.zip` file terlampir (kode Lambda). Bucket ini harus berada di Wilayah AWS yang sama dengan sumber daya (EC2 instance) yang ingin Anda pantau.	Arsitek awan
Menyebarkan CloudFormation template.	Buka konsol Cloudform ation di Wilayah AWS yang sama dengan bucket S3, dan terapkan `ec2-requi re-tags.yaml` file yang disediakan di lampiran. Dalam epik berikutnya, berikan nilai untuk parameter template.	Arsitek awan

Lengkapi parameter dalam CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan nama bucket S3.	Masukkan nama bucket S3 yang Anda buat atau pilih di epik pertama. Bucket S3 ini berisi file.zip untuk kode Lambda dan harus berada di Wilayah AWS yang sama dengan CloudFormation template dan EC2 instance yang ingin Anda pantau.	Arsitek awan
Berikan kunci S3.	Berikan lokasi file kode Lambda .zip di bucket S3 Anda, tanpa garis miring di depan (misalnya, atau). `index.zip controls/index.zip`	Arsitek awan
Berikan alamat email.	Berikan alamat email aktif tempat Anda ingin menerima pemberitahuan pelanggaran.	Arsitek awan
Tentukan tingkat logging.	Tentukan tingkat logging dan verbositas. `Info`menunjuk pesan informasi rinci tentang kemajuan aplikasi dan harus digunakan hanya untuk debugging. `Error`menunjuk peristiwa kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. `Warning`menunjuk situasi yang berpotensi berbahaya.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masukkan kunci tag yang diperlukan.	Masukkan tombol tag yang ingin Anda periksa. Jika Anda ingin menentukan beberapa kunci, pisahkan dengan koma, tanpa spasi. (Misalnya , `ApplicationId,CreatedBy,Environment, Organization` mencari empat kunci.) CloudWatch Acara Acara mencari kunci tag ini dan mengirimkan pemberitahuan jika tidak ditemukan.	Arsitek awan

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan langganan email.	Ketika CloudFormation template berhasil digunakan , ia mengirimkan pesan email berlangganan ke alamat email yang Anda berikan. Untuk menerima pemberitahuan, Anda harus mengonfirmasi langganan email ini.	Arsitek awan

Sumber daya terkait

- Membuat ember (dokumentasi Amazon S3)
- Mengunggah objek (dokumentasi Amazon S3)
- Tandai EC2 sumber daya Amazon Anda (EC2 dokumentasi Amazon)
- Membuat aturan CloudWatch Peristiwa yang dipicu pada panggilan AWS API menggunakan AWS CloudTrail (CloudWatch dokumentasi Amazon)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Bersihkan AWS Account Factory untuk sumber daya Terraform (AFT) dengan aman setelah kehilangan file status

Gokendra Malviya, Amazon Web Services

Ringkasan

Saat Anda menggunakan AWS Account Factory for Terraform (AFT) untuk mengelola AWS Control Tower lingkungan Anda, AFT menghasilkan file status Terraform untuk melacak status dan konfigurasi sumber daya yang dibuat oleh Terraform. Kehilangan file status Terraform dapat menciptakan tantangan signifikan untuk manajemen dan pembersihan sumber daya. Pola ini memberikan pendekatan sistematis untuk mengidentifikasi dan menghapus sumber daya terkait AFT dengan aman sambil menjaga integritas lingkungan Anda AWS Control Tower .

Proses ini dirancang untuk memastikan penghapusan yang tepat dari semua komponen AFT, bahkan tanpa referensi file status asli. Proses ini memberikan jalur yang jelas untuk berhasil membangun kembali dan mengkonfigurasi ulang AFT di lingkungan Anda, untuk membantu memastikan gangguan minimal pada operasi Anda. AWS Control Tower

Untuk informasi lebih lanjut tentang AFT, lihat AWS Control Tower dokumentasi.

Prasyarat dan batasan

Prasyarat

- Pemahaman menyeluruh tentang arsitektur AFT.
- Akses administrator ke akun berikut:
  - Akun Manajemen AFT
  - AWS Control Tower Akun manajemen
  - Akun Arsip Log
  - Akun audit
- Verifikasi bahwa tidak ada kebijakan kontrol layanan (SCPs) yang mengandung batasan atau batasan yang akan memblokir penghapusan sumber daya terkait AFT.

Batasan

- Proses ini dapat membersihkan sumber daya secara efektif, tetapi tidak dapat memulihkan file status yang hilang, dan beberapa sumber daya mungkin memerlukan identifikasi manual.

- Durasi proses pembersihan tergantung pada kompleksitas lingkungan Anda dan mungkin memakan waktu beberapa jam.
- Pola ini telah diuji dengan AFT versi 1.12.2 dan menghapus sumber daya berikut. Jika Anda menggunakan versi AFT yang berbeda, Anda mungkin harus menghapus sumber daya tambahan.

Nama layanan	Jumlah sumber daya
AWS CodeBuild	6
AWS CodeCommit	4
AWS CodePipeline	4
Amazon DynamoDB	5
Amazon Elastic Compute Cloud (Amazon EC2)	16
Amazon EventBridge	4
AWS Identity and Access Management Peran (IAM)	40
AWS Key Management Service (AWS KMS)	2
AWS Lambda	17
Amazon Simple Storage Service (Amazon S3)	2
Amazon Simple Notification Service (Amazon SNS)	2
Amazon Simple Queue Service (Amazon SQS)	2
AWS Systems Manager	62
AWS Step Functions	4

> **Important**
>
> Sumber daya yang dihapus oleh langkah-langkah dalam pola ini tidak dapat dipulihkan. Sebelum Anda mengikuti langkah-langkah ini, verifikasi nama sumber daya dengan hati-hati dan pastikan nama tersebut dibuat oleh AFT.

Arsitektur

Diagram berikut menunjukkan komponen AFT dan alur kerja tingkat tinggi. AFT menyiapkan pipeline Terraform yang membantu Anda menyediakan dan menyesuaikan akun Anda. AWS Control Tower AFT mengikuti GitOps model untuk mengotomatiskan proses penyediaan akun di. AWS Control Tower Anda membuat file Terraform untuk permintaan akun dan mengkomitmenkannya ke repositori, yang menyediakan input yang memicu alur kerja AFT untuk penyediaan akun. Setelah penyediaan akun selesai, AFT dapat menjalankan langkah penyesuaian tambahan secara otomatis.

Dalam arsitektur ini:

- AWS Control Tower Akun manajemen adalah akun Akun AWS yang didedikasikan untuk AWS Control Tower layanan ini. Ini juga biasanya disebut sebagai akun AWS pembayar atau akun AWS Organizations Manajemen.
- Akun Manajemen AFT adalah akun Akun AWS yang didedikasikan untuk operasi manajemen AFT. Ini berbeda dengan akun manajemen organisasi Anda.
- Vending account adalah akun Akun AWS yang berisi semua komponen dasar dan kontrol yang Anda pilih. AFT menggunakan AWS Control Tower untuk menjual akun baru.

Untuk informasi tambahan tentang arsitektur ini, lihat Pengantar AFT di AWS Control Tower bengkel.

Alat

Layanan AWS

- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif.
- AWS Account Factory for Terraform (AFT) menyiapkan pipeline Terraform untuk membantu Anda menyediakan dan menyesuaikan akun dan sumber daya. AWS Control Tower

- AWS Organizationsmembantu Anda mengelola dan mengatur lingkungan Anda secara terpusat saat Anda tumbuh dan meningkatkan sumber daya Anda AWS . Dengan menggunakan Organizations, Anda dapat membuat akun dan mengalokasikan sumber daya, mengelompokkan akun untuk mengatur alur kerja, menerapkan kebijakan tata kelola, dan menyederhanakan penagihan dengan menggunakan satu metode pembayaran untuk semua akun Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya. Pola ini membutuhkan peran dan izin IAM.

Alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Praktik terbaik

- Untuk AWS Control Tower, lihat Praktik terbaik untuk AWS Control Tower administrator dalam AWS Control Tower dokumentasi.
- Untuk IAM, lihat Praktik terbaik keamanan dalam dokumentasi IAM.

Epik

Hapus sumber daya AFT di akun Manajemen AFT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya yang diidentifikasi oleh tag AFT.	1. Masuk ke akun manajemen AFT dengan izin administrator. 2. Buka konsol Grup Sumber Daya AWS. 3. Pilih Wilayah tempat AWS Control Tower telah digunakan.	Administrator AWS, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Di panel navigasi, pilih Editor Tag. 5. Untuk jenis Sumber Daya, pilih Semua jenis sumber daya yang didukung. 6. Untuk Tag, ketik managed_by sebagai kunci tag dan AFT sebagai nilai tag. 7. Pilih Cari sumber daya. Pencarian ini menampilkan semua sumber daya yang dibuat oleh AFT. 8. Identifikasi nama sumber daya dan hapus dengan menggunakan konsol layanan yang sesuai. Misalnya, untuk menghapus sumber daya Parameter Store: a. Buka konsol AWS Systems Manager. b. Di panel navigasi, pilih Penyimpanan Parameter . c. Di kotak pencarian, klik untuk menampilkan dropdown, pilih Nama, pilih sama, lalu ketik /aft. d. Hapus parameter dalam batch 10. (Ini adalah jumlah maksimum yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dapat Anda hapus secara bersamaan.) Untuk AFT versi 1.12.2, akan ada sekitar 62 sumber daya Parameter Store untuk dihapus. Semua nama parameter akan dimulai dengan/aft. Namun, tidak semua sumber daya dapat diidentif ikasi oleh Grup Sumber Daya AWS. Pada langkah-langkah berikut, Anda akan menemukan dan menghapus sumber daya yang tersisa.	
Hapus peran IAM.	1. Masuk ke akun manajemen AFT dengan izin administr ator. 2. Buka konsol IAM. 3. Hapus peran ini dalam urutan yang terdaftar (urutannya penting karena dependensi): • `aft-*` • `AWSAFTAdmin` • `AWSAFTExecution` • `AWSAFTService` • `codebuild_trigger_role`	Administrator AWS, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus brankas AWS Backup cadangan.	1. Buka konsol AWS Backup. 2. Temukan brankas cadangan bernama`aws_backup_vault` . 3. Konfirmasikan bahwa vault tidak berisi cadangan aktif apa pun. 4. Hapus`aws_backup_vault` .	Administrator AWS, AWS DevOps, DevOps insinyur
Hapus CloudWatch sumber daya Amazon.	1. Buka konsol CloudWatch . 2. Hapus sumber daya berikut dalam urutan yang tercantum: a. Bus acara: Hapus`aws_cloudwatch_event_bus` . b. Log: Cari awalan AFT dan hapus semua grup log terkait. c. Definisi kueri: Hapus kueri berikut: • `Customization Logs by Account ID` • `Customization Logs by Customization Request ID`	Administrator AWS, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus AWS KMS sumber daya.	1. Beralih ke Wilayah sekunder, yang berfungsi sebagai backend untuk melacak status status AFT sendiri. 2. Buka konsol AWS KMS. 3. Hapus alias bernama AFT.	Administrator AWS, AWS DevOps, DevOps insinyur

Hapus sumber daya AFT di akun Arsip Log

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus ember S3.	1. Masuk ke akun Arsip Log dengan izin administrator. 2. Buka konsol Amazon S3. 3. Kosongkan ember berikut: • `aws-aft-logs-471112509802-us-east-1` • `aws-aft-s3-access-logs-471112509802-us-east-1` (Ganti `111122223333` dengan ID akun Anda.) 4. Hapus dua ember.	Administrator AWS, AWS DevOps, DevOps insinyur
Hapus peran IAM.	1. Buka konsol IAM. 2. Verifikasi bahwa peran berikut tidak digunakan oleh layanan aktif apa pun: • `AWSAFTService`	Administrator AWS, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `AWSAFTExecution` 3. Hapus kedua peran tersebut.	

Hapus sumber daya AFT di akun Audit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus peran IAM.	1. Masuk ke akun Audit dengan izin administrator. 2. Buka konsol IAM. 3. Verifikasi bahwa peran berikut tidak digunakan oleh layanan aktif apa pun: • `AWSAFTService` • `AWSAFTExecution` 4. Hapus kedua peran tersebut.	Administrator AWS, AWS DevOps, DevOps insinyur

Hapus sumber daya AFT di akun AWS Control Tower Manajemen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus peran IAM.	1. Masuk ke akun AWS Control Tower Manajemen dengan izin administrator. 2. Buka konsol IAM. 3. Verifikasi bahwa peran berikut tidak digunakan oleh layanan aktif apa pun: • `AWSAFTService`	Administrator AWS, AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `AWSAFTExecution` • `aft-control-tower-events-rule` 4. Hapus ketiga peran tersebut.	
Hapus EventBridge aturan.	1. Buka EventBridge konsol Amazon. 2. Di panel navigasi kiri, pilih Aturan. 3. Temukan dan pilih aturan bernama`aft-capture-ct-events` . 4. Pilih Hapus dan konfirmasikan penghapusan saat diminta.	Administrator AWS, AWS DevOps, DevOps insinyur

Pemecahan masalah

Isu	Solusi
Melepaskan gateway internet tidak berhasil.	Saat menghapus sumber daya yang diidentifikasi oleh tag AFT, jika mengalami masalah ini saat melepaskan atau menghapus gateway internet, pertama-tama Anda harus menghapus titik akhir VPC: 1. Masuk ke akun Manajemen AFT, lalu buka konsol Amazon VPC. 2. Di panel navigasi, dalam daftar Filter menurut VPC, pilih VPC bernama. aft-management-vpc 3. Di panel navigasi, pilih Titik Akhir.

Isu	Solusi
	4. Pilih titik akhir yang terkait dengan VPC aft-management-vpc. • Periksa kembali kolom ID VPC sebelum penghapusan untuk menghindari penghapusan titik akhir yang salah. • Berhati-hatilah untuk menghapus hanya titik akhir yang terkait dengan VPC AFT. 5. Pilih Tindakan, Hapus titik akhir VPC. 6. Di kotak dialog konfirmasi, ketik hapus lalu pilih Hapus. 7. Tunggu status endpoint berubah menjadi Dihapus. Penghapusan mungkin memakan waktu beberapa menit untuk diselesaikan.

Isu	Solusi
Anda tidak dapat menemukan CloudWatch kueri yang ditentukan.	Jika Anda tidak dapat menemukan CloudWatch kueri yang dibuat oleh AFT, ikuti langkah-langkah berikut: 1. Masuk ke akun Manajemen AFT, lalu buka CloudWatch konsol. 2. Di panel navigasi, di bawah Log, pilih Wawasan Log. 3. Di sudut kanan atas, pilih ikon Kueri tersimpan dan sampel. Anda sekarang harus dapat melihat kueri AFT. Untuk tangkapan layar, lihat bagian Informasi tambahan. 4. Pilih kueri berikut, lalu pilih Tindakan, Hapus untuk menghapusnya. • `Customization Logs by Account ID` • `Customization Logs by Customization Request ID`

Sumber daya terkait

- BELAKANG:
  - GitHub Repositori
  - Lokakarya
  - Dokumentasi
- AWS Control Tower dokumentasi

Informasi tambahan

Untuk melihat kueri AFT di dasbor Wawasan CloudWatch Log, pilih ikon Kueri tersimpan dan contoh dari sudut kanan atas, seperti yang diilustrasikan pada tangkapan layar berikut:

Buat pipeline di Wilayah AWS yang tidak mendukung AWS CodePipeline

Anand Krishna Varanasi, Amazon Web Services

Ringkasan

Pemberitahuan: AWS CodeCommit tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS CodeCommit dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

AWS CodePipeline adalah layanan orkestrasi pengiriman berkelanjutan (CD) yang merupakan bagian dari seperangkat DevOps alat dari Amazon Web Services (AWS). Ini terintegrasi dengan berbagai macam sumber (seperti sistem kontrol versi dan solusi penyimpanan), produk dan layanan integrasi berkelanjutan (CI) dari AWS dan AWS Partners, dan produk sumber terbuka untuk menyediakan layanan end-to-end alur kerja untuk penerapan aplikasi dan infrastruktur yang cepat.

Namun, CodePipeline tidak didukung di semua Wilayah AWS, dan berguna untuk memiliki orkestrator tak terlihat yang menghubungkan layanan AWS. CI/CD Pola ini menjelaskan cara menerapkan end-to-end alur kerja di Wilayah AWS yang CodePipeline belum didukung dengan menggunakan CI/CD layanan AWS seperti AWS CodeCommit, AWS CodeBuild, dan AWS CodeDeploy.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Cloud Development Kit (AWS CDK) CLI versi 2.28 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

Diagram berikut menunjukkan pipa yang dibuat di Wilayah yang tidak mendukung CodePipeline, seperti Wilayah Afrika (Cape Town). Pengembang mendorong file CodeDeploy konfigurasi (juga disebut skrip hook siklus hidup penerapan) ke repositori Git yang di-host oleh. CodeCommit (Lihat GitHub repositori yang disediakan dengan pola ini.) EventBridge Aturan Amazon secara otomatis dimulai. CodeBuild

File CodeDeploy konfigurasi diambil dari CodeCommit sebagai bagian dari tahap sumber pipa dan ditransfer ke CodeBuild.

Pada fase berikutnya, CodeBuild melakukan tugas-tugas ini:

1. Mengunduh file TAR kode sumber aplikasi. Anda dapat mengonfigurasi nama file ini dengan menggunakan Parameter Store, kemampuan AWS Systems Manager.
2. Mengunduh file CodeDeploy konfigurasi.
3. Membuat arsip gabungan kode sumber aplikasi dan file CodeDeploy konfigurasi yang khusus untuk jenis aplikasi.
4. Memulai CodeDeploy penerapan ke instans Amazon Elastic Compute Cloud (Amazon EC2) dengan menggunakan arsip gabungan.

Alat

Layanan AWS

- AWS CodeBuild adalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodeDeploy mengotomatiskan penerapan ke Amazon EC2 atau instans lokal, fungsi AWS Lambda, atau layanan Amazon Elastic Container Service (Amazon ECS).
- AWS CodePipeline membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.

Kode

Kode untuk pola ini tersedia di repositori Wilayah GitHub CodePipeline Tidak Didukung.

Epik

Siapkan workstation pengembang Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS CDK CLI.	Untuk petunjuk, lihat dokumentasi AWS CDK.	AWS DevOps
Instal klien Git.	Untuk membuat commit, Anda dapat menggunakan klien Git yang diinstal pada komputer lokal Anda, dan kemudian mendorong commit Anda ke repositori. CodeCommit Untuk mengatur CodeCommi t dengan klien Git Anda, lihat CodeCommit dokumentasinya.	AWS DevOps
Instal npm.	Instal manajer paket npm. Untuk informasi selengkapnya, lihat dokumentasi npm.	AWS DevOps

Siapkan pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode.	Kloning repositori Wilayah GitHub CodePipeline Tidak Didukung ke mesin lokal Anda dengan menjalankan perintah berikut. `git clone https://g ithub.com/aws-samp les/invisible-code`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`pipeline-unsupported-regions`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan parameter di cdk.json.	Buka `cdk.json` file dan berikan nilai untuk parameter berikut: `"pipeline_account" :"XXXXXXXXXXXX", "pipeline_region":" us-west-2", "repo_name": "app-dev-repo", "ec2_tag_key": "test-vm", "configName" : "cbdeployconfig", "deploymentGroupNa me": "cbdeploygroup", "applicationName" : "cbdeployapplicati on", "projectName" : "CodeBuildProject"` di mana: • `pipeline_ account` adalah akun AWS tempat pipeline akan dibangun. • `pipeline_region` adalah Wilayah AWS tempat pipa akan dibangun. • `repo_name` adalah nama CodeCommit repositori. • `ec2_tag_key` adalah tag yang dilampirkan pada EC2 instance yang ingin Anda gunakan kodenya.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `configName` adalah nama file CodeDeploy konfigurasi. • `deploymentGroupNam e` adalah nama grup CodeDeploy penyebaran. • `applicationName` adalah nama CodeDeploy aplikasi. • `projectName` adalah nama CodeBuild proyek.	

Siapkan pustaka konstruksi AWS CDK.	Di GitHub repositori kloning, gunakan perintah berikut untuk menginstal pustaka konstruksi AWS CDK, membangun aplikasi Anda, dan mensintesis untuk menghasilkan template AWS CloudFormation untuk aplikasi.
	`npm i aws-cdk-lib`
	`npm run build`
	`cdk synth`
		AWS DevOps

Terapkan contoh aplikasi AWS CDK.	Menyebarkan kode dengan menjalankan perintah berikut di Wilayah yang tidak didukung (seperti`af-south-1`). `cdk deploy`	AWS DevOps

Siapkan CodeCommit repositori untuk CodeDeploy

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan CI/CD untuk aplikasi.	Kloning CodeCommit repositori yang Anda tentukan dalam `cdk.json` file (ini disebut secara `app-dev-repo` default) untuk mengatur CI/CD pipeline untuk aplikasi. `git clone https://git-codecommit.us-west-2.amazonaws.com/v1/repos/app-dev-repo` di mana nama repositori dan Wilayah bergantung pada nilai yang Anda berikan dalam file. `cdk.json`	AWS DevOps

Uji pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji pipa dengan instruksi penerapan.	`CodeDeploy_Files` Folder repositori Wilayah GitHub CodePipeline Tidak Didukung menyertakan file sampel yang menginstruksikan CodeDeploy untuk menyebarkan aplikasi. `appspec.yml` File adalah file CodeDeploy konfigurasi yang berisi kait untuk mengontrol aliran penerapan aplikasi. Anda dapat menggunakan file sampel`index.htm`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`l ,, start_ser ver.sh stop_server.sh ,` dan `install_dependenci es.sh` untuk memperbar ui situs web yang di-host di Apache. Ini adalah contoh —Anda dapat menggunak an kode di GitHub repositor i untuk menyebarkan semua jenis aplikasi. Ketika file didorong ke CodeCommit repositori, pipeline tak terlihat dimulai secara otomatis. Untuk hasil penerapan, periksa hasil fase individu di CodeBuild dan CodeDeploy konsol.	

Sumber daya terkait

- Memulai (dokumentasi AWS CDK)
- Pengantar Cloud Development Kit (CDK) (AWS Workshop Studio)
- Lokakarya AWS CDK

Sesuaikan nama peran default dengan menggunakan AWS CDK aspek dan lubang keluar

SANDEEP SINGH dan James Jacob, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menyesuaikan nama default peran yang dibuat oleh AWS Cloud Development Kit (AWS CDK) konstruksi. Menyesuaikan nama peran seringkali diperlukan jika organisasi Anda memiliki batasan khusus berdasarkan konvensi penamaan. Misalnya, organisasi Anda mungkin menetapkan batas izin AWS Identity and Access Management (IAM) atau kebijakan kontrol layanan (SCPs) yang memerlukan awalan tertentu dalam nama peran. Dalam kasus seperti itu, nama peran default yang dihasilkan oleh AWS CDK konstruksi mungkin tidak memenuhi konvensi ini dan mungkin harus diubah. Pola ini membahas persyaratan tersebut dengan menggunakan lubang keluar dan aspek di. AWS CDK Anda menggunakan escape hatch untuk menentukan nama peran kustom, dan aspek untuk menerapkan nama kustom ke semua peran, untuk memastikan kepatuhan terhadap kebijakan dan batasan organisasi Anda.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Prasyarat yang ditentukan dalam dokumentasi AWS CDK

Batasan

- Aspek memfilter sumber daya berdasarkan jenis sumber daya, sehingga semua peran memiliki awalan yang sama. Jika Anda memerlukan awalan peran yang berbeda untuk peran yang berbeda, pemfilteran tambahan berdasarkan properti lain diperlukan. Misalnya, untuk menetapkan awalan berbeda ke peran yang terkait dengan AWS Lambda fungsi, Anda dapat memfilter berdasarkan atribut atau tag peran tertentu, dan menerapkan satu awalan untuk peran terkait Lambda dan awalan yang berbeda untuk peran lain.
- Nama peran IAM memiliki panjang maksimum 64 karakter, jadi nama peran yang dimodifikasi harus dipangkas untuk memenuhi batasan ini.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Tumpukan teknologi target

- AWS CDK
- AWS CloudFormation

Arsitektur target

- AWS CDK Aplikasi terdiri dari satu atau lebih CloudFormation tumpukan, yang disintesis dan digunakan untuk mengelola sumber daya. AWS
- Untuk memodifikasi properti sumber daya yang AWS CDK dikelola yang tidak diekspos oleh konstruksi layer 2 (L2), Anda menggunakan escape hatch untuk mengganti CloudFormation properti yang mendasarinya (dalam hal ini, nama peran), dan aspek untuk menerapkan peran ke semua sumber daya dalam AWS CDK aplikasi selama proses sintesis tumpukan. AWS CDK

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS CDK Command Line Interface (AWS CDK CLI) (juga disebut sebagai AWS CDK Toolkit) adalah kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan aplikasi Anda. AWS CDK `cdk`Perintah CLI adalah alat utama untuk berinteraksi dengan aplikasi Anda. AWS CDK Ini menjalankan aplikasi Anda, menginterogasi model aplikasi yang Anda tentukan, dan menghasilkan serta menerapkan CloudFormation template yang dihasilkan oleh. AWS CDK
- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.

Repositori kode

Kode sumber dan template untuk pola ini tersedia di repositori GitHub CDK Aspects Override.

Praktik terbaik

Lihat Praktik terbaik untuk menggunakan AWS CDK in TypeScript untuk membuat proyek IAc di situs web AWS Prescriptive Guidance.

Epik

Instal AWS CDK CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS CDK CLI.	Untuk menginstal AWS CDK CLI secara global, jalankan perintah: `npm install -g aws-cdk`	AWS DevOps
Verifikasi versinya.	Jalankan perintah : `cdk --version` Konfirmasikan bahwa Anda menggunakan AWS CDK CLI versi 2.	AWS DevOps
Bootstrap AWS CDK lingkungan.	Sebelum Anda menyebark an CloudFormation template, siapkan akun dan Wilayah AWS yang ingin Anda gunakan. Jalankan perintah : `cdk bootstrap <account> /<Region>` Untuk informasi selengkapnya, lihat AWS CDK bootstrap di dokumentasi. AWS	AWS DevOps

Terapkan AWS CDK aplikasi untuk mendemonstrasikan penggunaan aspek

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan proyek.	1. Kloning GitHub repositori untuk pola ini ke komputer lokal Anda:
	`git clone https://github.com/aws-samples/cdk-aspects-override`
	2. Arahkan ke direktori proyek di komputer lokal Anda.
	3. Instal dependensi proyek:
	`npm ci`
		AWS DevOps

Terapkan tumpukan dengan nama peran default yang ditetapkan oleh. AWS CDK	Terapkan dua CloudFormation tumpukan (`ExampleStack1` dan`ExampleStack2`) yang berisi fungsi Lambda dan peran terkaitnya: `npm run deploy:ExampleAppWithoutAspects` Kode tidak secara eksplisit meneruskan properti peran, sehingga nama peran akan dibuat oleh. AWS CDK Misalnya output, lihat bagian Informasi tambahan.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan tumpukan dengan aspek.	Pada langkah ini, Anda menerapkan aspek yang memberlakukan konvensi nama peran dengan menambahkan awalan ke semua peran IAM yang diterapkan dalam proyek. AWS CDK Aspek didefinisikan dalam `lib/aspects.ts` file. Aspek menggunak an escape hatch untuk mengganti nama peran dengan menambahkan awalan. Aspek ini diterapkan pada tumpukan dalam `bin/app-with-aspects.ts` file. Awalan nama peran yang digunakan dalam contoh ini adalah`dev-unicorn` .
	1. Edit file `bin/app-with-aspects.ts`
	2. Dalam file, perbarui variabel `ROLE_NAME_PREFIX` dengan awalan`dev-unicorn` :
	`const app = new cdk.App();`
	`// Define a prefix for the role names`
	`const ROLE_NAME_PREFIX = 'dev-unicorn';`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	``
	// Instantiate the
	 RoleNamingConventi
	onAspect with the
	 desired prefix
	const roleNamin
	gConventionAspect
	 = new RoleNamin
	gConventionAspect(
	ROLE_NAME_PREFIX);
	``
	3. Terapkan AWS CDK aplikasi dengan aspek:
	``
	npm run deploy:Ex
	ampleAppWithAspects
	``
	Misalnya output, lihat bagian Informasi tambahan.	

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus AWS CloudFormation tumpukan Anda.	Setelah Anda selesai menggunakan pola ini, jalankan perintah berikut untuk membersihkan sumber daya agar tidak menimbulkan biaya tambahan: ``` cdk destroy --all -f && cdk --app npx ts-node bin/app-with-aspec ts.ts' destroy --all - f ```	AWS DevOps

Pemecahan masalah

Isu	Solusi
Anda mengalami masalah menggunakan AWS CDK.	Lihat Memecahkan AWS CDK masalah umum dalam dokumentasi. AWS CDK

Sumber daya terkait

- AWS Cloud Development Kit (AWS CDK)
- AWS CDK dokumentasi
- AWS CDK pada GitHub
- Melarikan diri menetas
- Aspek dan AWS CDK

Informasi tambahan

Nama peran yang dibuat oleh CloudFormation tanpa aspek

```
Outputs:
ExampleStack1WithoutAspects.Function1RoleName = example-stack1-without-as-
Function1LambdaFunctionSe-y7FYTY6FXJXA
ExampleStack1WithoutAspects.Function2RoleName = example-stack1-without-as-
Function2LambdaFunctionSe-dDZV4rkWqWnI
...

Outputs:
ExampleStack2WithoutAspects.Function3RoleName = example-stack2-without-as-
Function3LambdaFunctionSe-ygMv49iTyMq0
```

Nama peran yang dibuat oleh CloudFormation dengan aspek

```
Outputs:
ExampleStack1WithAspects.Function1RoleName = dev-unicorn-
Function1LambdaFunctionServiceRole783660DC
ExampleStack1WithAspects.Function2RoleName = dev-unicorn-
Function2LambdaFunctionServiceRole2C391181
...
```

```
Outputs:
ExampleStack2WithAspects.Function3RoleName = dev-unicorn-
Function3LambdaFunctionServiceRole4CAA721C
```

Terapkan cluster Cassandra di Amazon EC2 dengan statis IPs pribadi untuk menghindari penyeimbangan kembali

Dipin Jain, Amazon Web Services

Ringkasan

IP pribadi instans Amazon Elastic Compute Cloud (Amazon EC2) dipertahankan sepanjang siklus hidupnya. Namun, IP pribadi mungkin berubah selama kerusakan sistem yang direncanakan atau tidak direncanakan; misalnya, selama pemutakhiran Amazon Machine Image (AMI). Dalam beberapa skenario, mempertahankan IP statis pribadi dapat meningkatkan kinerja dan waktu pemulihan beban kerja. Misalnya, menggunakan IP statis untuk node benih Apache Cassandra mencegah cluster menimbulkan overhead penyeimbangan kembali.

Pola ini menjelaskan cara melampirkan elastic network interface sekunder ke EC2 instance agar IP tetap statis selama rehosting. Pola ini berfokus pada cluster Cassandra, tetapi Anda dapat menggunakan implementasi ini untuk arsitektur apa pun yang mendapat manfaat dari statis pribadi. IPs

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Service (AWS) yang aktif

Versi produk

- DataStax versi 5.11.1
- Sistem operasi: Ubuntu 16.04.6 LTS

Arsitektur

Arsitektur sumber

Sumbernya bisa berupa cluster Cassandra di mesin virtual lokal (VM) atau pada EC2 instance di AWS Cloud. Diagram berikut menggambarkan skenario kedua. Contoh ini mencakup empat node cluster: tiga node benih dan satu node manajemen. Dalam arsitektur sumber, setiap node memiliki antarmuka jaringan tunggal yang terpasang.

Arsitektur target

Cluster tujuan di-host pada EC2 instance dengan elastic network interface sekunder yang melekat pada setiap node, seperti yang diilustrasikan dalam diagram berikut.

Otomatisasi dan skala

Anda juga dapat mengotomatiskan melampirkan antarmuka jaringan elastis kedua ke grup Auto EC2 Scaling, seperti yang dijelaskan dalam video AWS Knowledge Center.

Epik

Konfigurasikan cluster Cassandra di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan EC2 node untuk meng-host cluster Cassandra.	Di EC2 konsol Amazon, luncurkan empat EC2 instance untuk node Ubuntu Anda di akun AWS Anda. Tiga node (seed) digunakan untuk cluster Cassandra, dan node keempat bertindak sebagai node manajemen cluster di mana Anda akan menginstal DataStax Enterprise (DSE). OpsCenter Untuk petunjuk, lihat EC2 dokumentasi Amazon.	Insinyur awan
Konfirmasikan komunikasi simpul.	Pastikan bahwa keempat node dapat berkomunikasi satu sama lain melalui database dan port manajemen cluster.	Insinyur jaringan
Instal DSE OpsCenter pada node manajemen.	Instal DSE OpsCenter 6.1 dari paket Debian pada node	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	manajemen. Untuk instruksi, lihat DataStax dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat antarmuka jaringan sekunder.	Cassandra menghasilkan pengidentifikasi unik universal (UUID) untuk setiap node berdasarkan alamat IP instance untuk node tersebut EC2 . UUID ini digunakan untuk mendistribusikan node virtual (vnodes) pada ring. Ketika Cassandra digunakan pada EC2 instance, alamat IP ditetapkan secara otomatis ke instance saat dibuat. Jika terjadi pemadaman yang direncanakan atau tidak direncanakan, alamat IP untuk EC2 instance baru berubah, distribusi data berubah, dan seluruh cincin harus diseimbangkan kembali. Ini tidak diinginkan. Untuk mempertahankan alamat IP yang ditetapkan, gunakan elastic network interface sekunder dengan alamat IP tetap. 1. Di EC2 konsol Amazon, pilih Antarmuka Jaringan, Buat antarmuka jaringan. 2. Untuk Subnet, pilih subnet tempat Anda membuat EC2 instance. 3. Untuk IPv4 Alamat pribadi, pilih Tetapkan otomatis.	Insinyur Awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk Grup keamanan, pilih grup keamanan, lalu pilih Buat antarmuka jaringan. Untuk informasi selengkapnya tentang membuat antarmuka jaringan, lihat EC2 dokumentasi Amazon.	
Pasang antarmuka jaringan sekunder ke node cluster.	1. Di EC2 konsol Amazon, pilih Instans. 2. Pilih kotak centang untuk EC2 contoh yang Anda buat sebelumnya. 3. Pilih Tindakan, Jaringan, Lampirkan antarmuka jaringan. 4. Pilih antarmuka jaringan yang Anda buat pada langkah sebelumnya, lalu pilih Lampirkan. Untuk informasi selengkap nya tentang melampirkan antarmuka jaringan, lihat EC2 dokumentasi Amazon.	Insinyur awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan rute di Amazon EC2 untuk mengatasi perutean asimetris.	Saat Anda memasang antarmuka jaringan kedua, jaringan kemungkinan besar akan melakukan routing asimetris. Untuk menghindari hal ini, Anda dapat menambahkan rute untuk antarmuka jaringan baru. Untuk penjelasan mendalam dan remediasi perutean asimetris, lihat video AWS Knowledge Center atau Mengatasi Perutean Asimetris di Server Multi-Rumah (artikel di Jurnal Linux oleh Patrick, 5 April 2004). McManus	Insinyur jaringan
Perbarui entri DNS untuk menunjuk ke IP antarmuka jaringan sekunder.	Arahkan nama domain yang sepenuhnya memenuhi syarat (FQDN) dari node ke IP antarmuka jaringan sekunder.	Insinyur jaringan
Instal dan konfigurasikan cluster Cassandra dengan menggunakan DSE. OpsCenter	Ketika node cluster siap dengan antarmuka jaringan sekunder, Anda dapat menginstal dan mengkonfi gurasi cluster Cassandra.	DBA

Pulihkan cluster dari kegagalan node

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AMI untuk node benih cluster.	Buat cadangan node sehingga Anda dapat mengembalikannya dengan binari database jika terjadi kegagalan node. Untuk petunjuknya, lihat Membuat AMI di EC2 dokumentasi Amazon.	Administrator cadangan
Pulihkan dari kegagalan node.	Ganti node yang gagal dengan EC2 instance baru yang diluncurkan dari AMI, dan lampirkan antarmuka jaringan sekunder dari node yang gagal.	Administrator cadangan
Verifikasi bahwa cluster Cassandra sehat.	Saat node pengganti sudah habis, verifikasi kesehatan cluster di DSE OpsCenter.	DBA

Sumber daya terkait

- Menginstal DSE OpsCenter 6.1 dari paket Debian (dokumentasi) DataStax
- Cara membuat antarmuka jaringan sekunder berfungsi dalam EC2 instance Ubuntu (video AWS Knowledge Center)
- Praktik Terbaik untuk Menjalankan Apache Cassandra di Amazon (postingan blog EC2 AWS)

Memperluas VRFs ke AWS dengan menggunakan AWS Transit Gateway Connect

Adam Till, Yashar Araghi, Vikas Dewangan, dan Mohideen, Amazon Web Services HajaMohideen

Ringkasan

Virtual routing and forwarding (VRF) adalah fitur jaringan tradisional. Ini menggunakan domain routing logis terisolasi, dalam bentuk tabel rute, untuk memisahkan lalu lintas jaringan dalam infrastruktur fisik yang sama. Anda dapat mengonfigurasi AWS Transit Gateway untuk mendukung isolasi VRF saat menghubungkan jaringan lokal ke AWS. Pola ini menggunakan arsitektur sampel untuk menghubungkan lokal VRFs ke tabel rute gateway transit yang berbeda.

Pola ini menggunakan transit virtual interface (VIFs) di AWS Direct Connect dan lampiran transit gateway Connect untuk memperluas. VRFs VIF transit digunakan untuk mengakses satu atau lebih gateway transit VPC Amazon yang terkait dengan gateway Direct Connect. Lampiran Connect gateway transit menghubungkan gateway transit dengan alat virtual pihak ketiga yang berjalan di VPC. Lampiran transit gateway Connect mendukung protokol terowongan Generic Routing Encapsulation (GRE) untuk kinerja tinggi, dan mendukung Border Gateway Protocol (BGP) untuk perutean dinamis.

Pendekatan yang dijelaskan dalam pola ini memiliki manfaat sebagai berikut:

- Dengan menggunakan Transit Gateway Connect, Anda dapat mengiklankan hingga 1.000 rute ke peer Transit Gateway Connect dan menerima hingga 5.000 rute darinya. Menggunakan fitur VIF transit Direct Connect tanpa Transit Gateway Connect dibatasi hingga 20 awalan per gateway transit.
- Anda dapat mempertahankan isolasi lalu lintas dan menggunakan Transit Gateway Connect untuk menyediakan layanan yang dihosting di AWS, terlepas dari skema alamat IP yang digunakan pelanggan Anda.
- Lalu lintas VRF tidak perlu melintasi antarmuka virtual publik. Ini membuatnya lebih mudah untuk mematuhi persyaratan kepatuhan dan keamanan di banyak organisasi.
- Setiap terowongan GRE mendukung hingga 5 Gbps, dan Anda dapat memiliki hingga empat terowongan GRE per lampiran Connect gateway transit. Ini lebih cepat daripada banyak jenis koneksi lainnya, seperti koneksi AWS Site-to-Site VPN yang mendukung hingga 1,25 Gbps.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang diperlukan telah dibuat (lihat arsitektur untuk detailnya)
- Izin untuk mengambil peran AWS Identity and Access Management (IAM) di setiap akun.
- Peran IAM di setiap akun harus memiliki izin untuk menyediakan sumber daya AWS Transit Gateway dan AWS Direct Connect. Untuk informasi selengkapnya, lihat Otentikasi dan kontrol akses untuk gateway transit Anda dan lihat Manajemen identitas dan akses untuk Direct Connect.
- Koneksi Direct Connect telah berhasil dibuat. Untuk informasi selengkapnya, lihat Membuat sambungan menggunakan wizard Koneksi.

Batasan

- Ada batasan untuk lampiran gateway transit ke akun VPCs produksi, QA, dan pengembangan. Untuk informasi selengkapnya, lihat Lampiran gateway transit ke VPC.
- Ada batasan untuk membuat dan menggunakan gateway Direct Connect. Untuk informasi selengkapnya, lihat kuota AWS Direct Connect.

Arsitektur

Arsitektur target

Contoh arsitektur berikut menyediakan solusi yang dapat digunakan kembali untuk menerapkan transit VIFs dengan lampiran Connect gateway transit. Arsitektur ini memberikan ketahanan dengan menggunakan beberapa lokasi Direct Connect. Untuk informasi selengkapnya, lihat Ketahanan maksimum dalam dokumentasi Direct Connect. Jaringan lokal memiliki produksi, QA, dan pengembangan VRFs yang diperluas ke AWS dan diisolasi dengan menggunakan tabel rute khusus.

Di lingkungan AWS, dua akun didedikasikan untuk memperluas VRFs: akun Direct Connect dan akun hub jaringan. Akun Direct Connect berisi koneksi dan transit VIFs untuk setiap router. Anda membuat transit VIFs dari akun Direct Connect tetapi menyebarkannya ke akun hub jaringan sehingga Anda dapat mengaitkannya dengan gateway Direct Connect di akun hub jaringan. Akun hub jaringan berisi gateway Direct Connect dan gateway transit. Sumber daya AWS terhubung sebagai berikut:

1. Transit VIFs menghubungkan router di lokasi Direct Connect dengan AWS Direct Connect di akun Direct Connect.

2. VIF transit menghubungkan Direct Connect dengan gateway Direct Connect di akun hub jaringan.
3. Asosiasi gateway transit menghubungkan gateway Direct Connect dengan gateway transit di akun hub jaringan.
4. Lampiran transit gateway Connect menghubungkan gateway transit dengan VPCs akun produksi, QA, dan pengembangan.

Arsitektur Transit VIF

Diagram berikut menunjukkan detail konfigurasi untuk transit VIFs. Contoh arsitektur ini menggunakan VLAN untuk sumber terowongan, tetapi Anda juga bisa menggunakan loopback.

Berikut ini adalah detail konfigurasi, seperti nomor sistem otonom (ASNs), untuk transit VIFs.

Sumber daya	Item	Detail
router-01	ASN	65534
router-02	ASN	65534
router-03	ASN	65534
router-04	ASN	65534
Gateway Direct Connect	ASN	64601
Gateway transit	ASN	64600
	Blok CIDR	10.100.254.0/24

Transit gateway Connect arsitektur

Diagram dan tabel berikut menjelaskan cara mengkonfigurasi VRF tunggal melalui lampiran Connect gateway transit. Untuk tambahan VRFs, tetapkan terowongan unik IDs, alamat IP GRE gateway transit, dan BGP di dalam blok CIDR. Alamat IP GRE peer cocok dengan alamat IP peer router dari VIF transit.

Tabel berikut berisi rincian konfigurasi router.

Router	Terowongan	Alamat IP	Sumber	Destinasi
router-01	Terowongan 1	169.254.101.17	VLAN 60 169.254.100.1	10.100.254.1
router-02	Terowongan 11	169.254.101.81	VLAN 61 169.254.100.5	10.100.254.11
router-03	Terowongan 21	169.254.101.145	VLAN 62 169.254.100.9	10.100.254.21
router-04	Terowongan 31	169.254.101.209	VLAN 63 169.254.100.13	10.100.254.31

Tabel berikut berisi rincian konfigurasi gateway transit.

Terowongan	Alamat IP GRE gateway transit	Alamat IP Peer GRE	BGP di dalam blok CIDR
Terowongan 1	10.100.254.1	VLAN 60 169.254.100.1	169.254.101.16/29
Terowongan 11	10.100.254.11	VLAN 61 169.254.100.5	169.254.101.80/29
Terowongan 21	10.100.254.21	VLAN 62 169.254.100.9	169.254.101.144/29
Terowongan 31	10.100.254.31	VLAN 63 169.254.100.13	169.254.101.208/29

Deployment

Bagian Epics menjelaskan cara menerapkan konfigurasi sampel untuk satu VRF di beberapa router pelanggan. Setelah langkah 1—5 selesai, Anda dapat membuat lampiran Connect gateway transit baru dengan menggunakan langkah 6-7 untuk setiap VRF baru yang Anda perluas ke AWS:

1. Buat gateway transit.
2. Buat tabel rute Transit Gateway untuk setiap VRF.
3. Buat antarmuka virtual transit.
4. Buat gateway Direct Connect.
5. Buat antarmuka virtual gateway Direct Connect dan asosiasi gateway dengan awalan yang diizinkan.
6. Buat lampiran Connect gateway transit.
7. Buat peer Transit Gateway Connect.
8. Kaitkan lampiran Connect gateway transit dengan tabel rute.
9. Iklankan rute ke router.

Alat

Layanan AWS

- AWS Direct Connect menautkan jaringan internal Anda ke lokasi Direct Connect melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke layanan AWS publik sambil melewati penyedia layanan internet di jalur jaringan Anda.
- AWS Transit Gateway adalah hub pusat yang menghubungkan cloud pribadi virtual (VPCs) dan jaringan lokal.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Epik

Rencanakan arsitekturnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat diagram arsitektur kustom.	1. Di bagian Lampiran, unduh templat diagram. 2. Buka diagram terlampir di Microsoft Office PowerPoint. 3. Pada slide ikhtisar Arsitektur, sesuaikan diagram arsitektur untuk lingkungan Anda. Identifikasi lokal VRFs yang perlu diperluas ke lingkungan AWS Anda. 4. Pada slide Transit VIF, sesuaikan diagram arsitektur. Identifikasi nomor AS dari router, gateway Direct Connect, dan gateway transit. Identifikasi alamat IP di setiap ujung VIF transit. 5. Pada slide Transit Gateway Connect, sesuaikan diagram arsitektur untuk setiap VRF. Identifikasi semua alamat IP yang diperlukan untuk mengkonfigurasi router dan rekan Transit Gateway Connect.	Arsitek cloud, Administrator jaringan

Buat sumber daya Transit Gateway

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gateway transit.	1. Masuk ke akun hub jaringan. 2. Ikuti petunjuk di Buat gateway transit. Perhatikan hal berikut untuk pola ini: • Untuk Amazon side Autonomous System Number (ASN), masukkan ASN unik. Untuk keperluan contoh ini, ASN adalah`64600`. • Pilih dukungan DNS. • Untuk contoh arsitektur ini, dukungan VPN ECMP, asosiasi tabel rute default, prorogasi tabel rute default, dan dukungan Multicast tidak diperlukan. • Untuk blok CIDR gateway Transit, masukkan blok IPv4 CIDR untuk gateway transit Anda. Untuk keperluan contoh ini, blok CIDR adalah`10.100.254.0/24` .	Administrator jaringan, arsitek Cloud
Buat tabel rute gateway transit.	Ikuti petunjuk di Buat tabel rute gateway transit. Perhatikan hal berikut untuk pola ini:	Arsitek cloud, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk tag Nama, berikan nama untuk tabel rute gateway transit. Sebaiknya gunakan nama yang sesuai dengan VRF, seperti. `routetable-dev-vrf` • Untuk ID gateway Transit, pilih gateway transit yang Anda buat sebelumnya.	

Buat antarmuka virtual transit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat antarmuka virtual transit.	1. Masuk ke akun Direct Connect. 2. Ikuti petunjuk di Buat antarmuka virtual transit ke gateway Direct Connect. Perhatikan hal berikut untuk pola ini: • Untuk nama antarmuka Virtual, masukkan nama untuk VIF transit. Sebaiknya gunakan nama yang sesuai dengan router, seperti`transit-vif-router01` . • Untuk Koneksi, pilih router, seperti`router-01` .	Arsitek cloud, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Untuk pemilik antarmuka Virtual, masukkan ID akun akun hub jaringan. Untuk petunjuk, lihat Melihat ID akun AWS Anda.
	• Untuk gateway Direct Connect, jangan membuat pilihan apa pun. Anda melampirkan gateway Direct Connect pada langkah berikutnya.
	• Untuk VLAN, masukkan VLAN router, seperti. `60`
	• Untuk BGP ASN, masukkan ASN router, seperti. `65534`
	• Di bawah Pengaturan Tambahan, lakukan hal berikut:
	  • Pilih IPv4.
	  • Untuk IP peer router Anda, masukkan alamat IP peer router, seperti. `169.254.100.1`
	  • Untuk IP peer router Amazon. masukkan IP peer router Amazon, seperti. `169.254.100.2`
	  • Untuk kunci otentikasi BGP, diperlukan kata	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sandi. Jika ini dibiarkan kosong, AWS membuat kunci yang hanya dapat diakses di akun ini. 3. Ulangi instruksi ini untuk membuat semua transit VIFs untuk VRF.	

Buat sumber daya Direct Connect

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gateway Direct Connect.	1. Masuk ke akun hub jaringan. 2. Ikuti petunjuk di Membuat gateway Direct Connect. Perhatikan hal berikut untuk pola ini: • Untuk ASN sisi Amazon, masukkan ASN gateway Direct Connect, seperti. `64601` • Jangan memilih gateway pribadi virtual.	Arsitek cloud, Administrator jaringan
Pasang gateway Direct Connect ke transit VIFs.	1. Di akun hub jaringan, buka konsol AWS Direct Connect di https://console.aws.amazon.com/directconnect/v2/. 2. Di panel navigasi, pilih Antarmuka Jaringan.	Arsitek cloud, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pilih VIF transit baru, lalu pilih Terima. 4. Pilih gateway Direct Connect yang Anda buat. 5. Ulangi instruksi ini untuk setiap transit VIF.	
Buat asosiasi gateway Direct Connect dengan awalan yang diizinkan.	Di akun hub jaringan, ikuti petunjuk di Untuk mengaitkan gateway transit. Perhatikan hal berikut untuk pola ini: • Untuk Gateway, pilih gateway transit yang Anda buat sebelumnya. • Untuk awalan Diizinkan, masukkan blok CIDR yang ditetapkan ke gateway transit, seperti. **`10.100.254.0/24`** Membuat asosiasi ini secara otomatis membuat lampiran Transit Gateway yang memiliki tipe sumber daya Direct Connect Gateway. Lampiran ini tidak perlu dikaitkan dengan tabel rute gateway transit.	Arsitek cloud, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat lampiran Connect gateway transit.	1. Di akun hub jaringan, buka konsol VPC Amazon di. https://console.aws.amazon.com/vpc/ 2. Di panel navigasi, pilih Lampiran gateway transit. 3. Pilih Buat lampiran gateway transit. 4. Untuk tag Nama, masukkan nama untuk lampiran. Sebaiknya gunakan nama yang sesuai dengan VRF, seperti. `PROD-VRF` 5. Untuk ID gateway Transit, pilih gateway transit yang Anda buat sebelumnya. 6. Untuk jenis Lampiran, pilih Connect. 7. Untuk ID lampiran Transport, pilih gateway Direct Connect yang Anda buat sebelumnya. 8. Pilih Buat lampiran gateway transit. 9. Ulangi langkah ini untuk setiap VRF yang Anda perluas.	Arsitek cloud, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peer Transit Gateway Connect.	1. Di akun hub jaringan, ikuti petunjuk di Buat peer Transit Gateway Connect (terowongan GRE). Perhatikan hal berikut untuk pola ini: • Untuk tag Nama, masukkan nama untuk rekan Transit Gateway Connect. Kami merekomendasikan menggunakan nama yang sesuai dengan router, seperti `connectpeer-router01` . • Untuk alamat GRE gateway Transit, masukkan alamat IP yang ditetapkan dari blok CIDR gateway transit, seperti. `10.100.254.1` • Untuk alamat Peer GRE, masukkan alamat IP yang ditetapkan ke VLAN yang dibuat pada router untuk VIF transit, seperti. `169.254.100.1` Asalkan AWS dapat mencapai alamat IP, Anda dapat menggunakan antarmuka apa pun, seperti VLAN atau	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Loopback, untuk alamat GRE rekan. • Untuk BGP Inside CIDR Blocks (IPv4), masukkan BGP di dalam alamat IP blok CIDR, seperti. `169.254.101.16/29` • Untuk Peer ASN, masukkan ASN router, seperti. `65534` 2. Ulangi instruksi ini untuk membuat terowongan GRE untuk setiap router.	

Iklankan rute ke router

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Iklankan rute.	Kaitkan lampiran Connect gateway transit baru dengan tabel rute yang Anda buat sebelumnya untuk VRF ini. Misalnya, kaitkan lampiran Connect gateway transit produksi dengan tabel `Production-VRF` rute. Buat rute statis untuk awalan yang diiklankan ke router. 1. Masuk ke akun hub jaringan.	Administrator jaringan, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buka konsol Amazon VPC di. https://console.aws.amazon.com/vpc/ 3. Di panel navigasi, di bawah Transit Gateways, pilih Tabel rute gateway transit. 4. Pilih tabel `Production-VRF` rute. 5. Pada menu Tindakan, pilih Buat rute statis. 6. Untuk CIDR, masukkan blok CIDR untuk rute yang diiklankan ke lampiran gateway transit di VPC target, seperti. `10.100.1.0/24` 7. Untuk Pilih Lampiran, pilih lampiran Connect gateway transit yang relevan. 8. Pilih Buat rute statis.	

Sumber daya terkait

Dokumentasi AWS

- Dokumentasi Direct Connect
  - Bekerja dengan gateway Direct Connect
  - Asosiasi gerbang transit
  - Antarmuka virtual AWS Direct Connect
- Dokumentasi Transit Gateway
  - Bekerja dengan gateway transit
  - Lampiran gateway transit ke gateway Direct Connect

- Lampiran Transit gateway Connect dan rekan Transit Gateway Connect
- Membuat lampiran Connect gateway transit

Postingan blog AWS

- Segmentasi jaringan hybrid dengan AWS Transit Gateway connect
- Menggunakan AWS Transit Gateway terhubung untuk memperluas VRFs dan meningkatkan iklan awalan IP

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Dapatkan notifikasi Amazon SNS saat status kunci kunci AWS KMS berubah

Shubham Harsora, Aromal Raj Jayarajan, dan Navdeep Pareek, Amazon Web Services

Ringkasan

Data dan metadata yang terkait dengan kunci AWS Key Management Service (AWS KMS) hilang saat kunci tersebut dihapus. Penghapusan tidak dapat diubah dan Anda tidak dapat memulihkan data yang hilang (termasuk data terenkripsi). Anda dapat mencegah kehilangan data dengan menyiapkan sistem notifikasi untuk memberi tahu Anda tentang perubahan status pada status kunci AWS KMS Anda.

Pola ini menunjukkan kepada Anda cara memantau perubahan status ke kunci AWS KMS dengan menggunakan Amazon dan EventBridge Amazon Simple Notification Service (Amazon SNS) untuk mengeluarkan notifikasi otomatis setiap kali status kunci AWS KMS berubah menjadi atau. `Disabled` `PendingDeletion` Misalnya, jika pengguna mencoba menonaktifkan atau menghapus kunci AWS KMS, Anda akan menerima pemberitahuan email dengan detail tentang upaya perubahan status. Anda juga dapat menggunakan pola ini untuk menjadwalkan penghapusan kunci AWS KMS.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan pengguna AWS Identity and Access Management (IAM)
- Kunci AWS KMS

Arsitektur

Tumpukan teknologi

- Amazon EventBridge
- AWS Key Management Service (AWS KMS)
- Amazon Simple Notification Service (Amazon SNS)

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk membangun proses pemantauan dan pemberitahuan otomatis untuk mendeteksi perubahan apa pun pada status kunci AWS KMS.

Diagram menunjukkan alur kerja berikut:

1. Pengguna menonaktifkan atau menjadwalkan penghapusan kunci AWS KMS.
2. EventBridge Aturan mengevaluasi jadwal `Disabled` atau `PendingDeletion` acara.
3. EventBridge Aturan tersebut memanggil topik Amazon SNS.
4. Amazon SNS mengirimkan pesan notifikasi email kepada pengguna.

> **Note**
>
> Anda dapat menyesuaikan pesan email untuk memenuhi kebutuhan organisasi Anda. Sebaiknya sertakan informasi tentang entitas tempat kunci AWS KMS digunakan. Ini dapat membantu pengguna memahami dampak penghapusan kunci AWS KMS. Anda juga dapat menjadwalkan pemberitahuan email pengingat yang dikirim satu atau dua hari sebelum kunci AWS KMS dihapus.

Otomatisasi dan skala

CloudFormation Tumpukan AWS menyebarkan semua sumber daya dan layanan yang diperlukan agar pola ini berfungsi. Anda dapat menerapkan pola secara independen dalam satu akun, atau dengan menggunakan AWS CloudFormation StackSets untuk beberapa akun independen atau unit organisasi di AWS Organizations.

Alat

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun AWS dan Wilayah AWS. CloudFormation Template untuk pola ini menjelaskan semua sumber daya AWS yang Anda inginkan, serta menyediakan CloudFormation serta mengonfigurasi sumber daya tersebut untuk Anda.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. EventBridge mengirimkan aliran data real-time dari aplikasi dan layanan AWS Anda sendiri, dan merutekan

data tersebut ke target seperti AWS Lambda. EventBridge menyederhanakan proses membangun arsitektur berbasis peristiwa.

- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

Kode

Kode untuk pola ini tersedia di kunci GitHub Monitor AWS KMS nonaktifkan dan repositori penghapusan terjadwal.

Epik

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Kloning kunci GitHub Monitor AWS KMS menonaktifkan dan menjadwalkan repositori penghapusan ke mesin lokal Anda dengan menjalankan perintah berikut: `git clone https://github.com/aws-samples/aws-kms-deletion-notification`	Administrator AWS, arsitek Cloud
Perbarui parameter template.	Di editor kode, buka `Alerting-KMS-Events.yaml` CloudFormation template yang Anda kloning dari repositori, lalu perbarui parameter berikut:	Administrator AWS, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk `DestinationEmailAddress`, masukkan alamat email aktif yang Anda rencanakan untuk digunakan untuk menerima pemberitahuan SNS. • Untuk `SNSTopicName`, masukkan nama untuk topik SNS Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	1. Masuk ke AWS Management Console dan buka CloudFormation konsol. 2. Di panel navigasi, pilih Buat tumpukan, lalu pilih Dengan sumber daya baru (standar). 3. Pada halaman Identifikasi sumber daya, pilih Berikutnya. 4. Pada halaman Tentukan templat, untuk Sumber templat, pilih Unggah file templat. 5. Pilih Pilih file, pilih **`Alerting-KMS-Events.yaml`** file dari GitHub repositori kloning Anda, lalu pilih Berikutnya. 6. Untuk nama Stack, masukkan nama tumpukan Anda. 7. Pilih Kirim.	Administrator AWS, arsitek Cloud

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan email berlangganan.	Setelah CloudFormation template berhasil digunakan, Amazon SNS mengirimkan pesan konfirmasi berlangga	Administrator AWS, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nan ke alamat email yang Anda berikan di CloudForm ation template. Untuk menerima pemberita huan, Anda harus mengonfir masi langganan email ini. Untuk informasi selengkapnya, lihat Konfirmasi langganan di Panduan Pengembang Amazon SNS.	

Uji pemberitahuan berlangganan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Nonaktifkan kunci AWS KMS.	1. Masuk ke AWS Management Console dan buka konsol AWS KMS. 2. Untuk mengubah Wilayah, pilih nama Wilayah yang ditampilkan saat ini, lalu pilih Wilayah yang ingin Anda alihkan. 3. Di panel navigasi, pilih Kunci yang dikelola pelanggan. 4. Pilih kotak centang untuk kunci AWS KMS yang ingin Anda aktifkan atau nonaktifkan.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Untuk menonaktifkan kunci AWS KMS, pilih Tindakan kunci, lalu pilih Nonaktifkan.	
Validasi langganan.	Konfirmasikan bahwa Anda menerima email notifikasi Amazon SNS.	Administrator AWS

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus CloudFormation tumpukan.	1. Masuk ke AWS Management Console dan buka CloudFormation konsol. 2. Di panel navigasi, pilih Stacks (Tumpukan). 3. Pilih tumpukan yang Anda buat sebelumnya, lalu pilih Hapus.	Administrator AWS

Sumber daya terkait

- AWS CloudFormation (dokumentasi AWS)
- Membuat tumpukan di CloudFormation konsol AWS (CloudFormation dokumentasi AWS)
- Membangun arsitektur berbasis peristiwa di AWS (dokumentasi AWS Workshop Studio)
- Praktik Terbaik AWS Key Management Service (AWS Whitepaper)
- Praktik terbaik keamanan untuk AWS Key Management Service (Panduan Pengembang AWS KMS)

Informasi tambahan

Amazon SNS menyediakan enkripsi dalam transit secara default. Untuk menyelaraskan dengan praktik terbaik keamanan, Anda juga dapat mengaktifkan enkripsi sisi server untuk Amazon SNS dengan menggunakan kunci terkelola pelanggan AWS KMS.

Pertahankan ruang IP yang dapat dirutekan dalam desain VPC multi-akun untuk subnet non-beban kerja

Adam Spicer, Amazon Web Services

Ringkasan

Amazon Web Services (AWS) telah menerbitkan praktik terbaik yang merekomendasikan penggunaan subnet khusus di cloud pribadi virtual (VPC) untuk lampiran gateway transit dan titik akhir Load Balancer Gateway (untuk mendukung AWS Network Firewall atau peralatan pihak ketiga). Subnet ini digunakan untuk memuat antarmuka jaringan elastis untuk layanan ini. Jika Anda menggunakan AWS Transit Gateway dan Load Balancer Gateway, dua subnet dibuat di setiap Availability Zone untuk VPC. Karena VPCs caranya dirancang, subnet tambahan ini tidak boleh lebih kecil dari mask/28 dan dapat mengkonsumsi ruang IP routable yang berharga yang dapat digunakan untuk beban kerja yang dapat dirutekan. Pola ini menunjukkan bagaimana Anda dapat menggunakan rentang Classless Inter-Domain Routing (CIDR) sekunder yang tidak dapat dirutekan untuk subnet khusus ini untuk membantu mempertahankan ruang IP yang dapat dirutekan.

Prasyarat dan batasan

Prasyarat

- Strategi multi-VPC untuk ruang IP yang dapat dirutekan
- Rentang CIDR yang tidak dapat dirutekan untuk layanan yang Anda gunakan (lampiran gateway transit dan Load Balancer Gatewayatau titik akhir Network Firewall)

Arsitektur

Arsitektur target

Pola ini mencakup dua arsitektur referensi: satu arsitektur memiliki subnet untuk lampiran transit gateway (TGW) dan titik akhir Gateway Load Balancer (GWLBe), dan arsitektur kedua hanya memiliki subnet untuk lampiran TGW.

Arsitektur 1 - VPC terpasang TGW dengan perutean masuk ke alat

Diagram berikut mewakili arsitektur referensi untuk VPC yang mencakup dua Availability Zones. Saat masuk, VPC menggunakan pola perutean ingress untuk mengarahkan lalu lintas yang ditujukan

untuk subnet publik ke alat untuk inspeksi firewall. bump-in-the-wire Lampiran TGW mendukung jalan keluar dari subnet pribadi ke VPC terpisah.

Pola ini menggunakan rentang CIDR yang tidak dapat dirutekan untuk subnet lampiran TGW dan subnet. GWLBe Dalam tabel routing TGW, CIDR yang tidak dapat dirutekan ini dikonfigurasi dengan rute blackhole (statis) dengan menggunakan serangkaian rute yang lebih spesifik. Jika rute disebarkan ke tabel perutean TGW, rute lubang hitam yang lebih spesifik ini akan berlaku.

Dalam contoh ini, /23 routable CIDR dibagi dan sepenuhnya dialokasikan ke subnet routable.

Arsitektur 2 — VPC terlampir TGW

Diagram berikut mewakili arsitektur referensi lain untuk VPC yang mencakup dua Availability Zones. Lampiran TGW mendukung lalu lintas keluar (jalan keluar) dari subnet pribadi ke VPC terpisah. Ini menggunakan rentang CIDR yang tidak dapat dirutekan hanya untuk subnet lampiran TGW. Dalam tabel routing TGW, CIDR yang tidak dapat dirutekan ini dikonfigurasi dengan rute blackhole dengan menggunakan serangkaian rute yang lebih spesifik. Jika rute disebarkan ke tabel perutean TGW, rute lubang hitam yang lebih spesifik ini akan berlaku.

Dalam contoh ini, /23 routable CIDR dibagi dan sepenuhnya dialokasikan ke subnet routable.

Alat

Layanan dan sumber daya AWS

- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan. Dalam pola ini, VPC sekunder digunakan untuk mempertahankan ruang IP CIDRs yang dapat dirutekan dalam beban kerja. CIDRs
- Perutean masuknya gateway Internet (asosiasi tepi) dapat digunakan bersama dengan titik akhir Gateway Load Balancer untuk subnet khusus yang tidak dapat dirutekan.
- AWS Transit Gateway adalah hub pusat yang menghubungkan VPCs dan jaringan lokal. Dalam pola ini, terpasang VPCs secara terpusat ke gateway transit, dan lampiran gateway transit berada dalam subnet khusus yang tidak dapat dirutekan.
- Gateway Load Balancers membantu Anda menerapkan, menskalakan, dan mengelola peralatan virtual, seperti firewall, sistem deteksi dan pencegahan intrusi, dan sistem inspeksi paket

mendalam. Gateway berfungsi sebagai titik masuk dan keluar tunggal untuk semua lalu lintas. Dalam pola ini, titik akhir untuk Load Balancer Gateway dapat digunakan dalam subnet khusus yang tidak dapat dirutekan.

- AWS Network Firewall adalah firewall jaringan yang stateful, dikelola, dan layanan deteksi dan pencegahan intrusi untuk AWS Cloud VPCs . Dalam pola ini, titik akhir untuk firewall dapat digunakan dalam subnet non-routable khusus.

Repositori kode

Runbook dan CloudFormation templat AWS untuk pola ini tersedia di repositori Pola CIDR Sekunder GitHub Non-Routable. Anda dapat menggunakan file sampel untuk menyiapkan lab kerja di lingkungan Anda.

Praktik terbaik

AWS Transit Gateway

- Gunakan subnet terpisah untuk setiap lampiran VPC gateway transit.
- Alokasikan subnet /28 dari rentang CIDR non-routable sekunder untuk subnet lampiran gateway transit.
- Di setiap tabel routing gateway transit, tambahkan rute statis yang lebih spesifik untuk rentang CIDR yang tidak dapat dirutekan sebagai lubang hitam.

Load Balancer Gateway dan perutean ingress

- Gunakan perutean ingress untuk mengarahkan lalu lintas dari internet ke titik akhir Load Balancer Gateway.
- Gunakan subnet terpisah untuk setiap titik akhir Load Balancer Gateway.
- Alokasikan subnet /28 dari rentang CIDR sekunder yang tidak dapat dirutekan untuk subnet titik akhir Gateway Load Balancer.

Epik

Buat VPCs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan rentang CIDR yang tidak dapat dirutekan.	Tentukan rentang CIDR yang tidak dapat dirutekan yang akan digunakan untuk subnet lampiran gateway transit dan (opsional) untuk setiap subnet titik akhir Gateway Load Balancer atau Network Firewall. Rentang CIDR ini akan digunakan sebagai CIDR sekunder untuk VPC. Itu tidak boleh dirutekan dari rentang CIDR utama VPC atau jaringan yang lebih besar.	Arsitek awan
Tentukan rentang CIDR yang dapat dirutekan untuk. VPCs	Tentukan satu set rentang CIDR yang dapat dirutekan yang akan digunakan untuk Anda. VPCs Rentang CIDR ini akan digunakan sebagai CIDR utama untuk Anda. VPCs	Arsitek awan
Buat VPCs.	Buat Anda VPCs dan lampirkan ke gateway transit. Setiap VPC harus memiliki rentang CIDR primer yang dapat dirutekan dan rentang CIDR sekunder yang tidak dapat dirutekan, berdasarkan rentang yang Anda tentukan dalam dua langkah sebelumnya.	Arsitek awan

Konfigurasikan rute lubang hitam Transit Gateway

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat non-routable yang lebih spesifik CIDRs sebagai lubang hitam.	Setiap tabel routing gateway transit harus memiliki satu set rute blackhole yang dibuat untuk non-routable. CIDRs Ini dikonfigurasi untuk memastikan bahwa setiap lalu lintas dari CIDR VPC sekunder tetap tidak dapat dirutekan dan tidak bocor ke jaringan yang lebih besar. Rute ini harus lebih spesifik daripada CIDR yang tidak dapat dirutekan yang ditetapkan sebagai CIDR sekunder pada VPC. Misalnya, jika CIDR non-routable sekunder adalah 100.64.0.0/26, rute blackhole di tabel routing gateway transit harus 100.64.0.0/27 dan 100.64.0.32/27.	Arsitek awan

Sumber daya terkait

- Praktik terbaik untuk men-deploy Penyeimbang Beban Gateway
- Arsitektur Inspeksi Terdistribusi dengan Load Balancer Gateway
- Hari Perendaman Jaringan - Internet ke Lab Firewall VPC
- Praktik terbaik desain gateway transit

Informasi tambahan

Rentang CIDR sekunder yang tidak dapat dirutekan juga dapat berguna saat bekerja dengan penerapan kontainer berskala lebih besar yang memerlukan sekumpulan besar alamat IP. Anda dapat menggunakan pola ini dengan Gateway NAT pribadi untuk menggunakan subnet yang tidak dapat dirutekan untuk meng-host penerapan kontainer Anda. Untuk informasi lebih lanjut, lihat posting blog Cara mengatasi kelelahan IP Pribadi dengan Solusi NAT Pribadi.

Menyediakan produk Terraform AWS Service Catalog dengan menggunakan repositori kode

Dr. Rahul Sharad Gaikwad dan Tamilselvan P, Amazon Web Services

Ringkasan

AWS Service Catalog mendukung penyediaan layanan mandiri dengan tata kelola untuk konfigurasi Terraform Anda. HashiCorp Jika Anda menggunakan Terraform, Anda dapat menggunakan Service Catalog sebagai alat tunggal untuk mengatur, mengatur, dan mendistribusikan konfigurasi Terraform Anda dalam skala besar. AWS Anda dapat mengakses fitur utama Service Catalog, termasuk membuat katalog templat infrastruktur standar dan pra-persetujuan sebagai kode (IAc), kontrol akses, penyediaan sumber daya cloud dengan akses hak istimewa paling sedikit, pembuatan versi, berbagi hingga ribuan, dan penandaan. Akun AWS Pengguna akhir, seperti insinyur, administrator database, dan ilmuwan data, melihat daftar produk dan versi yang dapat mereka akses, dan mereka dapat menerapkannya melalui satu tindakan.

Pola ini membantu Anda menerapkan AWS sumber daya dengan menggunakan kode Terraform. Kode Terraform di GitHub repositori diakses melalui Service Catalog. Dengan menggunakan pendekatan ini, Anda mengintegrasikan produk dengan alur kerja Terraform yang ada. Administrator dapat membuat portofolio Service Catalog dan menambahkan AWS Launch Wizard produk ke dalamnya dengan menggunakan Terraform.

Berikut ini adalah manfaat dari solusi ini:

- Karena fitur rollback di Service Catalog, jika terjadi masalah selama penerapan, Anda dapat mengembalikan produk ke versi sebelumnya.
- Anda dapat dengan mudah mengidentifikasi perbedaan antara versi produk. Ini membantu Anda menyelesaikan masalah selama penerapan.
- Anda dapat mengonfigurasi koneksi repositori di Service Catalog, seperti ke GitHub atau. GitLab Anda dapat membuat perubahan produk secara langsung melalui repositori.

Untuk informasi tentang manfaat keseluruhan AWS Service Catalog, lihat Apa itu Service Catalog.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.

- A GitHub, BitBucket, atau repositori lain yang berisi file konfigurasi Terraform dalam format ZIP.
- AWS Serverless Application Model Antarmuka Baris Perintah (AWS SAM CLI), diinstal.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi.
- Pergi, diinstal.
- Python versi 3.9, diinstal. AWS SAM CLI membutuhkan versi Python ini.
- Izin untuk menulis dan menjalankan AWS Lambda fungsi dan izin untuk mengakses dan mengelola produk dan portofolio Service Catalog.

Arsitektur

Diagram menunjukkan alur kerja berikut:

1. Saat konfigurasi Terraform siap, pengembang membuat file.zip yang berisi semua kode Terraform. Pengembang mengunggah file.zip ke dalam repositori kode yang terhubung ke Service Catalog.
2. Administrator mengaitkan produk Terraform ke portofolio di Service Catalog. Administrator juga membuat batasan peluncuran yang memungkinkan pengguna akhir untuk menyediakan produk.
3. Di Service Catalog, pengguna akhir meluncurkan AWS resource dengan menggunakan konfigurasi Terraform. Mereka dapat memilih versi produk mana yang akan digunakan.

Alat

Layanan AWS

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Service Catalogmembantu Anda mengelola katalog layanan TI secara terpusat yang disetujui. AWS Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.

Layanan lainnya

- Go adalah bahasa pemrograman open source yang didukung Google.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Jika Anda memerlukan contoh konfigurasi Terraform yang dapat Anda terapkan melalui Service Catalog, Anda dapat menggunakan konfigurasi di Amazon Macie GitHub Organization Setup Menggunakan repositori Terraform. Penggunaan sampel kode dalam repositori ini tidak diperlukan.

Praktik terbaik

- Alih-alih memberikan nilai untuk variabel dalam file konfigurasi Terraform (`terraform.tfvars`), konfigurasikan nilai variabel saat meluncurkan produk melalui Service Catalog.
- Berikan akses ke portofolio hanya untuk pengguna atau administrator tertentu.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Memberikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi AWS Identity and Access Management (IAM).

Epik

Siapkan workstation lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Instal Docker.	Jika Anda ingin menjalankan AWS Lambda fungsi di lingkungan pengembangan Anda, instal Docker. Untuk petunjuk, lihat Menginstal Docker Engine di dokumentasi Docker.	DevOps insinyur
Instal AWS Service Catalog Mesin untuk Terraform.	1. Masukkan perintah berikut untuk mengkloning repositori AWS Service Catalog Engine for Terraform. `git clone https://github.com/aws-samples/service-catalo`	DevOps insinyur, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`g-engine-for-terra form-os.git`
	2. Arahkan ke direktori root dari repositori kloning.
	3. Masukkan perintah berikut. Ini menginstal mesin.
	`run ./bin/bash/ deploy-tre.sh -r`
	Wilayah AWS Set di profil default Anda tidak digunakan selama instalasi otomatis. Sebagai gantinya, Anda memberikan Region saat Anda menjalankan perintah ini.	

Connect GitHub repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat koneksi ke GitHub repositori.	1. Masuk ke Konsol Manajemen AWS, lalu buka konsol Alat Pengemban g. Anda dapat mengakses konsol Alat Pengembang dengan memilih layanan seperti AWS CodePipeline atau AWS CodeDeploy. 2. Di panel navigasi kiri, pilih Pengaturan, lalu pilih Koneksi. 3. Pilih Buat koneksi.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pilih repositori tempat Anda memelihara kode sumber Terraform. Misalnya, Anda dapat memilih Bitbucket , GitHub, atau GitHub Enterprise Server. 5. Masukkan nama untuk koneksi, lalu pilih Connect. 6. Ketika Anda diminta, otentikasi repositori. Setelah otentikasi selesai, koneksi dibuat dan status berubah menjadi aktif.	

Buat produk Terraform di Service Catalog

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat produk Service Catalog.	1. Buka konsol AWS Service Catalog. 2. Arahkan ke bagian Administrasi, lalu pilih Daftar produk. 3. Pilih Buat produk. 4. Pada halaman Buat produk di bagian Detail produk, pilih Jenis produk eksternal. Service Catalog menggunakan jenis produk ini untuk mendukung produk Terraform Community Edition.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Masukkan nama dan pemilik untuk produk Service Catalog. 6. Pilih Tentukan repositori kode Anda menggunakan penyedia. CodeStar 7. Masukkan informasi berikut untuk repositori Anda: • Connect ke provider Anda menggunakan AWS CodeConnections — Pilih koneksi yang Anda buat sebelumnya. • Repositori — Pilih repositori. • Cabang — Pilih cabang. • Jalur file template - Pilih jalur tempat file template kode disimpan. Nama file harus diakhiri dengan`tar.gz`. 8. Di bawah Nama dan deskripsi Versi, berikan informasi tentang versi produk. 9. Pilih Buat produk.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat portofolio.	1. Buka konsol AWS Service Catalog. 2. Arahkan ke bagian Administrasi, lalu pilih, pilih Portofolio. 3. Pilih Buat portofolio. 4. Masukkan nilai berikut: • Nama portofolio – `Sample terraform` • Deskripsi portofolio - `Sample portfolio for Terraform configurations` • Pemilik — Informasi kontak Anda, seperti alamat email 5. Pilih Buat.	Administrator AWS
Tambahkan produk Terraform ke portofolio.	1. Buka konsol AWS Service Catalog. 2. Arahkan ke bagian Administrasi, lalu pilih Daftar produk. 3. Pilih produk Terraform yang Anda buat sebelumnya. 4. Pilih Tindakan, lalu pilih Tambahkan produk ke portofolio. 5. Pilih `Sample terraform` portofolio. 6. Pilih Tambahkan produk ke portofolio.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan akses.	1. Buka konsol AWS Identity and Access Management (IAM). 2. Di panel navigasi, pilih Kebijakan. 3. Di panel konten, pilih Buat kebijakan. 4. Pilih opsi JSON. 5. Masukkan contoh kebijakan JSON di Kebijakan akses di bagian Informasi tambahan dari pola ini. 6. Pilih Berikutnya. 7. Pada halaman Tinjau dan buat, di kotak Nama kebijakan, masukkan`TerraformResourceCreationAndArtifactAccessPolicy` . 8. Pilih Buat kebijakan.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan kepercayaan khusus.	1. Buka konsol IAM. 2. Di panel navigasi, pilih Peran. 3. Pilih Buat peran. 4. Di bawah Jenis entitas tepercaya, pilih Kebijakan kepercayaan khusus. 5. Di editor kebijakan JSON, masukkan contoh kebijakan JSON di Kebijakan kepercayaan di bagian Informasi tambahan dari pola ini. 6. Pilih Berikutnya. 7. Di bawah Kebijakan Izin, pilih `TerraformResourceCreationAndArtifactAccessPolicy` yang sebelumnya Anda buat. 8. Pilih Berikutnya. 9. Di bawah Rincian peran, di kotak Nama peran, masukkan `SCLaunch-product` . **Important** Nama peran harus dimulai dengan `SCLaunch`. 10. Pilih Buat peran.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan batasan peluncuran ke produk Service Catalog.	1. Masuk ke Konsol Manajemen AWS sebagai pengguna dengan izin administratif. 2. Buka konsol AWS Service Catalog. 3. Di panel navigasi, pilih Portofolio. 4. Pilih portofolio yang Anda buat sebelumnya. 5. Pada halaman Detail portofolio, pilih tab Constraints, lalu pilih Create constraint. 6. Untuk Produk, pilih produk Terraform yang Anda buat sebelumnya. 7. Di bawah Batasan peluncuran, untuk Metode, pilih Masukkan nama peran. 8. Di kotak Nama peran, masukkan`SCLaunch-product` . 9. Pilih Buat.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan akses ke produk.	1. Buka konsol AWS Service Catalog. 2. Di panel navigasi, pilih Portofolio. 3. Pilih portofolio yang Anda buat sebelumnya. 4. Pilih tab Access, lalu pilih Grant access. 5. Pilih tab Peran, lalu pilih peran yang seharusny a memiliki akses untuk menyebarkan produk ini. 6. Pilih Akses Hibah.	Administrator AWS
Luncurkan produk.	1. Masuk ke pengguna Konsol Manajemen AWS sebagai pengguna dengan izin untuk menyebarkan produk Service Catalog. 2. Buka konsol AWS Service Catalog. 3. Di panel navigasi, pilih Produk. 4. Pilih produk yang Anda buat sebelumnya, lalu pilih Luncurkan produk. 5. Masukkan nama produk dan tentukan parameter yang diperlukan. 6. Pilih Luncurkan produk.	DevOps insinyur

Verifikasi penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan.	Ada dua mesin AWS Step Functions status untuk alur kerja penyediaan Service Catalog: • `ManageProvisionedProductStateMachine` — Service Catalog memanggil mesin status ini saat menyediakan produk Terraform baru dan saat memperbarui produk yang disediakan Terraform yang ada. • `TerminateProvisionedProductStateMachine` — Service Catalog memanggil mesin status ini saat menghentikan produk yang disediakan Terraform yang ada. Anda memeriksa log untuk mesin `ManageProvisionedProductStateMachine` status untuk mengonfirmasi bahwa produk telah disediakan. 1. Masuk ke Konsol Manajemen AWS, lalu buka AWS Step Functions konsol.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di panel navigasi kiri, pilih mesin Negara. 3. Pilih`ManagePro visionedProductSta teMachine` . 4. Dalam daftar Eksekusi, masukkan ID produk yang disediakan untuk menemukan eksekusi. **Note** Nama bucket backend file status dimulai dengan. `sc-terraform-engine-state-` 5. Validasi bahwa semua sumber daya yang diperlukan telah dibuat di akun.	

Membersihkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus produk yang disediaka n.	1. Masuk ke pengguna Konsol Manajemen AWS sebagai pengguna dengan izin untuk menyebarkan produk Service Catalog. 2. Buka konsol AWS Service Catalog.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di navigasi kiri, pilih Produk yang disediakan. 4. Pilih produk yang Anda buat. 5. Dalam daftar Tindakan, pilih Hentikan. 6. Di kotak teks konfirmasi, masukkan`terminate` , lalu pilih Hentikan produk yang disediakan. 7. Ulangi langkah-langkah ini untuk menghentikan semua produk yang disediakan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lepaskan AWS Service Catalog Mesin untuk Terraform.	1. Masuk ke Konsol Manajemen AWS sebagai pengguna dengan izin administratif. 2. Buka konsol Amazon Simple Storage Service (Amazon S3). 3. Di panel navigasi, pilih Bucket. 4. Pilih `sc-terraform-engine-logging-XXXX` ember. 5. Pilih Kosong. 6. Ulangi langkah 4—5 untuk ember berikut: • `sc-terraform-engine-state-XXXX` • `terraform-engine-bootstrap-XXXX` 7. Buka AWS CloudFormation konsol, lalu validasi Anda benar Wilayah AWS. 8. Di navigasi kiri, pilih Stacks. 9. Pilih`SAM-TRE`, lalu pilih Hapus. Tunggu sampai tumpukan telah dihapus. 10. Pilih`Bootstrap-TRE` , lalu pilih Hapus. Tunggu sampai tumpukan telah dihapus.	Administrator AWS

Sumber daya terkait

AWS dokumentasi

- Memulai dengan produk Terraform

Dokumentasi Terraform

- Instalasi Terraform
- Konfigurasi backend Terraform
- Dokumentasi Penyedia Terraform AWS

Informasi tambahan

Kebijakan akses

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
                }
            }
        },
        {
            "Action": [
                "s3:CreateBucket*",
                "s3:DeleteBucket*",
                "s3:Get*",
                "s3:List*",
                "s3:PutBucketTagging"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
        },
```

```
        {
            "Action": [
                "resource-groups:CreateGroup",
                "resource-groups:ListGroupResources",
                "resource-groups:DeleteGroup",
                "resource-groups:Tag"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "tag:GetResources",
                "tag:GetTagKeys",
                "tag:GetTagValues",
                "tag:TagResources",
                "tag:UntagResources"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Kebijakan kepercayaan

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GivePermissionsToServiceCatalog",
            "Effect": "Allow",
            "Principal": {
                "Service": "servicecatalog.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_id:root"
            },
            "Action": "sts:AssumeRole",
```

```
            "Condition": {
                "StringLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
TerraformExecutionRole*",
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogExternalParameterParserRole*",
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogTerraformOSParameterParserRole*"
                    ]
                }
            }
        }
    ]
}
```

Daftarkan beberapa Akun AWS dengan satu alamat email dengan menggunakan Amazon SES

Joe Wozniak dan Shubhangi Vishwakarma, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana Anda dapat memisahkan alamat email asli dari alamat email yang terkait dengan alamat email. Akun AWS Akun AWS memerlukan alamat email unik yang akan diberikan pada saat pembuatan akun. Di beberapa organisasi, tim yang mengelola Akun AWS harus menanggung beban mengelola banyak alamat email unik dengan tim pesan mereka. Ini bisa sulit bagi organisasi besar yang mengelola banyak orang Akun AWS. Selain itu, jika sistem email Anda tidak mengizinkan pengalamatan plus atau sub-alamat seperti yang didefinisikan dalam Penyaringan Email Saringan: Ekstensi Subaddress (RFC 5233) —dengan menambahkan tanda plus (+) dan pengenal di akhir bagian lokal alamat email, seperti —pola ini dapat membantu mengatasi batasan ini. `admin+123456789123@example.com`

Pola ini memberikan solusi penjual alamat email unik yang memungkinkan Akun AWS pemilik untuk mengaitkan satu alamat email dengan beberapa Akun AWS alamat. Alamat email sebenarnya dari Akun AWS pemilik kemudian dikaitkan dengan alamat email yang dihasilkan ini dalam sebuah tabel. Solusi ini menangani semua email masuk untuk akun email unik, mencari pemilik setiap akun, dan kemudian meneruskan pesan yang diterima ke pemilik.

Prasyarat dan batasan

Prasyarat

- Akses administratif ke sebuah Akun AWS.
- Akses ke lingkungan pengembangan.
- (Opsional) Keakraban dengan AWS Cloud Development Kit (AWS CDK) alur kerja dan bahasa pemrograman Python akan membantu Anda memecahkan masalah atau membuat modifikasi.

Batasan

- Panjang alamat email yang dijual secara keseluruhan 64 karakter. Untuk detailnya, lihat CreateAccountdi referensi AWS Organizations API.

Versi produk

- Node.js versi 22.x atau yang lebih baru
- Python 3.13 atau yang lebih baru
- Paket Python pip dan virtualenv
- AWS CDK CLI versi 2.1019.2 atau yang lebih baru
- Docker 20.10.x atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- CloudFormation tumpukan
- AWS Lambda fungsi
- Aturan dan aturan Amazon Simple Email Service (Amazon SES) dan aturan
- AWS Identity and Access Management (IAM) peran dan kebijakan
- Kebijakan bucket dan bucket Amazon Simple Storage Service (Amazon S3)
- AWS Key Management Service (AWS KMS) kebijakan kunci dan kunci
- Kebijakan topik dan topik Amazon Simple Notification Service (Amazon SNS)
- Tabel Amazon DynamoDB

Arsitektur target

Diagram ini menunjukkan dua aliran:

- Alur penjual alamat email: Dalam diagram, alur penjual alamat email (bagian bawah) biasanya dimulai dengan solusi penjual akun atau otomatisasi luar, atau dipanggil secara manual. Dalam permintaan, fungsi Lambda dipanggil dengan payload yang berisi metadata yang dibutuhkan. Fungsi ini menggunakan informasi ini untuk menghasilkan nama akun dan alamat email yang unik, menyimpannya dalam database DynamoDB, dan mengembalikan nilai ke pemanggil. Nilai-nilai ini kemudian dapat digunakan untuk membuat baru Akun AWS (biasanya dengan menggunakan AWS Organizations).
- Alur penerusan email: Alur ini diilustrasikan di bagian atas diagram sebelumnya. Ketika Akun AWS dibuat dengan menggunakan email akun yang dihasilkan dari alur penjual alamat email, AWS mengirim berbagai email, seperti konfirmasi pendaftaran akun dan pemberitahuan berkala, ke alamat email tersebut. Dengan mengikuti langkah-langkah dalam pola ini, Anda mengonfigurasi

Anda Akun AWS dengan Amazon SES untuk menerima email untuk seluruh domain. Solusi ini mengonfigurasi aturan penerusan yang memungkinkan Lambda memproses semua email yang masuk, periksa untuk melihat apakah alamatnya T0 ada di tabel DynamoDB, dan meneruskan pesan ke alamat email pemilik akun sebagai gantinya. Menggunakan proses ini memberi pemilik akun kemampuan untuk mengaitkan beberapa akun dengan satu alamat email.

Otomatisasi dan skala

Pola ini menggunakan AWS CDK untuk sepenuhnya mengotomatiskan penerapan. Solusinya menggunakan layanan AWS terkelola yang akan (atau dapat dikonfigurasi untuk) menskalakan secara otomatis untuk memenuhi kebutuhan Anda. Fungsi Lambda mungkin memerlukan konfigurasi tambahan untuk memenuhi kebutuhan penskalaan Anda. Untuk informasi selengkapnya, lihat Memahami penskalaan fungsi Lambda dalam dokumentasi Lambda.

Alat

Layanan AWS

- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Email Service (Amazon SES) membantu Anda mengirim dan menerima email dengan menggunakan alamat email dan domain Anda sendiri.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat yang dibutuhkan untuk penyebaran

- Lingkungan pengembangan dengan AWS CLI dan akses IAM ke Anda Akun AWS. Untuk detailnya, lihat tautan di bagian Sumber daya terkait.
- Pada sistem pengembangan Anda, instal yang berikut ini:
  - Alat baris perintah Git, tersedia dari situs unduhan Git.
  - AWS CLI Untuk mengkonfigurasi kredenal akses untuk file. AWS CDK Lihat informasi yang lebih lengkap dalam dokumentasi AWS CLI.
  - Python versi 3.13 atau yang lebih baru, tersedia dari situs web unduhan Python.
  - UV untuk manajemen paket Python. Untuk petunjuk pemasangan, lihat panduan pemasangan UV.
  - Node.js versi 22.x atau yang lebih baru. Untuk petunjuk penginstalan, lihat dokumentasi Node.js.
  - AWS CDK CLI versi 2.1019.2 atau yang lebih baru. Untuk petunjuk instalasi, lihat AWS CDK dokumentasi.
  - Docker versi 20.10.x atau yang lebih baru. Untuk petunjuk penginstalan, lihat dokumentasi Docker.

Kode

Kode untuk pola ini tersedia di repositori email GitHub Akun AWS pabrik.

Epik

Mengalokasikan lingkungan penyebaran target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi atau buat Akun AWS.	Identifikasi yang sudah ada atau baru Akun AWS di mana Anda memiliki akses administratif penuh, untuk menyebarkan solusi email.	Administrator AWS, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan lingkungan penerapan.	Konfigurasikan lingkungan penerapan yang mudah digunakan dan atur dependensi dengan mengikuti langkah-langkah berikut: 1. Siapkan lingkungan pengembangan Anda dengan alat yang tercantum di bagian Alat. 2. Kloning basis kode repositori email GitHub Akun AWS pabrik ke lingkungan pengembangan Anda dengan menggunakan perintah: `git clone https://github.com/aws-samples/aws-account-factory-email` 3. Dalam `requirements.in` file (di root repositori), perbarui baris yang dimulai dengan `aws-cdk-lib==` agar sesuai dengan versi terbaru dari file. AWS CDK Untuk mengidentifikasi apa versi terbaru, jalankan `npm view aws-cdk-lib` dan cari kata kunci “latest:”. 4. Jika Anda membuat pembaruan ke	AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	versi`requirements.in` , jalankan perintah berikut untuk memperbarui `requirements.txt` file dengan menggunakan UV: `uv pip compile requirements.in -o requirements.txt --python-version 3.13`	

Siapkan domain terverifikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengidentifikasi dan mengalokasikan domain.	Fungsionalitas penerusan email membutuhkan domain khusus. Identifikasi dan alokasikan domain atau subdomain yang dapat Anda verifikasi dengan Amazon SES. Domain ini harus tersedia untuk menerima email masuk di Akun AWS tempat solusi penerusan email digunakan. Persyaratan domain: • Domain harus menjadi domain standar atau subdomain. • Domain harus dapat diselesaikan secara eksternal DNS, karena akan	Administrator cloud, Administrator jaringan, administrator DNS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	digunakan untuk menerima email dari luar organisasi.	
Verifikasi domain.	Verifikasi bahwa domain yang diidentifikasi dapat digunakan untuk menerima email masuk. Lengkapi petunjuk di Memverifikasi domain Anda untuk penerimaan email Amazon SES di dokumenta si Amazon SES. Ini akan membutuhkan koordinasi dengan orang atau tim yang bertanggung jawab atas catatan DNS domain.	Pengembang aplikasi, AWS DevOps
Siapkan catatan MX.	Siapkan domain Anda dengan data MX yang mengarah ke titik akhir Amazon SES di Wilayah Akun AWS dan Anda. Untuk informasi selengkap nya, lihat Menerbitkan data MX untuk penerimaan email Amazon SES di dokumentasi Amazon SES.	Administrator cloud, Administr ator jaringan, administrator DNS

Menyebarkan solusi penjual otomatis dan penerusan email

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah nilai default dicdk.json.	Edit beberapa nilai default dalam `cdk.json` file (di root repositori) sehingga solusi	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akan beroperasi dengan benar setelah digunakan. 1. Ubah `SES_DOMAIN_NAME` nilainya agar sesuai dengan nama domain yang Anda verifikasi sebelumnya. 2. Ubah `ADDRESS_FROM` nilai untuk menyertakan domain yang sama yang ada di`SES_DOMAIN_NAME` . Bagian lokal dari alamat harus ditentukan oleh tim cloud Anda. Alamat ini menjadi `FROM` alamat untuk setiap email yang diteruskan melalui solusi. 3. Ubah `ADDRESS_ADMIN` nilai agar sesuai dengan alamat email yang akan diteruskan pesan masuk yang tidak cocok. Nilai ini harus berupa alamat email yang valid dan beroperasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan solusi penjual dan penerusan email.	1. Buat lingkungan virtual Python:
	`uv venv --python 3.13`
	2. Aktifkan lingkungan virtual Python:
	`source .venv/bin/activate`
	Atau, pada platform Windows, gunakan:
	`% .venv\Scripts\activate.bat`
	3. Instal semua persyaratan Python tanpa kesalahan:
	`uv pip install -r requirements.txt`
	4. Sintesis CloudFormation template:
	`cdk synth`
	Konfirmasikan bahwa tidak ada kesalahan dan bahwa CloudFormation template lengkap berisi output yang diharapkan.
	5. (Opsional) Jika Anda menerapkan AWS CDK kode ke dalam saat ini
		Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Akun AWS atau Wilayah untuk pertama kalinya, bootstrap lingkungan. Untuk informasi selengkapnya, lihat AWS CDK Bootstrap ping di dokumentasi. AWS CDK
	`cdk bootstrap aws://AWS-ACCOUNT-NUMBER/REGION`
	Ganti `AWS-ACCOUNT-NUMBER` dan `REGION` dengan nilai aktual.
	6. Terapkan solusinya:
	`cdk bootstrap`
	`cdk deploy`
	Perintah harus selesai tanpa kesalahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa solusi telah diterapkan.	Verifikasi bahwa solusi berhasil diterapkan sebelum Anda mulai menguji: 1. Buka CloudFormation konsol dan cari CloudFormation tumpukan yang berisi nama `AwsMailFwdStack` . 2. Konfirmasikan bahwa `AwsMailFwdStack` tumpukan ini memiliki sumber daya berikut: • Fungsi Lambda • Aturan Amazon SES dan set aturan • Peran dan kebijakan IAM • Kebijakan bucket dan bucket Amazon S3 • AWS KMS kebijakan kunci dan kunci • Kebijakan topik dan topik Amazon SNS • Tabel DynamoDB	Pengembang aplikasi, AWS DevOps

Verifikasi bahwa penjual dan penerusan email beroperasi seperti yang diharapkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa API berfungsi .	Pada langkah ini, Anda mengirimkan data pengujian ke API solusi dan mengonfirmasi bahwa solusi menghasilkan output yang diharapka	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n dan operasi backend telah dilakukan seperti yang diharapkan. Jalankan fungsi Vend Email Lambda secara manual dengan menggunakan input pengujian. (Sebagai contoh, lihat file sample_vend_request.json.) Untuk `OwnerAddress`, gunakan alamat email yang valid. API harus mengembalikan nama akun dan email akun dengan nilai seperti yang diharapkan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa email sedang diteruskan.	Pada langkah ini, Anda mengirim email pengujian melalui sistem dan memverifikasi bahwa email diteruskan ke penerima yang diharapkan. 1. Dapatkan email akun dari langkah terakhir. 2. Kirim email ke alamat ini dengan subjek tes dan teks isi. 3. Konfirmasikan bahwa Anda menerima email di alamat email pemilik akun. 4. Konfirmasikan bahwa email yang Anda terima memiliki `FROM` alamat yang cocok dengan `ADDRESS_FROM` pengaturan`cdk.json`. 5. Konfirmasikan bahwa subjek dan isi email yang diterima sama dengan pesan terkirim asli.	Pengembang aplikasi, AWS DevOps

Pemecahan masalah

Isu	Solusi
Sistem tidak meneruskan email seperti yang diharapkan.	Verifikasi bahwa pengaturan Anda sudah benar: 1. Anda seharusnya telah menyelesaikan proses verifikasi Amazon SES untuk domain Anda.

Isu	Solusi
	2. Domain Anda harus disiapkan dengan benar dengan catatan MX yang mengarah ke titik akhir Amazon SES di Wilayah dan Anda Akun AWS . Untuk informasi selengkapnya, lihat Menerbitkan data MX untuk penerimaan email Amazon SES di dokumentasi Amazon SES. Setelah memverifikasi pengaturan domain, ikuti langkah-langkah berikut: 1. Buka CloudWatch konsol Amazon untuk akun dan Wilayah tempat Anda menerapkan solusi, dan navigasikan ke grup CloudWatch log di panel navigasi. 2. Cari daftar grup log untuk`SesMailForwardLogGroup` . 3. Selidiki log dalam grup ini untuk melihat apakah ada kesalahan yang dihasilkan selama proses penjual dan penerusan email.

Isu	Solusi
Saat Anda mencoba menerapkan AWS CDK tumpukan, Anda menerima kesalahan yang mirip dengan: “Kesalahan format templat: Jenis sumber daya yang tidak dikenal”	Dalam kebanyakan kasus, pesan kesalahan ini berarti bahwa Wilayah yang Anda targetkan tidak memiliki semua layanan AWS yang tersedia. Jika Anda menggunakan EC2 instans Amazon untuk menerapkan solusi, Anda mungkin menargetkan Wilayah yang berbeda dari Wilayah tempat instance berjalan. **Note** Secara default, AWS CDK penyebaran ke Wilayah dan akun yang Anda konfigurasikan di. AWS CLI Solusi yang mungkin: 1. Selidiki apakah semua layanan yang diperlukan untuk solusi ini (lihat bagian tumpukan teknologi Target sebelumnya dalam pola ini) ada dalam penargetan yang Wilayah AWS Anda targetkan dengan meninjau AWS Services by Region. 2. Jika Anda menggunakan EC2 instance dan menargetkan Wilayah yang berbeda dari Wilayah tempat instance Anda berjalan, pastikan untuk menyetel variabel `AWS_DEFAULT_REGION` lingkungan, atau setel Region dengan AWS CLI sebelum Anda menerapkan solusi. Untuk informasi selengkapnya, lihat Mengkonfigurasi variabel lingkungan untuk AWS CLI dokumentasi. AWS CLIAtau, Anda dapat memodifikasi `app.py` file di root repositori untuk menyertakan ID akun dan Wilayah dengan

Isu	Solusi
	kode keras dengan mengikuti instruksi dalam dokumentasi untuk lingkungan.AWS CDK
Saat Anda menerapkan solusi, Anda menerima pesan galat: “Penerapan gagal: Kesalahan:: Parameter SSM bootstrap/hnb659fds/version /cdk-tidak ditemukan. AwsMailFwdStack Apakah lingkungan telah di-bootstrap? Silakan jalankan 'cdk bootstrap'”	Jika Anda belum pernah menerapkan AWS CDK sumber daya apa pun ke Akun AWS dan Wilayah yang Anda targetkan, Anda harus terlebih dahulu menjalankan `cdk bootstrap` perintah seperti yang ditunjukkan oleh kesalahan. Jika Anda terus menerima kesalahan ini setelah menjalankan perintah bootstrap, Anda mungkin mencoba menerapkan solusi ke Wilayah yang berbeda dari Wilayah tempat lingkungan pengembangan Anda berjalan. Untuk mengatasi masalah ini, atur variabel `AWS_DEFAULT_REGION` lingkungan atau atur Region dengan AWS CLI sebelum Anda menerapkan solusi. Atau, Anda dapat memodifikasi `app.py` file di root repositori untuk menyertakan ID akun dan Wilayah dengan kode keras dengan mengikuti instruksi dalam dokumentasi untuk lingkungan.AWS CDK

Sumber daya terkait

- Untuk bantuan menginstal AWS CLI, lihat Menginstal atau memperbarui ke versi terbaru AWS CLI.
- Untuk bantuan menyiapkan kredensil akses IAM AWS CLI dengan IAM, lihat Mengonfigurasi setelan untuk. AWS CLI
- Untuk bantuan dengan AWS CDK, lihat Memulai dengan AWS CDK.

Informasi tambahan

Biaya

Ketika Anda menerapkan solusi ini, Akun AWS pemegang mungkin dikenakan biaya yang terkait dengan penggunaan layanan berikut. Penting bagi Anda untuk memahami bagaimana layanan ini ditagih sehingga Anda mengetahui adanya potensi biaya. Untuk informasi harga, lihat halaman berikut:

- Harga Amazon SES
- Harga Amazon S3
- AWS KMS harga
- AWS Lambda harga
- Harga Amazon DynamoDB

Menyiapkan resolusi DNS untuk jaringan hybrid di lingkungan AWS akun tunggal

Abdullahi Olaoye, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menyiapkan arsitektur Sistem Nama Domain (DNS) hybrid sepenuhnya yang memungkinkan resolusi end-to-end DNS dari sumber daya lokal, sumber daya AWS, dan kueri DNS internet, tanpa overhead administratif. Pola ini menjelaskan cara mengatur aturan penerusan Amazon Route 53 Resolver yang menentukan di mana kueri DNS yang berasal dari AWS harus dikirim, berdasarkan nama domain. Kueri DNS untuk sumber daya lokal diteruskan ke resolver DNS lokal. Kueri DNS untuk sumber daya AWS dan kueri DNS internet diselesaikan oleh Route 53 Resolver.

Pola ini mencakup resolusi DNS hybrid dalam lingkungan akun tunggal AWS. Untuk informasi tentang menyiapkan kueri DNS keluar di lingkungan multi-akun AWS, lihat pola Menyiapkan resolusi DNS untuk jaringan hybrid di lingkungan AWS multi-akun.

Prasyarat dan batasan

Prasyarat

- Akun AWS
- Virtual Private Cloud (VPC) di akun AWS
- Koneksi jaringan antara lingkungan lokal dan VPC Anda, melalui AWS Virtual Private Network (AWS VPN) atau AWS Direct Connect
- Alamat IP resolver DNS lokal Anda (dapat dijangkau dari VPC Anda)
- Nama domain/subdomain untuk diteruskan ke resolver lokal (misalnya, onprem.mydc.com)
- Nama domain/subdomain untuk zona host pribadi AWS (misalnya, myvpc.cloud.com)

Arsitektur

Tumpukan teknologi target

- Amazon Route 53 zona yang dihosting pribadi
- Amazon Route 53 Resolver

- Amazon VPC
- AWS VPN atau Direct Connect

Arsitektur target

Alat

- Amazon Route 53 Resolver membuat cloud hybrid lebih mudah bagi pelanggan perusahaan dengan memungkinkan resolusi kueri DNS yang mulus di seluruh cloud hybrid Anda. Anda dapat membuat titik akhir DNS dan aturan penerusan bersyarat untuk menyelesaikan ruang nama DNS antara pusat data lokal dan pusat data Anda. VPCs
- Amazon Route 53 private hosted zone adalah wadah yang menyimpan informasi tentang bagaimana Anda ingin Route 53 menanggapi kueri DNS untuk domain dan subdomainnya dalam satu atau lebih VPCs yang Anda buat dengan layanan Amazon VPC.

Epik

Konfigurasikan zona yang dihosting pribadi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat zona host pribadi Route 53 untuk nama domain yang dicadangkan AWS seperti myvpc.cloud.com.	Zona ini menyimpan catatan DNS untuk sumber daya AWS yang harus diselesai kan dari lingkungan lokal. Untuk petunjuknya, lihat Membuat zona host pribadi di dokumentasi Route 53.	Admin jaringan, Admin sistem
Kaitkan zona host pribadi dengan VPC Anda.	Untuk mengaktifkan sumber daya di VPC Anda untuk menyelesaikan catatan DNS di zona host pribadi ini, Anda harus mengaitkan VPC Anda dengan zona yang dihosting	Admin jaringan, Admin sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	. Untuk petunjuknya, lihat Membuat zona host pribadi di dokumentasi Route 53.	

Siapkan titik akhir Route 53 Resolver

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir masuk.	Route 53 Resolver menggunakan titik akhir masuk untuk menerima kueri DNS dari resolver DNS lokal. Untuk petunjuk, lihat Meneruskan kueri DNS masuk ke dokumentasi VPCs Route 53 Anda. Catat alamat IP titik akhir masuk.	Admin jaringan, Admin sistem
Buat titik akhir keluar.	Route 53 Resolver menggunakan titik akhir keluar untuk mengirim kueri DNS ke resolver DNS lokal. Untuk petunjuk, lihat Meneruskan kueri DNS keluar ke jaringan Anda di dokumentasi Route 53. Catat ID titik akhir keluaran.	Admin jaringan, Admin sistem

Siapkan aturan penerusan dan kaitkan dengan VPC Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan penerusan untuk domain lokal.	Aturan ini akan menginstruksikan Route 53 Resolver	Admin jaringan, Admin sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk meneruskan kueri DNS apa pun untuk domain lokal (seperti onprem.mydc.com) ke resolver DNS lokal. Untuk membuat aturan ini, Anda memerlukan alamat IP dari resolver DNS lokal dan ID titik akhir keluar untuk Route 53 Resolver. Untuk petunjuk, lihat Mengelola aturan penerusan dalam dokumentasi Route 53.	
Kaitkan aturan penerusan dengan VPC Anda.	Agar aturan penerusan berlaku, Anda harus mengaitkan aturan tersebut dengan VPC Anda. Route 53 Resolver kemudian mempertimbangkan aturan saat menyelesaikan domain. Untuk petunjuk, lihat Mengelola aturan penerusan dalam dokumentasi Route 53.	Admin jaringan, Admin sistem

Konfigurasikan resolver DNS lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan penerusan bersyarat di resolver DNS on-premise.	Agar kueri DNS dikirim ke zona host pribadi Route 53 dari lingkungan lokal, Anda harus mengonfig urasi penerusan bersyarat di resolver DNS lokal. Ini menginstruksikan resolver DNS untuk meneruskan	Admin jaringan, Admin sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	semua kueri DNS untuk domain AWS (misalnya, untuk myvpc.cloud.com) ke alamat IP titik akhir masuk untuk Route 53 Resolver.	

Uji end-to-end resolusi DNS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji resolusi DNS dari AWS ke lingkungan lokal.	Dari server di VPC, lakukan kueri DNS untuk domain lokal (seperti server1.onprem.mydc.com).	Admin jaringan, Admin sistem
Uji resolusi DNS dari lingkungan lokal ke AWS.	Dari server lokal, lakukan resolusi DNS untuk domain AWS (seperti server1.myvpc.cloud.com).	Admin jaringan, Admin sistem

Sumber daya terkait

- Manajemen DNS terpusat dari cloud hybrid dengan Amazon Route 53 dan AWS Transit Gateway (AWS Networking & Content Delivery blog)
- Sederhanakan manajemen DNS di lingkungan multi-akun dengan Route 53 Resolver (blog AWS Security)
- Bekerja dengan zona host pribadi (dokumentasi Route 53)
- Memulai dengan Route 53 Resolver (dokumentasi Route 53)

Siapkan bot UiPath RPA secara otomatis di Amazon EC2 dengan menggunakan AWS CloudFormation

Dr. Rahul Sharad Gaikwad dan Tamilselvan P, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana Anda dapat menerapkan robot robot otomatisasi proses (RPA) di instans Amazon Elastic Compute Cloud (EC2Amazon). Ini menggunakan pipeline EC2 Image Builder untuk membuat Amazon Machine Image (AMI) kustom. AMI adalah gambar mesin virtual (VM) yang telah dikonfigurasi sebelumnya yang berisi sistem operasi (OS) dan perangkat lunak yang sudah diinstal sebelumnya untuk menyebarkan instance. EC2 Pola ini menggunakan CloudFormation templat AWS untuk menginstal edisi UiPath Studio Community pada AMI kustom. UiPath adalah alat RPA yang membantu Anda mengatur robot untuk mengotomatiskan tugas Anda.

Sebagai bagian dari solusi ini, instance EC2 Windows diluncurkan dengan menggunakan AMI dasar, dan aplikasi UiPath Studio diinstal pada instance. Pola ini menggunakan alat Microsoft System Preparation (Sysprep) untuk menduplikasi instalasi Windows yang disesuaikan. Setelah itu, ia menghapus informasi host dan membuat AMI akhir dari instance. Anda kemudian dapat meluncurkan instans sesuai permintaan dengan menggunakan AMI akhir dengan konvensi penamaan dan pengaturan pemantauan Anda sendiri.

> Catatan: Pola ini tidak memberikan informasi apa pun tentang penggunaan bot RPA. Untuk informasi tersebut, lihat UiPath dokumentasinya. Anda juga dapat menggunakan pola ini untuk mengatur aplikasi bot RPA lainnya dengan menyesuaikan langkah-langkah instalasi berdasarkan kebutuhan Anda.

Pola ini memberikan otomatisasi dan manfaat berikut:

- Penerapan dan berbagi aplikasi: Anda dapat membangun Amazon EC2 AMIs untuk penerapan aplikasi dan membagikannya di beberapa akun melalui pipeline EC2 Image Builder, yang menggunakan CloudFormation templat AWS sebagai infrastruktur sebagai skrip kode (IAc).
- EC2 Penyediaan dan penskalaan Amazon: Templat CloudFormation IAc menyediakan urutan nama komputer khusus dan otomatisasi gabungan Active Directory.
- Observabilitas dan pemantauan: Pola mengatur CloudWatch dasbor Amazon untuk membantu Anda memantau EC2 metrik Amazon (seperti penggunaan CPU dan disk).

- Manfaat RPA untuk bisnis Anda: RPA meningkatkan akurasi karena robot dapat melakukan tugas yang diberikan secara otomatis dan konsisten. RPA juga meningkatkan kecepatan dan produktivitas karena menghilangkan operasi yang tidak menambah nilai dan menangani aktivitas berulang.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Izin AWS Identity and Access Management (IAM) untuk menerapkan template CloudFormation
- Kebijakan IAM untuk menyiapkan distribusi AMI lintas akun dengan EC2 Image Builder

Arsitektur

1. Administrator menyediakan basis Windows AMI dalam `ec2-image-builder.yaml` file dan menyebarkan tumpukan di CloudFormation konsol.
2. CloudFormation Tumpukan menyebarkan pipeline EC2 Image Builder, yang mencakup sumber daya berikut:
   - `Ec2ImageInfraConfiguration`
   - `Ec2ImageComponent`
   - `Ec2ImageRecipe`
   - `Ec2AMI`
3. Pipeline EC2 Image Builder meluncurkan EC2 instance Windows sementara dengan menggunakan AMI dasar dan menginstal komponen yang diperlukan (dalam hal ini, UiPath Studio).
4. EC2 Image Builder menghapus semua informasi host dan membuat AMI dari Windows Server.
5. Anda memperbarui `ec2-provisioning yaml` file dengan AMI kustom dan meluncurkan sejumlah EC2 instance berdasarkan kebutuhan Anda.
6. Anda menerapkan makro Count dengan menggunakan template. CloudFormation Makro ini menyediakan properti Count untuk CloudFormation sumber daya sehingga Anda dapat menentukan beberapa sumber daya dari jenis yang sama dengan mudah.

7. Anda memperbarui nama makro dalam CloudFormation `ec2-provisioning.yaml` file dan menyebarkan tumpukan.
8. Administrator memperbarui `ec2-provisioning.yaml` file berdasarkan persyaratan dan meluncurkan tumpukan.
9. Template menyebarkan EC2 instance dengan aplikasi UiPath Studio.

Alat

Layanan AWS

- AWS CloudFormation membantu Anda memodelkan dan mengelola sumber daya infrastruktur secara otomatis dan aman.
- Amazon CloudWatch membantu Anda mengamati dan memantau sumber daya dan aplikasi di AWS, di tempat, dan di cloud lainnya.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang aman dan dapat diubah ukurannya di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- EC2 Image Builder menyederhanakan pembuatan, pengujian, dan penerapan mesin virtual dan image kontainer untuk digunakan di AWS atau di lokasi.
- Amazon EventBridge membantu Anda membangun aplikasi berbasis peristiwa dalam skala besar di seluruh AWS, sistem yang ada, atau aplikasi perangkat lunak sebagai layanan (SaaS).
- AWS Identity and Access Management (IAM) membantu Anda mengontrol akses ke sumber daya AWS dengan aman. Dengan IAM, Anda dapat mengelola izin secara terpusat yang mengontrol sumber daya AWS mana yang dapat diakses pengguna. Anda menggunakan IAM untuk mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan sumber daya.
- AWS Lambda adalah layanan komputasi tanpa server dan berbasis peristiwa yang memungkinkan Anda menjalankan kode untuk hampir semua jenis aplikasi atau layanan backend tanpa menyediakan atau mengelola server. Anda dapat memanggil fungsi Lambda dari lebih dari 200 layanan AWS dan aplikasi SaaS, dan hanya membayar untuk apa yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Systems Manager Agent (Agen SSM) membantu Systems Manager memperbarui, mengelola, dan mengonfigurasi EC2 instans, perangkat edge, server lokal, dan mesin virtual (). VMs

Repositori kode

Kode untuk pola ini tersedia dalam pengaturan bot GitHub UiPath RPA menggunakan CloudFormation repositori. Pola ini juga menggunakan makro yang tersedia dari repositori AWS CloudFormation Macros.

Praktik terbaik

- AWS merilis Windows baru AMIs setiap bulan. Ini berisi patch OS terbaru, driver, dan agen peluncuran. Kami menyarankan Anda menggunakan AMI terbaru saat meluncurkan instans baru atau saat Anda membuat gambar kustom Anda sendiri.
- Terapkan semua patch keamanan Windows atau Linux yang tersedia selama pembuatan gambar.

Epik

Menyebarkan pipeline gambar untuk gambar dasar

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan pipeline EC2 Image Builder.	1. Kloning pengaturan bot UiPath RPA menggunakan CloudFormation repositori, atau unduh `ec2-image-builder.yaml` templat dari repositori. 2. Masuk ke AWS Management Console, dan buka CloudFormation konsol AWS. 3. Pilih Buat tumpukan. 4. Di bagian Tentukan templat, pilih Unggah file templat. 5. Cari dan unggah `ec2-image-builder.yaml` templat dari komputer Anda, lalu pilih Berikutnya.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Berikan parameter input untuk tumpukan Anda atau terima nilai default. Pilih Berikutnya. **Note** Jumlah dan nilai parameter dapat bervariasi tergantung pada nilai masukan Anda. 7. Secara opsional, konfigura sikan opsi tumpukan, lalu pilih Berikutnya. 8. Tinjau detail tumpukan Anda. 9. Di akhir layar, pilih kotak centang untuk mengakui kemampuan, lalu pilih Kirim. 10. Pantau kemajuan tumpukan. Saat statusnya `CREATE_COMPLETE` , penerapan sudah siap.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat pengaturan EC2 Image Builder.	Pengaturan EC2 Image Builder mencakup konfigura si infrastruktur, pengaturan distribusi, dan pengaturan pemindaian keamanan. Untuk melihat pengaturan: 1. Buka konsol EC2 Image Builder. 2. Dari panel navigasi, navigasikan ke berbagai pengaturan Image Builder. **Note** Sebagai praktik terbaik, Anda harus membuat pembaruan apa pun ke EC2 Image Builder hanya melalui CloudForm ation template.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat pipa gambar.	Untuk melihat pipeline gambar yang digunakan: 1. Pada konsol EC2 Image Builder, pilih Pipeline gambar dari panel navigasi. 2. Pilih pipeline gambar yang Anda buat. 3. Lihat detail konfigurasi gambar keluaran, resep gambar, konfigurasi infrastruktur, pengatura n distribusi, EventBridge aturan Amazon, dan tag.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat log Image Builder.	EC2 Log Image Builder digabungkan dalam grup CloudWatch log. Untuk melihat log di CloudWatch: 1. Buka konsol CloudWatch . 2. Di panel navigasi, pilih Log, Grup log. 3. Pilih nama grup log. EC2 Log Image Builder digabungkan dalam grup `/aws/imagebuilder/XXX` log. 4. Periksa log terbaru di aliran log masing-masing untuk setiap kesalahan yang ditemui saat menjalankan pipeline gambar. EC2 Log Image Builder juga disimpan dalam bucket S3. Untuk melihat log di ember: 1. Buka konsol Amazon S3. 2. Di dalam daftar Bucket, pilih nama bucket. Log dikumpulkan dalam ember S3. `<stack-name>-XXXXXX`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah UiPath file ke bucket S3.	1. Unduh `.msi` file untuk UiPath Studio dari lokasi https://download.uipath.com/UiPathStudioCommunity.msi. 2. Unggah file ke bucket S3. 3. Perbarui nama bucket dan kunci file di `ec2-image-builder.yaml` template, di bagian data pengguna, nomor baris 310.	AWS DevOps

Menyebarkan dan menguji makro Count

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan makro Count.	1. Kloning atau unduh CloudFormation makro Count. 2. Arahkan ke `Count` folder. 3. Anda akan membutuhkan ember S3 untuk menyimpan CloudFormation artefak. Jika Anda belum memiliki bucket S3, buat bucket dengan namanya`aws s3 mb s3://<bucket name>`. 4. Package template makro Count. Template menggunakan AWS Serverless Application Model (SAM), sehingga	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	harus diubah sebelum Anda dapat menerapkannya.
	
	`aws cloudformation package \ --template-file template.yaml \ --s3-bucket <your bucket name here> \ --output-template-file packaged.yaml`
	
	Contoh:
	
	`aws cloudformation package \ --template-file template.yaml \ --s3-bucket count-macro-ec2 \ --output-template-file packaged.yaml`
	
	5. Menyebarkan template yang dikemas untuk membuat tumpukan. CloudFormation
	
	`aws cloudformation deploy \ --stack-name Count-macro \ --template-file packaged.yaml \ --capabilities CAPABILITY_IAM`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jika Anda ingin menggunak an konsol, ikuti instruksi di epik sebelumnya atau dalam CloudFormation dokumentasi.	
Uji makro Count.	Untuk menguji kemampuan makro, coba luncurkan contoh template yang disediakan dengan makro. `aws cloudformation deploy \ --stack-name Count-test \ --template-file test.yaml \ --capabilities CAPABILITY_IAM`	DevOps insinyur

Terapkan CloudFormation tumpukan untuk menyediakan instance dengan gambar khusus

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan template EC2 penyediaan Amazon.	Untuk menerapkan EC2 Image Pipeline dengan menggunakan CloudForm ation: 1. Unduh `ec2-provi sioning.yaml` template dari GitHub repositori, atau temukan di komputer Anda jika Anda mengkloning repositori. 2. Buka konsol CloudForm ation .	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Ulangi langkah-langkah dari epik pertama (atau ikuti instruksi dalam CloudFormation dokumentasi) untuk menyebarkan `ec2-provisioning.yaml` .	
Lihat EC2 pengaturan Amazon.	EC2 Pengaturan Amazon mencakup keamanan, jaringan, penyimpanan, pemeriksaan status, pemantauan, dan konfigurasi tag. Untuk melihat konfigurasi ini: 1. Buka EC2 konsol Amazon. 2. Di panel navigasi, pilih Instans, lalu pilih EC2 instance yang dibuat oleh template EC2 penyediaan Amazon. 3. Dalam ringkasan instance, pilih tab untuk melihat EC2 pengaturan Amazon yang sesuai.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat CloudWatch dasbor.	1. Buka konsol CloudWatch . 2. Pada panel navigasi, silakan pilih Dasbor. 3. Pilih dasbor yang memiliki nama tumpukan Anda. **Note** Setelah Anda menyediakan tumpukan, dibutuhkan waktu untuk mengisi dasbor dengan metrik. Dasbor menyediakan metrik berikut:`CPUUtilization` ,,`DiskUtilization` ,`MemoryUtilization` , `NetworkIn` `NetworkOut` ,`StatusCheckFailed` .	AWS DevOps
Lihat metrik khusus untuk penggunaan memori dan disk.	1. Di CloudWatch konsol, pilih Dasbor. 2. Pada panel navigasi, silakan pilih Metrik, Semua metrik. 3. Pilih ruang nama Kustom,. CWAgent	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat alarm untuk penggunaan memori dan disk.	1. Di CloudWatch konsol, di panel navigasi, pilih Dasbor. 2. Pilih Semua alarm.	AWS DevOps
Verifikasi aturan lifecyle snapshot.	1. Buka EC2 konsol Amazon. 2. Di panel navigasi, pilih Lifecycle Manager. 3. Verifikasi pengaturan untuk siklus hidup AMI.	AWS DevOps

Hapus lingkungan (opsional)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus tumpukan.	Ketika PoC atau proyek percontohan Anda selesai, kami sarankan Anda menghapus tumpukan yang Anda buat untuk memastikan bahwa Anda tidak dikenakan biaya untuk sumber daya ini. 1. Buka CloudFormation konsol AWS. 2. Di panel navigasi, pilih Tumpukan, lalu pilih salah satu atau kedua tumpukan yang Anda buat sebelumnya yang ingin Anda hapus. Tumpukan pasti sedang berjalan. 3. Di panel detail tumpukan, pilih Hapus.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Saat diminta, pilih Hapus tumpukan. ⚠ Important Operasi penghapusan tumpukan tidak dapat dihentikan setelah dimulai. Tumpukan diproses ke `DELETE_IN_PROGRESS` status. Jika penghapusan gagal, tumpukan akan berada dalam status. `DELETE_FAILED` Untuk solusinya, lihat Hapus tumpukan gagal dalam dokumentasi CloudFormation pemecahan masalah AWS. Untuk informasi tentang melindungi tumpukan agar tidak terhapus secara tidak sengaja, lihat Melindungi tumpukan agar tidak dihapus dalam CloudFormation dokumentasi AWS.	

Pemecahan masalah

Isu	Solusi
Saat Anda menerapkan template EC2 penyediaan Amazon, Anda mendapatkan kesalahan: Menerima respons cacat dari transform 123xxxx: :Count.	Ini adalah masalah yang diketahui. (Lihat solusi khusus dan PR di repositori CloudFormation makro AWS.) Untuk memperbaiki masalah ini, buka konsol AWS Lambda dan perbarui `index.py` dengan konten dari repositori. GitHub

Sumber daya terkait

GitHub repositori

- UiPath Pengaturan bot RPA menggunakan CloudFormation
- Hitung CloudFormation Makro

Referensi AWS

- Membuat tumpukan di CloudFormation konsol AWS (CloudFormation dokumentasi)
- Pemecahan masalah CloudFormation (dokumentasi) CloudFormation
- Memantau memori dan metrik disk untuk EC2 instans Amazon (dokumentasi Amazon) EC2
- Bagaimana saya bisa menggunakan CloudWatch agen untuk melihat metrik untuk Monitor Kinerja di server Windows? (AWS re:Artikel posting)

Referensi tambahan

- UiPath dokumentasi
- Mengatur Nama Host di SysPreped AMI (posting blog oleh Brian Beach)
- Bagaimana cara membuat Cloudformation memproses ulang template menggunakan makro saat parameter berubah? (Tumpukan Luapan)

Siapkan PeopleSoft arsitektur yang sangat tersedia di AWS

Ramanathan Muralidhar, Amazon Web Services

Ringkasan

Saat Anda memigrasikan PeopleSoft beban kerja Anda ke AWS, ketahanan adalah tujuan penting. Ini memastikan bahwa PeopleSoft aplikasi Anda selalu sangat tersedia dan dapat pulih dari kegagalan dengan cepat.

Pola ini menyediakan arsitektur untuk PeopleSoft aplikasi Anda di AWS untuk memastikan ketersediaan tinggi (HA) di tingkat jaringan, aplikasi, dan basis data. Ini menggunakan Amazon Relational Database Service (Amazon RDS) untuk Oracle atau Amazon RDS untuk database Amazon RDS for SQL Server untuk tingkat database. Arsitektur ini juga mencakup layanan AWS seperti Amazon Route 53, instans Amazon Elastic Compute Cloud (Amazon EC2) Linux, Amazon Elastic Block Storage (Amazon EBS), Amazon Elastic File System (Amazon EFS), dan Application Load Balancer, dan dapat diskalakan.

Oracle PeopleSoft menyediakan seperangkat alat dan aplikasi untuk manajemen tenaga kerja dan operasi bisnis lainnya.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- PeopleSoft Lingkungan dengan lisensi yang diperlukan untuk mengaturnya di AWS
- Virtual private cloud (VPC) disiapkan di akun AWS Anda dengan sumber daya berikut:
  - Setidaknya dua Availability Zone
  - Satu subnet publik dan tiga subnet pribadi di setiap Availability Zone
  - Gateway NAT dan gateway internet
  - Rute tabel untuk setiap subnet untuk merutekan lalu lintas
  - Daftar kontrol akses jaringan (jaringan ACLs) dan grup keamanan yang ditetapkan untuk membantu memastikan keamanan PeopleSoft aplikasi sesuai dengan standar organisasi Anda

Batasan

- Pola ini memberikan solusi ketersediaan tinggi (HA). Itu tidak mendukung skenario pemulihan bencana (DR). Dalam kejadian langka bahwa seluruh Wilayah AWS untuk implementasi HA turun, aplikasi akan menjadi tidak tersedia.

Versi produk

- PeopleSoft aplikasi yang berjalan PeopleTools 8.52 dan yang lebih baru

Arsitektur

Arsitektur target

Downtime atau pemadaman aplikasi PeopleSoft produksi Anda berdampak pada ketersediaan aplikasi dan menyebabkan gangguan besar pada bisnis Anda.

Kami menyarankan Anda merancang aplikasi PeopleSoft produksi Anda sehingga selalu tersedia. Anda dapat mencapai ini dengan menghilangkan satu titik kegagalan, menambahkan titik crossover atau failover yang andal, dan mendeteksi kegagalan. Diagram berikut mengilustrasikan arsitektur HA untuk PeopleSoft AWS.

Penerapan arsitektur ini menggunakan Amazon RDS for Oracle sebagai database, EC2 dan instance yang berjalan PeopleSoft di Red Hat Enterprise Linux (RHEL). Anda juga dapat menggunakan Amazon RDS for SQL Server sebagai database Peoplesoft.

Arsitektur ini berisi komponen-komponen berikut:

- Amazon Route 53 digunakan sebagai Domain Name Server (DNS) untuk routing request dari internet ke aplikasi. PeopleSoft
- AWS WAF membantu Anda melindungi dari eksploitasi web umum dan bot yang dapat memengaruhi ketersediaan, membahayakan keamanan, atau mengkonsumsi sumber daya yang berlebihan. AWS Shield Advanced (tidak diilustrasikan) memberikan perlindungan yang jauh lebih luas.
- Application Load Balancer menyeimbangkan lalu lintas HTTP dan HTTPS dengan routing permintaan lanjutan yang ditargetkan pada server web.
- Server web, server aplikasi, server penjadwal proses, dan server Elasticsearch yang mendukung PeopleSoft aplikasi berjalan di beberapa Availability Zone dan menggunakan Amazon Auto EC2 Scaling.

- Basis data yang digunakan oleh PeopleSoft aplikasi berjalan di Amazon RDS dalam konfigurasi Multi-AZ.
- Berbagi file yang digunakan oleh PeopleSoft aplikasi dikonfigurasi di Amazon EFS dan digunakan untuk mengakses file di seluruh instance.
- Amazon Machine Images (AMI s) digunakan oleh Amazon EC2 Auto Scaling untuk memastikan bahwa PeopleSoft komponen dikloning dengan cepat saat diperlukan.
- Gateway NAT menghubungkan instance di subnet pribadi ke layanan di luar VPC Anda, dan memastikan bahwa layanan eksternal tidak dapat memulai koneksi dengan instans tersebut.
- Gateway internet adalah komponen VPC berskala horizontal, berlebihan, dan sangat tersedia yang memungkinkan komunikasi antara VPC Anda dan internet.
- Host benteng di subnet publik menyediakan akses ke server di subnet pribadi dari jaringan eksternal, seperti internet atau jaringan lokal. Host benteng menyediakan akses terkontrol dan aman ke server di subnet pribadi.

Detail arsitektur

PeopleSoft Basis data disimpan di database Amazon RDS for Oracle (atau Amazon RDS for SQL Server) dalam konfigurasi Multi-AZ. Fitur Amazon RDS Multi-AZ mereplikasi pembaruan basis data di dua Availability Zone untuk meningkatkan daya tahan dan ketersediaan. Amazon RDS secara otomatis gagal ke database siaga untuk pemeliharaan terencana dan gangguan yang tidak direncanakan.

PeopleSoft Web dan tingkat menengah diinstal pada EC2 instance. Instans ini tersebar di beberapa Availability Zone dan diikat oleh grup Auto Scaling. Ini memastikan bahwa komponen-komponen ini selalu sangat tersedia. Jumlah minimum instans yang diperlukan dipertahankan untuk memastikan bahwa aplikasi selalu tersedia dan dapat diskalakan saat diperlukan.

Kami menyarankan Anda menggunakan jenis EC2 instans generasi saat ini untuk EC2 instans OEM. Jenis instans generasi saat ini, seperti instans yang dibangun di AWS Nitro System, mendukung perangkat keras mesin virtual ()HVMs. HVM AMIs diperlukan untuk memanfaatkan jaringan yang ditingkatkan, dan mereka juga menawarkan peningkatan keamanan. EC2 Instans yang merupakan bagian dari setiap grup Auto Scaling menggunakan AMI mereka sendiri saat mengganti atau meningkatkan instance. Sebaiknya pilih jenis EC2 instans berdasarkan beban yang ingin ditangani PeopleSoft aplikasi dan nilai minimum yang direkomendasikan oleh Oracle untuk PeopleSoft aplikasi dan PeopleTools rilis Anda. Untuk informasi selengkapnya tentang persyaratan perangkat keras dan perangkat lunak, lihat situs web dukungan Oracle.

PeopleSoft Web dan tingkat menengah berbagi mount Amazon EFS untuk berbagi laporan, file data, dan (jika diperlukan) `PS_HOME` direktori. Amazon EFS dikonfigurasi dengan target pemasangan di setiap Availability Zone untuk alasan kinerja dan biaya.

Application Load Balancer disediakan untuk mendukung lalu lintas yang mengakses PeopleSoft aplikasi dan menyeimbangkan lalu lintas di antara server web di berbagai Availability Zone. Application Load Balancer adalah perangkat jaringan yang menyediakan HA di setidaknya dua Availability Zone. Server web mendistribusikan lalu lintas ke server aplikasi yang berbeda dengan menggunakan konfigurasi load balancing. Load balancing antara server web dan server aplikasi memastikan bahwa beban didistribusikan secara merata di seluruh instance, dan membantu menghindari kemacetan dan gangguan layanan karena instance yang kelebihan beban.

Amazon Route 53 digunakan sebagai layanan DNS untuk merutekan lalu lintas ke Application Load Balancer dari internet. Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.

Rincian HA

- Database: Fitur Multi-AZ Amazon RDS mengoperasikan dua database di beberapa Availability Zone dengan replikasi sinkron. Ini menciptakan lingkungan yang sangat tersedia dengan failover otomatis. Amazon RDS memiliki deteksi peristiwa failover dan memulai failover otomatis saat peristiwa ini terjadi. Anda juga dapat memulai failover manual melalui Amazon RDS API. Untuk penjelasan rinci, lihat posting blog Amazon RDS Under The Hood: Multi-AZ. Failover mulus dan aplikasi secara otomatis terhubung kembali ke database ketika itu terjadi. Namun, setiap pekerjaan penjadwal proses selama failover menghasilkan kesalahan dan harus dikirim ulang.
- PeopleSoft server aplikasi: Server aplikasi tersebar di beberapa Availability Zone dan memiliki grup Auto Scaling yang ditentukan untuknya. Jika instance gagal, grup Auto Scaling segera menggantinya dengan instance sehat yang dikloning dari AMI template server aplikasi. Secara khusus, penyatuan sentakan diaktifkan, jadi ketika instance server aplikasi turun, sesi secara otomatis gagal ke server aplikasi lain, dan grup Auto Scaling secara otomatis memutar instance lain, memunculkan server aplikasi, dan mendaftarkannya di mount Amazon EFS. Server aplikasi yang baru dibuat secara otomatis ditambahkan ke server web dengan menggunakan `PSSTRSETUP.SH` skrip di server web. Ini memastikan bahwa server aplikasi selalu sangat tersedia dan pulih dari kegagalan dengan cepat.
- Penjadwal proses: Server penjadwal proses tersebar di beberapa Availability Zone dan memiliki grup Auto Scaling yang ditentukan untuknya. Jika instans gagal, grup Auto Scaling segera menggantinya dengan instance sehat yang dikloning dari AMI templat server penjadwal proses. Khususnya, ketika instance penjadwal proses turun, grup Auto Scaling secara otomatis memutar

instance lain dan memunculkan penjadwal proses. Pekerjaan apa pun yang berjalan saat instance gagal harus dikirimkan ulang. Ini memastikan bahwa penjadwal proses selalu tersedia dan pulih dari kegagalan dengan cepat.

- Server Elasticsearch: Server Elasticsearch memiliki grup Auto Scaling yang ditentukan untuk mereka. Jika instance gagal, grup Auto Scaling segera menggantinya dengan instance sehat yang dikloning dari AMI template server Elasticsearch. Secara khusus, ketika instance Elasticsearch turun, Application Load Balancer yang melayani permintaan untuk mendeteksi kegagalan dan berhenti mengirim lalu lintas ke sana. Grup Auto Scaling secara otomatis memutar instance lain dan memunculkan instance Elasticsearch. Ketika instance Elasticsearch di-back up, Application Load Balancer mendeteksi bahwa itu sehat dan mulai mengirim permintaan ke sana lagi. Ini memastikan bahwa server Elasticsearch selalu sangat tersedia dan pulih dari kegagalan dengan cepat.
- Server web: Server web memiliki grup Auto Scaling yang ditentukan untuk mereka. Jika instance gagal, grup Auto Scaling segera menggantinya dengan instance sehat yang dikloning dari AMI template server web. Secara khusus, ketika instance server web turun, Application Load Balancer yang melayani permintaan untuk mendeteksi kegagalan dan berhenti mengirim lalu lintas ke sana. Grup Auto Scaling secara otomatis memutar instance lain dan memunculkan instance server web. Ketika instance server web di-back up, Application Load Balancer mendeteksi bahwa itu sehat dan mulai mengirim permintaan ke sana lagi. Ini memastikan bahwa server web selalu sangat tersedia dan pulih dari kegagalan dengan cepat.

Alat

Layanan AWS

- Application Load Balancer mendistribusikan lalu lintas aplikasi yang masuk di beberapa target, seperti EC2 instance, di beberapa Availability Zone.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon Elastic Compute Cloud (Amazon). EC2
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di AWS Cloud.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.

- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.

Praktik terbaik

Praktik terbaik operasional

- Saat Anda menjalankan PeopleSoft AWS, gunakan Route 53 untuk merutekan lalu lintas dari internet dan lokal. Gunakan opsi failover untuk mengalihkan lalu lintas ke situs pemulihan bencana (DR) jika instans DB utama tidak tersedia.
- Selalu gunakan Application Load Balancer di depan lingkungan. PeopleSoft Ini memastikan bahwa lalu lintas seimbang beban ke server web dengan cara yang aman.
- Dalam pengaturan grup target Application Load Balancer, pastikan lengket dihidupkan dengan cookie yang dihasilkan load balancer.

> **Note**
>
> Anda mungkin perlu menggunakan cookie berbasis aplikasi jika Anda menggunakan sistem masuk tunggal eksternal (SSO). Ini memastikan bahwa koneksi konsisten di seluruh server web dan server aplikasi.

- Untuk aplikasi PeopleSoft produksi, batas waktu idle Application Load Balancer harus sesuai dengan apa yang diatur dalam profil web yang Anda gunakan. Ini mencegah sesi pengguna kedaluwarsa di lapisan penyeimbang beban.
- Untuk aplikasi PeopleSoft produksi, atur jumlah daur ulang server aplikasi ke nilai yang meminimalkan kebocoran memori.
- Jika Anda menggunakan database Amazon RDS untuk aplikasi PeopleSoft produksi Anda, seperti yang dijelaskan dalam pola ini, jalankan dalam format Multi-AZ untuk ketersediaan tinggi.
- Jika database Anda berjalan pada EC2 instance untuk aplikasi PeopleSoft produksi Anda, pastikan database siaga berjalan di Availability Zone lain untuk ketersediaan tinggi.
- Untuk DR, pastikan database atau EC2 instans Amazon RDS Anda memiliki siaga yang dikonfigurasi di Wilayah AWS terpisah dari database produksi. Ini memastikan bahwa jika terjadi bencana di Wilayah, Anda dapat mengalihkan aplikasi ke Wilayah lain.
- Untuk DR, gunakan Amazon Elastic Disaster Recovery untuk menyiapkan komponen tingkat aplikasi di Wilayah terpisah dari komponen produksi. Ini memastikan bahwa jika terjadi bencana di Wilayah, Anda dapat mengalihkan aplikasi ke Wilayah lain.

- Gunakan Amazon EFS (untuk I/O persyaratan moderat) atau Amazon FSx (untuk I/O persyaratan tinggi) untuk menyimpan PeopleSoft laporan, lampiran, dan file data Anda. Ini memastikan bahwa konten disimpan di satu lokasi pusat dan dapat diakses dari mana saja di dalam infrastruktur.
- Gunakan Amazon CloudWatch (dasar dan terperinci) untuk memantau sumber daya AWS Cloud yang digunakan PeopleSoft aplikasi Anda dalam waktu dekat. Ini memastikan bahwa Anda diberitahu tentang masalah secara instan dan dapat mengatasinya dengan cepat sebelum memengaruhi ketersediaan lingkungan.
- Jika Anda menggunakan database Amazon RDS sebagai PeopleSoft database, gunakan Enhanced Monitoring. Fitur ini menyediakan akses ke lebih dari 50 metrik, termasuk CPU, memori, sistem I/O, and disk I/O file.
- Gunakan AWS CloudTrail untuk memantau panggilan API pada sumber daya AWS yang digunakan PeopleSoft aplikasi Anda. Ini membantu Anda melakukan analisis keamanan, pelacakan perubahan sumber daya, dan audit kepatuhan.

Praktik terbaik keamanan

- Untuk melindungi PeopleSoft aplikasi Anda dari eksploitasi umum seperti injeksi SQL atau cross-site scripting (XSS), gunakan AWS WAF. Pertimbangkan untuk menggunakan AWS Shield Advanced untuk layanan deteksi dan mitigasi yang disesuaikan.
- Tambahkan aturan ke Application Load Balancer untuk mengarahkan lalu lintas dari HTTP ke HTTPS secara otomatis untuk membantu mengamankan aplikasi Anda. PeopleSoft
- Siapkan grup keamanan terpisah untuk Application Load Balancer. Grup keamanan ini seharusnya hanya mengizinkan lalu lintas HTTPS/HTTP masuk dan tidak ada lalu lintas keluar. Ini memastikan bahwa hanya lalu lintas yang dimaksudkan yang diizinkan dan membantu mengamankan aplikasi Anda.
- Gunakan subnet pribadi untuk server aplikasi, server web, dan database, dan gunakan gateway NAT untuk lalu lintas internet keluar. Ini memastikan bahwa server yang mendukung aplikasi tidak dapat dijangkau oleh publik, sementara menyediakan akses publik hanya ke server yang membutuhkannya.
- Gunakan berbeda VPCs untuk menjalankan lingkungan PeopleSoft produksi dan non-produksi Anda. Gunakan AWS Transit Gateway, peering VPC, jaringan ACLs, dan grup keamanan untuk mengontrol arus lalu lintas antara VPC dan, jika perlu, pusat data lokal Anda.
- Ikuti prinsip hak istimewa yang paling sedikit. Berikan akses ke sumber daya AWS yang digunakan oleh PeopleSoft aplikasi hanya untuk pengguna yang benar-benar membutuhkannya.

Berikan hanya hak istimewa minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat pilar keamanan AWS Well-Architected Framework.

- Jika memungkinkan, gunakan AWS Systems Manager untuk mengakses EC2 instans yang digunakan PeopleSoft aplikasi.

Praktik terbaik keandalan

- Saat Anda menggunakan Application Load Balancer, daftarkan satu target untuk setiap Availability Zone yang diaktifkan. Ini membuat penyeimbang beban paling efektif.
- Kami menyarankan Anda memiliki tiga yang berbeda URLs untuk setiap lingkungan PeopleSoft produksi: satu URL untuk mengakses aplikasi, satu untuk melayani broker integrasi, dan satu untuk melihat laporan. Jika memungkinkan, setiap URL harus memiliki server web dan server aplikasi khusus sendiri. Desain ini membantu membuat PeopleSoft aplikasi Anda lebih aman, karena setiap URL memiliki fungsi yang berbeda dan akses terkontrol. Ini juga meminimalkan ruang lingkup dampak jika layanan yang mendasarinya gagal.
- Kami menyarankan Anda mengonfigurasi pemeriksaan kesehatan pada grup target penyeimbang beban untuk PeopleSoft aplikasi Anda. Pemeriksaan kesehatan harus dilakukan di server web alih-alih EC2 instance yang menjalankan server tersebut. Ini memastikan bahwa jika server web mogok atau EC2 instance yang menghosting server web turun, Application Load Balancer mencerminkan informasi tersebut secara akurat.
- Untuk aplikasi PeopleSoft produksi, kami menyarankan Anda menyebarkan server web di setidaknya tiga Availability Zone. Ini memastikan bahwa PeopleSoft aplikasi selalu sangat tersedia bahkan jika salah satu Availability Zone turun.
- Untuk aplikasi PeopleSoft produksi, aktifkan jolt pooling (). `joltPooling=true` Ini memastikan bahwa aplikasi Anda gagal ke server aplikasi lain jika server down untuk tujuan patching atau karena kegagalan VM.
- Untuk aplikasi PeopleSoft produksi, atur `DynamicConfigReload` ke 1. Pengaturan ini didukung di PeopleTools versi 8.52 dan yang lebih baru. Ini menambahkan server aplikasi baru ke server web secara dinamis, tanpa me-restart server.
- Untuk meminimalkan waktu henti saat menerapkan PeopleTools tambalan, gunakan metode blue/green penerapan untuk konfigurasi peluncuran grup Auto Scaling untuk server web dan aplikasi. Untuk informasi selengkapnya, lihat Ikhtisar opsi penerapan di whitepaper AWS.
- Gunakan AWS Backup untuk mencadangkan PeopleSoft aplikasi Anda di AWS. AWS Backup adalah layanan berbasis kebijakan yang hemat biaya, dikelola sepenuhnya, yang menyederhanakan perlindungan data dalam skala besar.

Praktik terbaik kinerja

- Hentikan SSL di Application Load Balancer untuk kinerja lingkungan yang optimal, kecuali bisnis Anda memerlukan lalu lintas terenkripsi di seluruh lingkungan. PeopleSoft
- Buat titik akhir VPC antarmuka untuk layanan AWS seperti Amazon Simple Notification Service (Amazon SNS) CloudWatchdan agar lalu lintas selalu internal. Ini hemat biaya dan membantu menjaga keamanan aplikasi Anda.

Praktik terbaik pengoptimalan biaya

- Tandai semua sumber daya yang digunakan oleh PeopleSoft lingkungan Anda, dan aktifkan tag alokasi biaya. Tag ini membantu Anda melihat dan mengelola biaya sumber daya Anda.
- Untuk aplikasi PeopleSoft produksi, siapkan grup Auto Scaling untuk server web dan server aplikasi. Ini mempertahankan jumlah minimal server web dan aplikasi untuk mendukung aplikasi Anda. Anda dapat menggunakan kebijakan grup Auto Scaling untuk meningkatkan skala server ke atas dan ke bawah sesuai kebutuhan.
- Gunakan alarm penagihan untuk mendapatkan peringatan ketika biaya melebihi ambang anggaran yang Anda tentukan.

Praktik terbaik keberlanjutan

- Gunakan infrastruktur sebagai kode (IAc) untuk menjaga PeopleSoft lingkungan Anda. Ini membantu Anda membangun lingkungan yang konsisten dan mempertahankan kontrol perubahan.

Epik

Migrasi PeopleSoft database Anda ke Amazon RDS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup subnet DB.	Di konsol Amazon RDS, di panel navigasi, pilih grup Subnet, lalu buat grup subnet Amazon RDS DB dengan subnet di beberapa Availabil ity Zone. Ini diperlukan agar	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	database Amazon RDS berjalan dalam konfigurasi Multi-AZ.	
Buat database Amazon RDS.	Buat database Amazon RDS di Availability Zone Wilayah AWS yang Anda pilih untuk lingkungan PeopleSoft HA. Saat Anda membuat database Amazon RDS, pastikan untuk memilih opsi Multi-AZ (Buat instance siaga) dan grup subnet database yang Anda buat pada langkah sebelumnya. Untuk informasi selengkapnya, lihat dokumentasi Amazon RDS.	Administrator cloud, administrator basis data Oracle
Migrasikan PeopleSoft database Anda ke Amazon RDS.	Migrasikan PeopleSoft database yang ada ke database Amazon RDS dengan menggunakan AWS Database Migration Service (AWS DMS). Untuk informasi selengkapnya, lihat dokumentasi AWS DMS dan postingan blog AWS Migrating Oracle database dengan downtime mendekati nol menggunakan AWS DMS.	Administrator awan, PeopleSoft DBA

Siapkan sistem file Amazon EFS Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file.	Di konsol Amazon EFS, buat sistem file dan pasang target untuk setiap Availabil ity Zone. Untuk petunjuk, lihat dokumentasi Amazon EFS. Ketika sistem file telah dibuat, perhatikan nama DNS-nya. Anda akan menggunakan informasi ini ketika Anda me-mount sistem file.	Administrator awan

Siapkan PeopleSoft aplikasi dan sistem file Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan sebuah EC2 instance.	Luncurkan EC2 instance untuk PeopleSoft aplikasi Anda. Untuk petunjuk, lihat EC2 dokumentasi Amazon. • Untuk Nama, masukkan `APP_TEMPLATE` . • Untuk gambar OS, pilih Red Hat. • Untuk tipe Instance, pilih jenis instance yang sesuai untuk PeopleSoft aplikasi Anda. Untuk informasi selengkapnya, lihat Detail arsitektur di bagian Arsitektu r.	Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal PeopleSoft pada instance.	Instal PeopleSoft aplikasi Anda dan PeopleTools pada EC2 instance yang Anda buat. Untuk instruksi, lihat dokumentasi Oracle.	Administrator cloud, PeopleSoft administrator
Buat server aplikasi.	Buat server aplikasi untuk template AMI dan pastikan itu berhasil terhubung ke database Amazon RDS.	Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Pasang sistem file Amazon EFS.	Masuk ke EC2 instance sebagai pengguna root dan jalankan perintah berikut untuk memasang sistem file Amazon EFS ke folder yang dipanggil `PSFTMNT` di server.
	`sudo su -`
	`mkdir /psftmnt`
	`cat /etc/fstab`
	Tambahkan baris berikut ke `/etc/fstab` file. Gunakan nama DNS yang Anda catat saat Anda membuat sistem file.
	`fs-09e064308f11453 88.efs.us-east-1.a mazonaws.com:/ / psftmnt nfs4 nfsvers=4 .1,rsize=1048576,w size=1048576,hard, timeo=600,retrans= 2,noresvport,_netdev 0 0`
	`mount -a`
		Administrator cloud, PeopleSoft administrator

Periksa izin.	Pastikan `PSFTMNT` folder memiliki izin yang tepat sehingga PeopleSoft pengguna dapat mengakses nya dengan benar.	Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance tambahan.	Ulangi langkah sebelumnya dalam epik ini untuk membuat instance template untuk penjadwal proses, server web, dan server Elasticsearch. Sebutkan contoh ini`PRCS_TEMPLATE` ,`WEB_TEMPLATE` , dan`SRCH_TEMPLATE` . Untuk server web, atur `joltPooling=true` dan`DynamicConfigReload=1` .	Administrator cloud, PeopleSoft administrator

Buat skrip untuk mengatur server

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat skrip untuk menginstal server aplikasi.	Dalam EC2 `APP_TEMPLATE` contoh Amazon, sebagai PeopleSoft pengguna, buat skrip berikut. Beri nama `appstart.sh` dan letakkan di `PS_HOME` direktori. Anda akan menggunakan skrip ini untuk memunculkan server aplikasi dan juga merekam nama server di mount Amazon EFS.
	`#!/bin/ksh`
	`. /usr/homes/hcmdemo/.profile.`
	`psadmin -c configure -d HCMDEMO`
		PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`psadmin -c parallelb oot -d HCMDEMO touch /psftmnt/`echo $HOSTNAME`	
Buat skrip untuk menginstal server penjadwal proses.	Dalam EC2 `PRCS_TEMP LATE` contoh Amazon, sebagai PeopleSoft pengguna, buat skrip berikut. Beri nama `prcsstart.sh` dan letakkan di `PS_HOME` direktori . Anda akan menggunakan skrip ini untuk memunculkan server penjadwal proses. `#!/bin/ksh . /usr/homes/hcmdemo/. profile /* The following line ensures that the process scheduler always has a unique name during replaceme nt or scaling activity. */ sed -i "s/.*Prcs ServerName.*/`host name -I | awk -F. '{print "PrcsServ erName=PSUNX"$3$4} '`/" $HOME/appserv/ prcs/*/psprcs.cfg psadmin -p configure -d HCMDEMO psadmin -p start -d HCMDEMO`	PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat skrip untuk menginstal server Elasticsearch.	Dalam EC2 `SRCH_TEMPLATE` contoh Amazon, sebagai pengguna Elasticsearch, buat skrip berikut. Beri nama `srchstart.sh` dan letakkan di `HOME` direktori.
	`#!/bin/ksh`
	`/* The following line ensures that the correct IP is indicated in the elasticsearch.yaml file. */`
	`sed -i "s/.*network.host.*/`hostname -I
		awk '{print "host:"$0}'`/" $ES_HOME_DIR/config/elasticsearch.yaml`
		`nohup $ES_HOME_DIR/bin/elasticsearch &`
			PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skrip untuk menginstal server web.	Dalam EC2 `WEB_TEMPLATE` contoh Amazon, sebagai pengguna server web, buat skrip berikut di `HOME` direktori. `renip.sh`: Skrip ini memastikan bahwa server web memiliki IP yang benar ketika dikloning dari AMI. `#!/bin/ksh` `hn=`hostname`` `/* On the following` ` line, change the IP` ` with the hostname with` ` the hostname of the web` ` template. */` `for text_file in `find` ` * -type f -exec grep` ` -l '<hostname-of-the-` `web-template>' {} \;` `do` `sed -e 's/<hostn` `ame-of-the-web-tem` `plate>/'$hn'/g'` ` $text_file > temp` `mv -f temp $text_file` `done`  `psstrsetup.sh` : Skrip ini memastikan bahwa server web menggunakan server aplikasi IPs yang benar yang sedang berjalan. Ia mencoba untuk terhubung ke setiap server aplikasi pada port sentakan dan menambahk annya ke file konfigurasi.	PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`#!/bin/ksh` `c2=""` `for ctr in `ls -1 /psftmnt/*.internal` `do` `c1=`echo $ctr | awk -F "/" '{print $3}'` `/* In the following lines, 9000 is the jolt port. Change it if necessary. */` `if nc -z $c1 9000 2> /dev/null; then` `if [[ $c2 = "" ]]; then` `c2="psserver="`echo $c1`":9000"` `else` `c2=`echo $c2`","`echo $c1`":9000"` `fi` `fi` `done` `webstart.sh` : Skrip ini menjalankan dua skrip sebelumnya dan memulai server web. `#!/bin/ksh` `/* Change the path in the following if necessary. */` `cd /usr/homes/hcmdemo` `./renip.sh` `./psstrsetup.sh` `webserv/peoplesoft/bin/startPIA.sh`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan entri crontab.	Dalam EC2 `WEB_TEMPLATE` contoh Amazon, sebagai pengguna server web, tambahkan baris berikut ke crontab. Ubah waktu dan jalur untuk mencerminkan nilai yang Anda butuhkan. Entri ini memastikan bahwa server web Anda selalu memiliki entri server aplikasi yang benar dalam `configuration.properties` file. `* * * * * /usr/homes/hcmdemo/psstrsetup.sh`	PeopleSoft administrator

Buat AMIs dan templat grup Auto Scaling

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AMI untuk template server aplikasi.	Di EC2 konsol Amazon, buat gambar AMI dari EC2 `APP_TEMPLATE` instans Amazon. Beri nama `AMIPSAPPSRV-SCG-VER1` . Untuk petunjuk, lihat EC2 dokumentasi Amazon.	Administrator cloud, PeopleSoft administrator
Buat AMIs untuk server lain.	Ulangi langkah sebelumnya AMIs untuk membuat penjadwal proses, server Elasticsearch, dan server web.	Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat template peluncuran untuk grup Auto Scaling server aplikasi.	Buat template peluncuran untuk grup Auto Scaling server aplikasi. Beri nama template `PSAPPSRV_TEMPLATE`. Dalam template, pilih AMI yang Anda buat untuk `APP_TEMPLATE` instance. Untuk petunjuk, lihat EC2 dokumentasi Amazon.
	• Dalam template peluncuran, pilih jenis instans berdasarkan kebutuhan Anda.
	• Di bidang Data pengguna pada bagian Detail lanjutan, tambahkan entri berikut. Pastikan jalur dan informasi pengguna sudah benar. Anda membuat `appstart.sh` skrip di langkah sebelumnya.
	`#! /bin/ksh`
	`su -c "/usr/homes/hcmdemo/appstart.sh" - hcmdemo`
		Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat template peluncuran untuk grup Auto Scaling server penjadwal proses.	Ulangi langkah sebelumnya untuk membuat template peluncuran untuk grup Auto Scaling server penjadwal proses. Beri nama template`PSPRCS_TEMPLATE` . Dalam template, pilih AMI yang Anda buat untuk penjadwal proses.
	• Di bidang Data pengguna pada bagian Detail lanjutan, tambahkan entri berikut. Pastikan jalur dan informasi pengguna sudah benar. Anda membuat `prcsstart.sh` skrip di langkah sebelumnya.
	`#! /bin/ksh`
	`su -c "/usr/homes/hcmdemo/prcsstart.sh" - hcmdemo`
		Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat template peluncuran untuk grup Auto Scaling server Elasticsearch.	Ulangi langkah sebelumnya untuk membuat template peluncuran untuk grup Auto Scaling server Elasticsearch. Beri nama template`SRCH_TEMPLATE` . Dalam template, pilih AMI yang Anda buat untuk server pencarian.
	• Di bidang Data pengguna pada bagian Detail lanjutan, tambahkan entri berikut. Pastikan jalur dan informasi pengguna sudah benar. Anda membuat `srchstart.sh` skrip di langkah sebelumnya.
	`#! /bin/ksh`
	`su -c "/usr/hom`
	`es/essearch/srchst`
	`art.sh" - essearch`
		Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat template peluncuran untuk grup Auto Scaling server web.	Ulangi langkah sebelumny a untuk membuat template peluncuran untuk grup Auto Scaling server web. Beri nama template`WEB_TEMPLATE` . Dalam template, pilih AMI yang Anda buat untuk server web.
	• Di bidang Data pengguna pada bagian Detail lanjutan, tambahkan entri berikut. Pastikan jalur dan informasi pengguna sudah benar. Anda membuat `webstart.sh` skrip di langkah sebelumnya.
	`#! /bin/ksh`
	`su -c "/usr/homes/hcmdemo/webstart.sh" - hcmdemo`
		Administrator cloud, PeopleSoft administrator

Buat grup Auto Scaling

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup Auto Scaling untuk server aplikasi.	Di EC2 konsol Amazon, buat grup Auto Scaling yang dipanggil `PSAPPSRV_ASG` untuk server aplikasi dengan menggunakan template. `PSAPPSRV_TEMPLATE`	Administrator cloud, PeopleSoft administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk petunjuk, lihat EC2 dokumentasi Amazon. • Pada halaman Pilih opsi peluncuran instance, pilih VPC yang benar dan kemudian pilih beberapa subnet dari Availability Zone yang berbeda. • Pada halaman Configure advanced options, jangan pilih load balancer. • Pada halaman Konfigurasi ukuran grup dan kebijakan penskalaan, pilih setelan tergantung pada seberapa banyak beban yang Anda inginkan untuk merancang sistem Anda dan apakah Anda ingin menggunak an kebijakan penskalaan. Kami menyarankan Anda mengatur kapasitas yang diinginkan dan minimum menjadi 2 minimum sehingga setidaknya satu instance tersedia untuk melayani lalu lintas kapan saja. Untuk informasi selengkapnya tentang kebijakan Auto Scaling, lihat dokumentasi Amazon EC2 .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup Auto Scaling untuk server lain.	Ulangi langkah sebelumnya untuk membuat grup Auto Scaling untuk penjadwal proses, server Elasticsearch, dan server web.	Administrator cloud, PeopleSoft administrator

Buat dan konfigurasikan grup target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup target untuk server web.	Di EC2 konsol Amazon, buat grup target untuk server web. Untuk petunjuk, lihat dokumentasi Elastic Load Balancing. Atur port ke port tempat server web mendengarkan.	Administrator awan
Konfigurasikan pemeriksaan kesehatan.	Konfirmasikan bahwa pemeriksaan kesehatan memiliki nilai yang benar untuk mencerminkan kebutuhan bisnis Anda. Untuk informasi lebih lanjut, lihat Dokumentasi Penyeimbangan Beban Elastis.	Administrator awan
Buat grup target untuk server Elasticsearch.	Ulangi langkah sebelumnya untuk membuat grup target yang dipanggil `PSFTSRCH` untuk server Elasticsearch, dan atur port Elasticsearch yang benar.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan grup target ke grup Auto Scaling.	Buka grup Auto Scaling server web yang disebut `PSPIA_ASG` yang Anda buat sebelumnya. Pada tab Load balancing, pilih Edit dan kemudian tambahkan grup `PSFTWEB` target ke grup Auto Scaling. Ulangi langkah ini untuk grup Auto Scaling Elasticsearch untuk menambahkan `PSSRCH_ASG` `PSFTSRCH` grup target yang Anda buat sebelumnya.	Administrator awan
Atur kelengketan sesi.	Di grup target`PSFTWEB`, pilih tab Atribut, pilih Edit, dan atur kelengketan sesi. Untuk jenis lengket, pilih Cookie yang dihasilkan penyeimbang beban, dan atur durasinya ke 1. Untuk informasi lebih lanjut, lihat Dokumentasi Penyeimbangan Beban Elastis. Ulangi langkah ini untuk kelompok sasaran`PSFTSRCH`.	Administrator awan

Buat dan konfigurasikan penyeimbang beban aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat penyeimbang beban untuk server web.	Buat Application Load Balancer bernama `PSFTLB` untuk memuat keseimban	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	gan lalu lintas ke server web. Untuk petunjuk, lihat dokumentasi Elastic Load Balancing. • Berikan nama penyeimbang beban. • Untuk Skema, pilih yang menghadap ke Internet. • Di bagian pemetaan Jaringan, pilih VPC yang benar dan setidaknya dua subnet publik dari Availabil ity Zone yang berbeda. • Di bagian Listeners and routing, pilih grup target `PSFTWEB` dan tentukan protokol dan nomor port yang benar.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat penyeimbang beban untuk server Elasticsearch.	Buat Application Load Balancer bernama PSFTSCH untuk menyeimbangkan lalu lintas beban ke server Elasticsearch. • Berikan nama penyeimbang beban. • Untuk Skema, pilih Internal. • Di bagian Pemetaan jaringan, pilih VPC dan subnet pribadi yang benar. • Di bagian Listeners and routing, pilih grup target PSFTSRCH dan tentukan protokol dan nomor port yang benar.	Administrator awan
Konfigurasikan Route 53.	Di konsol Amazon Route 53, buat catatan di zona yang dihosting yang akan melayani PeopleSoft aplikasi. Untuk petunjuknya, lihat dokumentasi Amazon Route 53. Ini memastikan bahwa semua lalu lintas melewati penyeimbang PSFTLB beban.	Administrator awan

Sumber daya terkait

- Situs Oracle PeopleSoft
- Dokumentasi AWS

Siapkan pemulihan bencana untuk Oracle JD Edwards dengan EnterpriseOne AWS Elastic Disaster Recovery

Thanigaivel Thirumalai, Amazon Web Services

Ringkasan

Bencana yang dipicu oleh bencana alam, kegagalan aplikasi, atau gangguan layanan merugikan pendapatan dan menyebabkan downtime untuk aplikasi perusahaan. Untuk mengurangi dampak dari peristiwa tersebut, perencanaan pemulihan bencana (DR) sangat penting bagi perusahaan yang mengadopsi sistem perencanaan sumber daya EnterpriseOne perusahaan (ERP) JD Edwards dan perangkat lunak mission-critical dan business critical lainnya.

Pola ini menjelaskan bagaimana bisnis dapat menggunakan AWS Elastic Disaster Recovery sebagai opsi DR untuk aplikasi JD Edwards EnterpriseOne mereka. Ini juga menguraikan langkah-langkah untuk menggunakan failover dan failback Elastic Disaster Recovery untuk membangun strategi DR Lintas wilayah untuk database yang dihosting di instans Amazon Elastic Compute Cloud (Amazon EC2) di AWS Cloud.

> **Note**
>
> Pola ini mengharuskan Wilayah primer dan sekunder untuk implementasi DR Lintas wilayah dihosting di AWS.

Oracle JD Edwards EnterpriseOne adalah solusi perangkat lunak ERP terintegrasi untuk perusahaan menengah hingga besar di berbagai industri.

AWS Elastic Disaster Recovery meminimalkan waktu henti dan kehilangan data dengan pemulihan aplikasi berbasis cloud dan lokal yang cepat dan andal dengan menggunakan penyimpanan yang terjangkau, komputasi minimal, dan pemulihan. point-in-time

AWS menyediakan empat pola arsitektur DR inti. Dokumen ini berfokus pada setup, konfigurasi, dan optimasi dengan menggunakan strategi pilot light. Strategi ini membantu Anda membuat lingkungan DR berbiaya lebih rendah di mana Anda awalnya menyediakan server replikasi untuk mereplikasi data dari database sumber, dan Anda menyediakan server database yang sebenarnya hanya ketika Anda memulai latihan dan pemulihan DR. Strategi ini menghilangkan biaya pemeliharaan server database di Wilayah DR. Sebaliknya, Anda membayar untuk EC2 instance yang lebih kecil yang berfungsi sebagai server replikasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- EnterpriseOne Aplikasi JD Edwards yang berjalan di Oracle Database atau Microsoft SQL Server dengan database yang didukung dalam keadaan berjalan pada instance terkelola. EC2 Aplikasi ini harus mencakup semua komponen EnterpriseOne dasar JD Edwards (Server Perusahaan, Server HTML, dan Server Database) yang diinstal dalam satu Wilayah AWS.
- Peran AWS Identity and Access Management (IAM) untuk menyiapkan layanan Elastic Disaster Recovery.
- Jaringan untuk menjalankan Elastic Disaster Recovery dikonfigurasi sesuai dengan pengaturan konektivitas yang diperlukan.

Batasan

- Anda dapat menggunakan pola ini untuk mereplikasi semua tingkatan, kecuali database di-host di Amazon Relational Database Service (Amazon RDS), dalam hal ini kami menyarankan Anda menggunakan fungsionalitas salinan lintas wilayah Amazon RDS.
- Elastic Disaster Recovery tidak kompatibel dengan CloudEndure Disaster Recovery, tetapi Anda dapat meningkatkan dari CloudEndure Disaster Recovery. Untuk informasi selengkapnya, lihat FAQ di dokumentasi Elastic Disaster Recovery.
- Amazon Elastic Block Store (Amazon EBS) membatasi tingkat di mana Anda dapat mengambil foto. Anda dapat mereplikasi jumlah maksimum 300 server dalam satu akun AWS dengan menggunakan Elastic Disaster Recovery. Untuk mereplikasi lebih banyak server, Anda dapat menggunakan beberapa akun AWS atau beberapa Wilayah AWS target. (Anda harus mengatur Elastic Disaster Recovery secara terpisah untuk setiap akun dan Wilayah.) Untuk informasi selengkapnya, lihat Praktik terbaik dalam dokumentasi Pemulihan Bencana Elastis.
- Beban kerja sumber (EnterpriseOne aplikasi dan database JD Edwards) harus di-host pada instance. EC2 Pola ini tidak mendukung beban kerja yang ada di tempat atau di lingkungan cloud lainnya.
- Pola ini berfokus pada komponen JD Edwards EnterpriseOne . DR lengkap dan rencana kelangsungan bisnis (BCP) harus mencakup layanan inti lainnya, termasuk:
  - Jaringan (cloud pribadi virtual, subnet, dan grup keamanan)
  - Active Directory

- Amazon WorkSpaces
- Penyeimbang Beban Elastis
- Layanan database terkelola seperti Amazon Relational Database Service (Amazon RDS)

Untuk informasi tambahan tentang prasyarat, konfigurasi, dan batasan, lihat dokumentasi Pemulihan Bencana Elastis.

Versi produk

- Oracle JD Edwards EnterpriseOne (versi yang didukung Oracle dan SQL Server berdasarkan Persyaratan Teknis Minimum Oracle)

Arsitektur

Tumpukan teknologi target

- Wilayah tunggal dan cloud pribadi virtual tunggal (VPC) untuk produksi dan non-produksi, dan Wilayah kedua untuk DR
- Zona Ketersediaan Tunggal untuk memastikan latensi rendah antar server
- Application Load Balancer yang mendistribusikan lalu lintas jaringan untuk meningkatkan skalabilitas dan ketersediaan aplikasi Anda di beberapa Availability Zone
- Amazon Route 53 untuk menyediakan konfigurasi Sistem Nama Domain (DNS)
- Amazon WorkSpaces untuk menyediakan pengguna dengan pengalaman desktop di cloud
- Amazon Simple Storage Service (Amazon S3) untuk menyimpan backup, file, dan objek
- Amazon CloudWatch untuk pencatatan aplikasi, pemantauan, dan alarm
- Amazon Elastic Disaster Recovery untuk pemulihan bencana

Arsitektur target

Diagram berikut menunjukkan arsitektur pemulihan bencana lintas wilayah untuk JD Edwards EnterpriseOne menggunakan Elastic Disaster Recovery.

Prosedur

Berikut adalah tinjauan tingkat tinggi dari proses tersebut. Untuk detailnya, lihat bagian Epik.

- Replikasi Elastic Disaster Recovery dimulai dengan sinkronisasi awal. Selama sinkronisasi awal, AWS Replication Agent mereplikasi semua data dari disk sumber ke sumber daya yang sesuai di subnet area pementasan.
- Replikasi berkelanjutan berlanjut tanpa batas waktu setelah sinkronisasi awal selesai.
- Anda meninjau parameter peluncuran, yang mencakup konfigurasi khusus layanan dan templat EC2 peluncuran Amazon, setelah agen diinstal dan replikasi dimulai. Ketika server sumber ditunjukkan sebagai siap untuk pemulihan, Anda dapat memulai instance.
- Saat Elastic Disaster Recovery mengeluarkan serangkaian panggilan API untuk memulai operasi peluncuran, instans pemulihan segera diluncurkan di AWS sesuai dengan pengaturan peluncuran Anda. Layanan ini secara otomatis memutar server konversi selama startup.
- Instans baru diputar di AWS setelah konversi selesai dan siap digunakan. Status server sumber pada saat peluncuran diwakili oleh volume yang terkait dengan instance yang diluncurkan. Proses konversi melibatkan perubahan pada driver, jaringan, dan lisensi sistem operasi untuk memastikan bahwa instans melakukan booting secara native di AWS.
- Setelah peluncuran, volume yang baru dibuat tidak lagi disinkronkan dengan server sumber. AWS Replication Agent terus secara rutin mereplikasi perubahan yang dilakukan pada server sumber Anda ke volume area pementasan, tetapi instance yang diluncurkan tidak mencerminkan perubahan tersebut.
- Saat Anda memulai contoh bor atau pemulihan baru, data selalu tercermin dalam keadaan terbaru yang telah direplikasi dari server sumber ke subnet area pementasan.
- Ketika server sumber ditandai sebagai sedang dipersiapkan untuk pemulihan, Anda dapat memulai instance.

> **Note**
>
> Prosesnya bekerja dua arah: untuk failover dari Wilayah AWS utama ke Wilayah DR, dan gagal kembali ke situs utama, ketika telah dipulihkan. Anda dapat mempersiapkan failback dengan membalikkan arah replikasi data dari mesin target kembali ke mesin sumber dengan cara yang diatur sepenuhnya.

Manfaat dari proses ini yang dijelaskan dalam pola ini meliputi:

- Fleksibilitas: Server replikasi skala dan skala berdasarkan dataset dan waktu replikasi, sehingga Anda dapat melakukan tes DR tanpa mengganggu beban kerja sumber atau replikasi.

- Keandalan: Replikasi kuat, tidak mengganggu, dan berkelanjutan.
- Otomatisasi: Solusi ini menyediakan proses terpadu dan otomatis untuk pengujian, pemulihan, dan kegagalan kembali.
- Optimalisasi biaya: Anda hanya dapat mereplikasi volume yang dibutuhkan dan membayarnya, dan membayar sumber daya komputasi di situs DR hanya ketika sumber daya tersebut diaktifkan. Anda dapat menggunakan instance replikasi yang dioptimalkan biaya (sebaiknya gunakan jenis instans yang dioptimalkan komputasi) untuk beberapa sumber atau satu sumber dengan volume EBS yang besar.

Otomatisasi dan skala

Ketika Anda melakukan pemulihan bencana dalam skala besar, EnterpriseOne server JD Edwards akan memiliki dependensi pada server lain di lingkungan. Contoh:

- Server EnterpriseOne aplikasi JD Edwards yang terhubung ke database yang EnterpriseOne didukung JD Edwards saat boot memiliki dependensi pada database itu.
- EnterpriseOne Server JD Edwards yang memerlukan otentikasi dan perlu terhubung ke pengontrol domain saat boot untuk memulai layanan memiliki ketergantungan pada pengontrol domain.

Untuk alasan ini, kami menyarankan Anda mengotomatiskan tugas failover. Misalnya, Anda dapat menggunakan AWS Lambda atau AWS Step Functions untuk mengotomatiskan skrip EnterpriseOne startup JD Edwards dan perubahan penyeimbang beban untuk mengotomatiskan proses failover. end-to-end Untuk informasi selengkapnya, lihat posting blog Membuat rencana pemulihan bencana yang dapat diskalakan dengan AWS Elastic Disaster Recovery.

Alat

Layanan AWS

- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans. EC2
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Elastic Disaster Recovery meminimalkan waktu henti dan kehilangan data dengan pemulihan aplikasi berbasis cloud dan lokal yang cepat dan andal menggunakan penyimpanan yang terjangkau, komputasi minimal, dan pemulihan. point-in-time

- Amazon Virtual Private Cloud (Amazon VPC) memberi Anda kontrol penuh atas lingkungan jaringan virtual Anda, termasuk penempatan sumber daya, konektivitas, dan keamanan.

Praktik terbaik

Praktik terbaik umum

- Miliki rencana tertulis tentang apa yang harus dilakukan jika terjadi peristiwa pemulihan yang nyata.
- Setelah Anda mengatur Elastic Disaster Recovery dengan benar, buat CloudFormation template AWS yang dapat membuat konfigurasi sesuai permintaan, jika diperlukan. Tentukan urutan server dan aplikasi yang harus diluncurkan, dan catat ini dalam rencana pemulihan.
- Lakukan latihan biasa (EC2 tarif Amazon standar berlaku).
- Pantau kesehatan replikasi yang sedang berlangsung dengan menggunakan konsol Elastic Disaster Recovery atau secara terprogram.
- Lindungi point-in-time snapshot dan konfirmasikan sebelum menghentikan instance.
- Buat peran IAM untuk instalasi AWS Replication Agent.
- Aktifkan perlindungan terminasi untuk instance pemulihan dalam skenario DR nyata.
- Jangan gunakan tindakan Putuskan sambungan dari AWS di konsol Elastic Disaster Recovery untuk server tempat Anda meluncurkan instans pemulihan, bahkan dalam kasus peristiwa pemulihan nyata. Melakukan pemutusan akan menghentikan semua sumber daya replikasi yang terkait dengan server sumber ini, termasuk titik pemulihan point-in-time (PIT) Anda.
- Ubah kebijakan PIT untuk mengubah jumlah hari penyimpanan snapshot.
- Edit template peluncuran di pengaturan peluncuran Elastic Disaster Recovery untuk mengatur subnet, grup keamanan, dan jenis instans yang benar untuk server target Anda.
- Otomatiskan proses end-to-end failover dengan menggunakan Lambda atau Step Functions untuk mengotomatiskan skrip startup JD Edwards EnterpriseOne dan perubahan penyeimbang beban.

EnterpriseOne Optimalisasi dan pertimbangan JD Edwards

- Pindah PrintQueueke database.
- Pindah MediaObjectske database.
- Kecualikan log dan folder temp dari server batch dan logika.
- Kecualikan folder temp dari WebLogic Oracle.
- Buat skrip untuk startup setelah failover.

- Kecualikan tempdb untuk SQL Server.
- Kecualikan file temp untuk Oracle.

Epik

Lakukan tugas dan konfigurasi awal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan jaringan replikasi.	Terapkan EnterpriseOne sistem JD Edwards Anda di Wilayah AWS utama dan identifikasi Wilayah AWS untuk DR. Ikuti langkah-l angkah di bagian persyarat an jaringan Replikasi pada dokumentasi Pemulihan Bencana Elastis untuk merencanakan dan menyiapka n replikasi dan jaringan DR Anda.	Administrator AWS
Tentukan RPO dan RTO.	Identifikasi tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO) untuk server aplikasi dan database Anda.	Arsitek awan, arsitek DR
Aktifkan replikasi untuk Amazon EFS.	Jika berlaku, aktifkan replikasi dari AWS primer ke Wilayah DR untuk sistem file bersama seperti Amazon Elastic File System (Amazon EFS) dengan menggunakan AWS DataSync, rsync, atau alat lain yang sesuai.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kelola DNS dalam kasus DR.	Identifikasi proses untuk memperbarui Sistem Nama Domain (DNS) selama latihan DR atau DR.	Administrator awan
Buat peran IAM untuk penyiapan.	Ikuti petunjuk di bagian inisialisasi dan izin Pemulihan Bencana Elastis pada dokumentasi Pemulihan Bencana Elastis untuk membuat peran IAM guna menginisialisasi dan mengelola layanan AWS.	Administrator awan
Siapkan pengintip VPC.	Pastikan bahwa sumber dan target VPCs diintip dan dapat diakses satu sama lain. Untuk petunjuk konfigurasi, lihat dokumentasi Amazon VPC.	Administrator AWS

Konfigurasikan pengaturan replikasi Pemulihan Bencana Elastis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi Pemulihan Bencana Elastis.	Buka konsol Elastic Disaster Recovery, pilih Wilayah AWS target (tempat Anda akan mereplikasi data dan meluncurkan instans pemulihan), lalu pilih Setel pengaturan replikasi default.	Administrator AWS
Siapkan server replikasi.	1. Di panel Siapkan server replikasi, masukkan subnet area pementasan dan tipe	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	instance server replikasi. Pilih tipe instans `t3.small` yang dipilih secara default. Konfigurasikan setelan ini berdasarkan kebutuhan Anda, dan ingatlah untuk mempertimbangkan harga instans. Untuk informasi selengkapnya, lihat EC2 harga Amazon. 2. Di bagian Akses layanan, pilih Lihat detail untuk meninjau peran terkait layanan dan kebijakan tambahan yang dibuat selama inisialisasi layanan. 3. Pilih Berikutnya.	
Konfigurasikan volume dan grup keamanan.	1. Di panel Volume dan grup keamanan, pilih jenis volume EBS untuk server replikasi dan setel enkripsi Amazon EBS ke Default. 2. Pilih Selalu gunakan grup keamanan AWS Elastic Disaster Recovery sehingga Elastic Disaster Recovery secara otomatis melampirkan dan memantau grup keamanan default. 3. Pilih Berikutnya.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan tambahan.	1. Di panel Pengaturan tambahan, konfigurasikan perutean dan pelambata n data, kebijakan PIT, dan tag. • Data routing dan throttlin g mengontrol bagaimana data mengalir dari server eksternal ke server replikasi. Pilih Gunakan IP pribadi untuk replikasi data. Jika tidak, server replikasi akan secara otomatis diberi IP publik, dan data akan mengalir melalui internet publik. • Di bagian kebijakan Point in time (PIT), konfigura sikan kebijakan retensi yang menentukan durasi setelah snapshot tidak diperlukan. Periode penyimpanan default adalah tujuh hari. • Di bagian Tag, tambahkan tag khusus ke sumber daya yang dibuat oleh Elastic Disaster Recovery di akun AWS Anda. 2. Pilih Berikutnya, tinjau pengaturan di panel berikutnya, lalu pilih Buat	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	default untuk membuat templat default.	

Instal Agen Replikasi AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM.	Buat peran IAM yang berisi `AWSElasticDisasterRecoveryAgentInstallationPolicy` kebijakan. Di bagian Select AWS Access Type, aktifkan akses terprogram. Perhatikan ID kunci akses dan kunci akses rahasia. Anda akan memerlukan informasi ini selama penginstalan AWS Replication Agent.	Administrator AWS
Periksa persyaratan.	Periksa dan lengkapi prasyarat dalam dokumentasi Elastic Disaster Recovery untuk menginstal AWS Replication Agent.	Administrator AWS
Instal AWS Replication Agent.	Ikuti instruksi penginstalan untuk sistem operasi Anda dan instal AWS Replication Agent. • Untuk Microsoft Windows: Unduh file pengaturan dan jalankan file.exe sebagai administrator. Tanggapi	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	petunjuk untuk menyelesa ikan instalasi.
	• Untuk Linux: Salin perintah berikut (dalam urutan yang disajikan) dan tempelkan ke sesi Secure Shell (SSH) Anda. Perintah pertama mengunduh penginstal dan perintah kedua menjalank annya.
	`wget -O ./aws-replication-installer-init.py https://aws-elastic-disaster-recovery-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py`
	`sudo python3 aws-replication-installer-init.py`
	Tanggapi petunjuk untuk menyelesaikan instalasi.
	Note
	Ubah URL untuk mencerminkan Wilayah Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Ulangi langkah-langkah ini untuk server yang tersisa.	
Pantau replikasi.	Kembali ke panel server Elastic Disaster Recovery Source untuk memantau status replikasi. Sinkronisasi awal akan memakan waktu tergantung pada ukuran transfer data. Ketika server sumber sepenuhnya disinkronkan, status server akan diperbaru i ke Siap. Ini berarti bahwa server replikasi telah dibuat di area pementasan, dan volume EBS telah direplika si dari server sumber ke area pementasan.	Administrator AWS

Mengonfigurasi pengaturan peluncuran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Edit pengaturan peluncuran.	Untuk memperbarui pengatura n peluncuran untuk instance drill dan recovery, pada konsol Elastic Disaster Recovery, pilih server sumber, lalu pilih Actions, Edit launch settings. Atau Anda dapat memilih mesin sumber replikasi Anda dari halaman server Sumber, lalu pilih tab Launch Settings.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tab ini memiliki dua bagian: Pengaturan peluncuran umum dan templat EC2 peluncuran.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan peluncuran umum.	Merevisi pengaturan peluncuran umum sesuai dengan kebutuhan Anda. • Ukuran kanan tipe instans: Jika Anda memilih Basic, Elastic Disaster Recovery melewati jenis instans yang Anda pilih di template EC2 peluncuran Amazon dan secara otomatis memilih jenis instans berdasarkan sistem operasi, CPU, dan RAM server sumber. • Salin IP pribadi: Pilih apakah Anda ingin Elastic Disaster Recovery untuk memastikan bahwa IP pribadi yang digunakan oleh instance bor atau pemulihan cocok dengan IP pribadi yang digunakan oleh server sumber. Jika Anda memilih Ya, pastikan rentang IP subnet yang Anda tetapkan di template EC2 peluncura n Amazon menyertakan alamat IP pribadi. Untuk informasi selengkapnya, lihat Pengaturan peluncuran umum di dokumentasi Elastic Disaster Recovery.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan template EC2 peluncuran Amazon.	Elastic Disaster Recovery menggunakan template EC2 peluncuran Amazon untuk meluncurkan instans bor dan pemulihan untuk setiap server sumber. Template peluncura n dibuat secara otomatis untuk setiap server sumber yang Anda tambahkan ke Elastic Disaster Recovery setelah Anda menginstal AWS Replication Agent. Anda harus mengatur template EC2 peluncuran Amazon sebagai template peluncuran default jika Anda ingin menggunakannya dengan Elastic Disaster Recovery. Untuk informasi selengkap nya, lihat EC2 Luncurkan Template di dokumentasi Elastic Disaster Recovery.	Administrator AWS

Memulai bor DR dan failover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai Bor	1. Pada konsol Elastic Disaster Recovery, buka halaman server Sumber dan verifikasi bahwa status server sumber sudah Siap.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pilih semua server sumber yang ingin Anda lakukan latihan DR. 3. Dari menu Initiate recovery job, pilih Initiate drill dan pilih snapshot yang sesuai point-in-time. Ini memulai pekerjaan pemulihan untuk server sumber yang dipilih. Anda dapat memantau status pekerjaan di tab Riwayat pekerjaan pemulihan. Instans bor yang diluncurkan juga muncul di halaman instans Pemulihan. **Note** Perubahan lebih lanjut pada server sumber akan disinkronkan ke server replikasi, bukan ke instance bor. 4. Uji dan verifikasi instance bor DR. 5. Pada halaman instans Pemulihan, pilih instans bor, lalu pilih Tindakan, Putuskan sambungan dari AWS. Ini menghapus	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS Replication Agent dari instans pemulihan dan menghapus semua sumber daya yang terkait dengan instans pemulihan dari Elastic Disaster Recovery. 6. Pilih Hapus instans pemulihan. Ini menghapus representasi instance dari konsol Elastic Disaster Recovery dan sepenuhny a memisahkan instance dari layanan Elastic Disaster Recovery. Itu tidak menghapus EC2 instance yang mendasarinya. 7. Hentikan instans bor DR dari EC2 konsol Amazon. Untuk informasi selengkapnya, lihat Mempersiapkan failover dalam dokumentasi Elastic Disaster Recovery.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi bor.	Pada langkah sebelumnya, Anda meluncurkan instance target baru di Wilayah DR. Instance target adalah replika server sumber berdasarkan snapshot yang diambil saat Anda memulai peluncuran. Dalam prosedur ini, Anda terhubung ke mesin EC2 target Amazon Anda untuk mengonfirmasi bahwa mereka berjalan seperti yang diharapkan. 1. Buka EC2 konsol Amazon. 2. Pilih Instans (berjalan). 3. Pilih instance target dan catat IPv4 alamat pribadiny a. 4. Pastikan Anda dapat terhubung ke EC2 instance dan JD Edwards Enterpris eOne dan komponen terkait direplikasi seperti yang diharapkan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai failover.	Failover adalah pengalihan lalu lintas dari sistem primer ke sistem sekunder. Elastic Disaster Recovery membantu Anda melakukan failover dengan meluncurkan instans pemulihan di AWS. Ketika instans pemulihan telah diluncurkan, Anda mengarahkan lalu lintas dari sistem utama Anda ke instans ini. 1. Pada konsol Elastic Disaster Recovery, buka halaman server Sumber dan verifikasi bahwa kolom Siap untuk pemulihan untuk server sumber menunjukkan Siap, dan kolom status replikasi data menunjukkan Sehat. 2. Pilih server sumber. Dari menu Initiate recovery job, pilih Initiate recovery. 3. Pilih point-in-time snapshot untuk meluncurkan instance pemulihan, lalu pilih Memulai pemulihan. Ini memulai pekerjaan pemulihan. Anda dapat memantau status pekerjaan di halaman instans Pemulihan.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Uji dan verifikasi instance pemulihan. Jika diperlukan, sesuaikan konfigurasi DNS dan hubungkan Enterpris eOne aplikasi JD Edwards Anda ke database. 5. Anda sekarang dapat memutuskan dan menonaktifkan Enterpris eOne server sumber JD Edwards, karena semua perubahan telah ditulis ke instance pemulihan baru. 6. Daftarkan instans pemulihan sebagai server sumber di Wilayah DR dengan mengikuti proses yang dijelaskan dalam epik Install the AWS Replication Agent. Untuk informasi selengkapnya, lihat Melakukan failover dalam dokumentasi Elastic Disaster Recovery.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai kegagalan kembali.	Proses untuk memulai failback mirip dengan proses untuk memulai failover. 1. Buka konsol Elastic Disaster Recovery di Wilayah utama. Arahkan ke halaman instans Pemulihan , pilih instans bor, lalu pilih Tindakan, Putuskan sambungan dari AWS, Hapus instans pemulihan. 2. Buka konsol Elastic Disaster Recovery di Wilayah DR. Daftarkan server JD Edwards baru Anda sebagai Enterpris eOne server sumber di Wilayah DR dengan menginstal AWS Replicati on Agent. Data akan disinkronkan dengan server replikasi baru yang disediakan di subnet pementasan baru. **Note** Ketika server JD Edwards baru terdaftar sebagai EnterpriseOne server sumber, Anda mungkin melihat dua server	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber di konsol Elastic Disaster Recovery: satu server yang dibuat dari EC2 instance utama, dan server baru yang dibuat dari instance pemulihan. Kami menyarankan Anda menandai server dengan benar untuk menghindari kebingungan, dan sebaiknya tambahkan server baru ke template peluncuran. 3. Untuk memulai ulang replikasi DR dari Wilayah utama, lepaskan instans pemulihan yang diluncurkan dari konsol Elastic Disaster Recovery di Wilayah DR, dan daftarkan host sebagai server sumber di Wilayah utama. Untuk informasi selengkapnya, lihat Melakukan failback dalam dokumentasi Elastic Disaster Recovery.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai komponen JD Edwards EnterpriseOne .	1. Mulai EnterpriseOne database JD Edwards dengan masuk ke server database. 2. Saat database berjalan, mulai EnterpriseOne logika JD Edwards dan server batch. 3. Mulai WebLogic di server web, dan mulai instance JAS di server JAS. 4. Mulai WebLogic di server penyediaan dan di server untuk konsol SM. 5. Mulai SM Agent di server. 6. Konfirmasikan bahwa login ke JD Edwards Enterpris eOne berfungsi dengan benar. Anda harus memasukkan perubahan di Route 53 dan Application Load Balancer agar tautan JD Edwards berfungsi EnterpriseOne . Anda dapat mengotomatiskan langkah-langkah ini dengan menggunakan Lambda, Step Functions, dan Systems Manager (Run Command).	JD Edwards CNC Enterpris eOne

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Elastic Disaster Recovery melakukan replikasi tingkat blok dari volume EBS EC2 instance sumber yang menampung sistem operasi dan sistem file. Sistem file bersama yang dibuat dengan menggunakan Amazon EFS bukan bagian dari replikasi ini. Anda dapat mereplikasi sistem file bersama ke Wilayah DR dengan menggunakan AWS DataSync, seperti yang dicatat dalam epik pertama, dan kemudian memasang sistem file yang direplikasi ini di sistem DR.	

Pemecahan masalah

Isu	Solusi
Status replikasi data server sumber terhenti dan replikasi tertinggal. Jika Anda memeriksa	Periksa untuk mengonfirmasi bahwa server sumber yang macet sedang berjalan.

Isu	Solusi
detail, status replikasi data menampilkan Agen tidak terlihat.	**Note** Jika server sumber turun, server replikasi secara otomatis dihentikan. Untuk informasi selengkapnya tentang masalah lag, lihat Masalah lag replikasi dalam dokumentasi Elastic Disaster Recovery.
Instalasi AWS Replication Agent dalam EC2 instance sumber gagal di RHEL 8.2 setelah memindai disk. `aws_replication_agent_installer.log` mengungkapkan bahwa header kernel hilang.	Sebelum Anda menginstal AWS Replication Agent di RHEL 8, CentOS 8, atau Oracle Linux 8, jalankan: `sudo yum install elfutils-libelf-devel` Untuk informasi selengkapnya, lihat persyaratan instalasi Linux dalam dokumentasi Elastic Disaster Recovery.
Pada konsol Elastic Disaster Recovery, Anda melihat server sumber sebagai Siap dengan lag dan status replikasi data sebagai Stalled. Bergantung pada berapa lama Agen Replikasi AWS tidak tersedia, statusnya mungkin menunjukkan kelambatan yang tinggi, tetapi masalahnya tetap sama.	Gunakan perintah sistem operasi untuk mengonfirmasi bahwa AWS Replication Agent berjalan di EC2 instance sumber, atau mengonfirmasi bahwa instans sedang berjalan. Setelah Anda memperbaiki masalah apa pun, Elastic Disaster Recovery akan memulai kembali pemindaian. Tunggu hingga semua data telah disinkronkan dan status replikasi Sehat sebelum Anda memulai latihan DR.

Isu	Solusi
Replikasi awal dengan lag tinggi. Pada konsol Elastic Disaster Recovery, Anda dapat melihat bahwa status sinkronisasi awal sangat lambat untuk server sumber.	Periksa masalah kelambatan replikasi yang didokumentasikan di bagian Masalah lag replikasi pada dokumentasi Elastic Disaster Recovery. Server replikasi mungkin tidak dapat menangani beban karena operasi komputasi intrinsik. Jika demikian, coba tingkatkan jenis instans setelah berkonsultasi dengan tim AWS Technical Support.

Sumber daya terkait

- Panduan Pengguna AWS Elastic Disaster Recovery
- Membuat rencana pemulihan bencana yang dapat diskalakan dengan AWS Elastic Disaster Recovery (postingan blog AWS)
- AWS Elastic Disaster Recovery - Pengantar Teknis (kursus AWS Skill Builder; memerlukan login)
- Panduan memulai cepat AWS Elastic Disaster Recovery

Mengatur deteksi CloudFormation drift di organisasi multi-wilayah dan multi-akun

Ram Kandaswamy, Amazon Web Services

Ringkasan

Pengguna Amazon Web Services (AWS) sering mencari cara yang efisien untuk mendeteksi ketidakcocokan konfigurasi sumber daya, termasuk drift dalam AWS CloudFormation tumpukan, dan memperbaikinya sesegera mungkin. Ini terutama terjadi ketika AWS Control Tower digunakan.

Pola ini memberikan solusi preskriptif yang secara efisien memecahkan masalah dengan menggunakan perubahan konfigurasi sumber daya terkonsolidasi dan bertindak atas perubahan tersebut untuk menghasilkan hasil. Solusi ini dirancang untuk skenario di mana ada beberapa CloudFormation tumpukan yang dibuat di lebih dari satu Wilayah AWS, atau di lebih dari satu akun, atau kombinasi keduanya. Tujuan solusinya adalah sebagai berikut:

- Sederhanakan proses deteksi drift
- Siapkan notifikasi dan peringatan
- Menyiapkan pelaporan konsolidasi

Prasyarat dan batasan

Prasyarat

- AWS Config diaktifkan di semua Wilayah dan akun yang harus dipantau

Batasan

- Laporan yang dihasilkan hanya mendukung format output nilai yang dipisahkan koma (CSV) dan JSON.

Arsitektur

Diagram berikut menunjukkan AWS Organizations pengaturan dengan beberapa akun. AWS Config aturan berkomunikasi antar akun.

Alur kerja mencakup langkah-langkah berikut:

1. AWS Config Aturan mendeteksi penyimpangan.
2. Hasil deteksi drift yang ditemukan di akun lain dikirim ke akun manajemen.
3. CloudWatch Aturan Amazon memanggil AWS Lambda fungsi.
4. Fungsi Lambda menanyakan AWS Config aturan untuk hasil agregat.
5. Fungsi Lambda memberi tahu Amazon Simple Notification Service (Amazon SNS), yang mengirimkan pemberitahuan email tentang drift.

Otomatisasi dan skala

Solusi yang disajikan di sini dapat menskalakan untuk Wilayah dan akun tambahan.

Alat

Layanan AWS

- AWS Configmemberikan tampilan rinci tentang konfigurasi AWS sumber daya di Anda Akun AWS. Ini mencakup bagaimana sumber daya terkait satu sama lain dan bagaimana sumber daya tersebut dikonfigurasi di masa lalu sehingga Anda dapat melihat bagaimana konfigurasi dan hubungan berubah dari waktu ke waktu.
- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

Epik

Mengotomatiskan deteksi drift untuk CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat agregator.	1. Masuk ke Konsol Manajemen AWS dan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	buka AWS Config konsol di https://console.aws.amazon.com/config. 2. Buat agregator di akun manajemen. 3. Pastikan replikasi data diaktifkan sehingga AWS Config dapat mengambil data dari akun sumber. 4. Pilih semua Wilayah dan akun yang berlaku. Anda dapat memilih akun berdasarkan AWS Organizations. Kami merekomendasikan pendekatan ini karena akun baru dalam organisasi secara otomatis merupakan bagian dari agregator.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan AWS terkelola.	Tambahkan aturan `cloudformation-stack-drift-detection-check` AWS terkelola. Aturan membutuhkan satu nilai parameter:`cloudformationArn` . Masukkan peran IAM Amazon Resource Name (ARN) yang memiliki izin untuk mendeteksi stack drift. Peran harus memiliki kebijakan kepercayaan yang memungkinkan AWS Config untuk mengambil peran.	Arsitek awan
Buat bagian kueri lanjutan dari agregator.	Untuk mengambil tumpukan hanyut dari berbagai sumber, buat kueri berikut: `SELECT resourceId, configuration.driftInformation.stackDriftStatus WHERE resourceType = 'AWS::CloudFormation::Stack' AND configuration.driftInformation.stackDriftStatus IN ('DRIFTED')`	Arsitek awan, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengotomatiskan menjalankan kueri dan mempublikasikan.	1. Buat fungsi Lambda dengan menggunakan kode yang dilampirkan. Lambda akan mempublikasikan hasilnya ke topik SNS yang disediakan sebagai variabel lingkungan dalam fungsi Lambda. 2. Untuk menerima peringatan, buat langganan email ke topik SNS.	Arsitek awan, Pengembang
Buat CloudWatch aturan.	Buat CloudWatch aturan berbasis jadwal untuk memanggil fungsi Lambda, yang bertanggung jawab untuk memperingatkan.	Arsitek awan

Sumber daya terkait

Sumber

- Apa itu AWS Config?
- Agregasi data multi-wilayah multi-akun
- Mendeteksi perubahan konfigurasi yang tidak terkelola pada tumpukan dan sumber daya
- IAM: Lulus peran IAM ke peran tertentu Layanan AWS
- Apa itu Amazon SNS?

Informasi tambahan

Pertimbangan-pertimbangan

Sebaiknya gunakan solusi yang disajikan dalam pola ini daripada menggunakan solusi khusus yang melibatkan panggilan API pada interval tertentu untuk memulai deteksi drift pada setiap

kumpulan CloudFormation tumpukan atau tumpukan. Solusi khusus yang menggunakan panggilan API pada interval tertentu dapat menyebabkan sejumlah besar panggilan API dan memengaruhi kinerja. Karena jumlah panggilan API, pelambatan dapat terjadi. Masalah potensial lainnya adalah keterlambatan deteksi jika perubahan sumber daya diidentifikasi hanya berdasarkan jadwal.

Karena set tumpukan terbuat dari tumpukan, Anda dapat menggunakan solusi ini. Detail instance stack juga tersedia sebagai bagian dari solusi.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Berhasil mengimpor bucket S3 sebagai tumpukan AWS CloudFormation

Ram Kandaswamy, Amazon Web Services

Ringkasan

Jika Anda menggunakan sumber daya Amazon Web Services (AWS), seperti bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), dan ingin menggunakan pendekatan infrastruktur sebagai kode (IAc), Anda dapat mengimpor sumber daya ke CloudFormation AWS dan mengelolanya sebagai tumpukan.

Pola ini memberikan langkah-langkah untuk berhasil mengimpor bucket S3 sebagai CloudFormation tumpukan AWS. Dengan menggunakan pendekatan pola ini, Anda dapat menghindari kemungkinan kesalahan yang mungkin terjadi jika Anda mengimpor bucket S3 dalam satu tindakan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Bucket S3 dan kebijakan bucket S3 yang ada. Untuk informasi selengkapnya tentang ini, lihat Kebijakan bucket S3 apa yang harus saya gunakan untuk mematuhi aturan AWS Config s3-bucket-ssl-requests-only di Pusat Pengetahuan AWS.
- Kunci AWS Key Management Service (AWS KMS) yang ada dan aliasnya. Untuk informasi selengkapnya tentang ini, lihat Bekerja dengan alias dalam dokumentasi AWS KMS.
- Contoh CloudFormation template `CloudFormation-template-S3-bucket` AWS (terlampir), diunduh ke komputer lokal Anda.

Arsitektur

Diagram menunjukkan alur kerja berikut:

1. Pengguna membuat template AWS berformat JSON atau YAML. CloudFormation
2. Template membuat CloudFormation tumpukan AWS untuk mengimpor bucket S3.
3. CloudFormation Tumpukan AWS mengelola bucket S3 yang Anda tentukan dalam template.

Tumpukan teknologi

- AWS CloudFormation
- AWS Identity and Access Management (IAM)
- AWS KMS
- Amazon S3

Alat

- AWS CloudFormation — AWS CloudFormation membantu Anda membuat dan menyediakan penerapan infrastruktur AWS secara dapat diprediksi dan berulang kali.
- AWS Identity and Access Management (IAM) — IAM adalah layanan web untuk mengontrol akses ke layanan AWS secara aman.
- AWS KMS — AWS Key Management Service (AWS KMS) adalah layanan enkripsi dan manajemen kunci yang diskalakan untuk cloud.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk Internet.

Epik

Impor bucket S3 dengan enkripsi AWS KMS key berbasis sebagai tumpukan AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat template untuk mengimpor bucket S3 dan kunci KMS.	Di komputer lokal Anda, buat template untuk mengimpor bucket S3 dan kunci KMS Anda dengan menggunakan contoh template berikut:
	`AWSTemplateFormatVersion: 2010-09-09`
	`Parameters:`
	`  bucketName:`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> Type: String
	
	Resources:
	
	  S3Bucket:
	
	    Type: 'AWS::S3:
	:Bucket'
	
	    DeletionPolicy:
	 Retain
	
	    Properties:
	
	      BucketName: !Ref
	 bucketName
	
	      BucketEncryption:
	
	        ServerSid
	eEncryptionConfigu
	ration:
	
	          - ServerSid
	eEncryptionByDefault:
	
	              SSEAlgori
	thm: 'aws:kms'
	
	              KMSMaster
	KeyID: !GetAtt
	
	                -
	 KMSS3Encryption
	
	                - Arn
	
	  KMSS3Encryption:
	
	    Type: 'AWS::KMS
	::Key'</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> DeletionPolicy: Retain
	   Properties:
	     Enabled: true
	     KeyPolicy: !Sub |-
	       {
	           "Id": "key-consolepolicy-3",
	           "Version": "2012-10-17",
	           "Statement": [
	               {
	"Sid": "Enable IAM User Permissions",
	"Effect": "Allow",
	"Principal": {
	"AWS": ["arn:aws:iam::${AWS::AccountId}:root"]
	                   },</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Action": "kms:*",
	
	"Resource": "*"
	}
	}
	]
	}
	
	EnableKey
	Rotation: true</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tumpukan.	1. Masuk ke AWS Management Console, buka CloudFormation konsol AWS, pilih Lihat tumpukan, pilih Buat tumpukan, lalu pilih Dengan sumber daya yang ada (sumber daya impor). 2. Pilih Upload file template dan kemudian upload file template yang Anda buat sebelumnya. 3. Masukkan nama untuk tumpukan Anda dan konfigurasikan opsi yang tersisa sesuai dengan kebutuhan Anda. 4. Pilih Buat tumpukan dan tunggu status tumpukan berubah menjadi`IMPORT_COMPLETE` .	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat alias kunci KMS.	1. Di CloudFormation konsol AWS, pilih Stacks, pilih nama tumpukan yang Anda buat sebelumnya, pilih panel Template, lalu pilih View in Designer.
	2. Tambahkan cuplikan berikut ke `Resource` bagian template Anda, lalu pilih Buat tumpukan dan lengkapi wizard:
	
	`KMSS3EncryptionAlias:`
	
	`  Type: 'AWS::KMS ::Alias'`
	
	`  DeletionPolicy: Retain`
	
	`  Properties:`
	
	`  AliasName: alias/ S3BucketKey`
	
	`  TargetKeyId: !Ref KMSS3Encryption`
	
	Untuk informasi selengkapnya tentang ini, lihat pembaruan AWS CloudFormation stack di CloudFormation dokumentasi AWS.
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui tumpukan untuk menyertakan kebijakan bucket S3.	1. Di CloudFormation konsol AWS, pilih Stacks, pilih nama tumpukan yang Anda buat sebelumnya, pilih panel Template, lalu pilih View in Designer.
	2. Tambahkan cuplikan berikut ke `Resource` bagian template, lalu pilih Buat tumpukan dan lengkapi wizard:
	
	`S3BucketPolicy:`
	
	`  Type: 'AWS::S3::BucketPolicy'`
	
	`  Properties:`
	
	`    Bucket: !Ref S3Bucket`
	
	`    PolicyDocument: !Sub |-`
	
	`      {`
	
	`  "Version": "2008-10-17",`
	
	`                  "Id": "restricthttp",`
	
	`  "Statement": [`
	
	`                      {`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Sid": "denyhttp",
	"Effect": "Deny",
	"Principal": {
	"AWS": "*"
	},
	"Action": "s3:*",
	"Resource": ["arn:aws
	:s3:::${S3Bucket}"
	,"arn:aws:s3:::${S
	3Bucket}/*"],
	"Condition": {
	"Bool": {
	"aws:Secu
	reTransport": "false"
	}
	}
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>]
	
	}</pre>
	Note
	Kebijakan bucket S3 ini memiliki pernyataan penolakan yang membatasi panggilan API yang tidak aman.	

Perbarui kebijakan utama.	1. Di CloudFormation konsol AWS, pilih Stacks, pilih nama tumpukan yang Anda buat sebelumnya, pilih panel Template, lalu pilih View in Designer. 2. Ubah sumber daya KMS template untuk menyertakan kebijakan kunci yang memungkinkan administrator mengelola kunci KMS. 3. Pilih Buat tumpukan, pilih Berikutnya, dan kemudian lengkapi wizard sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya, lihat Kebijakan utama AWS KMS dalam dokumentasi AWS KMS.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan tag tingkat sumber daya.	1. Di CloudFormation konsol AWS, pilih Stacks, pilih nama tumpukan yang Anda buat sebelumnya, pilih panel Template, lalu pilih View in Designer.
	2. Tambahkan cuplikan berikut ke bagian `Properties` sumber daya Amazon S3 pada template, lalu pilih Buat tumpukan dan lengkapi wizard:
	`Tags:`
	`  - Key: createdBy`
	`  Value: Cloudformation`
		AWS DevOps

Sumber daya terkait

- Membawa sumber daya yang ada ke dalam CloudFormation manajemen AWS
- AWS re:Invent 2017: Menyelam jauh di CloudFormation AWS (video)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Sinkronisasi data antara sistem file Amazon EFS di Wilayah AWS yang berbeda dengan menggunakan AWS DataSync

Sarat Chandra Pothula dan Aditya Ambati, Amazon Web Services

Ringkasan

Solusi ini menyediakan kerangka kerja yang kuat untuk sinkronisasi data yang efisien dan aman antara instans Amazon Elastic File System (Amazon EFS) di Wilayah AWS yang berbeda. Pendekatan ini dapat diskalakan dan menyediakan replikasi data lintas wilayah yang terkontrol. Solusi ini dapat meningkatkan pemulihan bencana dan strategi redundansi data Anda.

Dengan menggunakan AWS Cloud Development Kit (AWS CDK), pola ini digunakan sebagai pendekatan infrastruktur sebagai kode (IAc) untuk menerapkan sumber daya solusi. Aplikasi AWS CDK menerapkan sumber daya AWS, DataSync Amazon EFS, Amazon Virtual Private Cloud (Amazon VPC), dan Amazon Elastic Compute Cloud (Amazon) yang penting. EC2 IAc ini menyediakan proses penerapan yang dapat diulang dan dikontrol versi yang sepenuhnya selaras dengan praktik terbaik AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI) versi 2.9.11 atau yang lebih baru, diinstal dan dikonfigurasi
- AWS CDK versi 2.114.1 atau yang lebih baru, diinstal dan di-bootstrap
- NodeJS versi 20.8.0 atau yang lebih baru, diinstal

Batasan

- Solusi ini mewarisi batasan dari DataSync dan Amazon EFS, seperti kecepatan transfer data, batasan ukuran, dan ketersediaan regional. Untuk informasi selengkapnya, lihat DataSync kuota AWS dan kuota Amazon EFS.
- Solusi ini hanya mendukung Amazon EFS. DataSync mendukung layanan AWS lainnya, seperti Amazon Simple Storage Service (Amazon S3) dan FSx Amazon for Lustre. Namun, solusi ini memerlukan modifikasi untuk menyinkronkan data dengan layanan lain ini.

Arsitektur

Solusi ini menerapkan tumpukan AWS CDK berikut:

- Tumpukan VPC Amazon — Tumpukan ini menyiapkan sumber daya virtual private cloud (VPC), termasuk subnet, gateway internet, dan gateway NAT di Wilayah AWS primer dan sekunder.
- Tumpukan Amazon EFS — Tumpukan ini menyebarkan sistem file Amazon EFS ke Wilayah primer dan sekunder dan menghubungkannya ke masing-masing VPCs.
- EC2 Tumpukan Amazon — Tumpukan ini meluncurkan EC2 instance di Wilayah primer dan sekunder. Instans ini dikonfigurasi untuk memasang sistem file Amazon EFS, yang memungkinkan mereka mengakses penyimpanan bersama.
- DataSync tumpukan lokasi — Tumpukan ini menggunakan konstruksi khusus yang dipanggil `DataSyncLocationConstruct` untuk membuat sumber daya DataSync lokasi di Wilayah primer dan sekunder. Sumber daya ini menentukan titik akhir untuk sinkronisasi data.
- DataSync tumpukan tugas - Tumpukan ini menggunakan konstruksi khusus yang dipanggil `DataSyncTaskConstruct` untuk membuat DataSync tugas di Wilayah utama. Tugas ini dikonfigurasi untuk menyinkronkan data antara Wilayah primer dan sekunder dengan menggunakan lokasi DataSync sumber dan tujuan.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS DataSync adalah layanan transfer dan penemuan data online yang membantu Anda memindahkan file atau data objek ke, dari, dan di antara layanan penyimpanan AWS.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di AWS Cloud.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang

akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Repositori kode

Kode untuk pola ini tersedia di repositori DataSync Proyek Lintas Wilayah GitHub Amazon EFS.

Praktik terbaik

Ikuti praktik terbaik yang dijelaskan dalam Praktik terbaik untuk menggunakan AWS CDK TypeScript untuk membuat proyek IAc.

Epik

Menerapkan aplikasi AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori proyek.	Masukkan perintah berikut untuk mengkloning repositori DataSync Proyek Lintas Wilayah Amazon EFS. `git clone https://github.com/aws-samples/aws-efs-crossregion-datasync.git`	AWS DevOps
Instal dependensi npm.	Masukkan perintah berikut. `npm ci`	AWS DevOps
Pilih Wilayah primer dan sekunder.	Di repositori kloning, arahkan ke direktori. `src/infra` Dalam `Launcher.ts` file, perbarui `PRIMARY_AWS_REGION` dan `SECONDARY_AWS_REGION` nilai. Gunakan kode Wilayah yang sesuai.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`const primaryRegion = { account: account, region: '<PRIMARY_AWS_REGION>' }; const secondaryRegion = { account: account, region: '<SECONDARY_AWS_REGION>' };`	
Bootstrap lingkungan.	Masukkan perintah berikut untuk mem-bootstrap akun AWS dan Wilayah AWS yang ingin Anda gunakan. `cdk bootstrap <aws_account>/<aws_region>` Untuk informasi selengkapnya, lihat Bootstrapping di dokumentasi AWS CDK.	AWS DevOps
Buat daftar tumpukan AWS CDK.	Masukkan perintah berikut untuk melihat daftar tumpukan AWS CDK di aplikasi. `cdk ls`	AWS DevOps
Sintesis tumpukan AWS CDK.	Masukkan perintah berikut untuk menghasilkan CloudFormation template AWS untuk setiap tumpukan yang ditentukan dalam aplikasi AWS CDK. `cdk synth`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan aplikasi AWS CDK.	Masukkan perintah berikut untuk menerapkan semua tumpukan ke akun AWS Anda, tanpa memerlukan persetujuan manual untuk perubahan apa pun. `cdk deploy --all --require-approval never`	AWS DevOps

Validasi penerapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke EC2 instance di Wilayah utama.	1. Menggunakan Session Manager, kemampuan AWS Systems Manager, masuk ke EC2 instans di Wilayah utama. Untuk petunjuk, lihat Connect ke instans Linux Anda dengan AWS Systems Manager Session Manager. 2. Ubah direktori ke jalur pemasangan Amazon EFS. `cd /mnt/efs`	AWS DevOps
Buat file sementara.	Masukkan perintah berikut untuk membuat file sementara di jalur pemasangan Amazon EFS. `sudo dd if=/dev/zero \`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`of=tmptst.dat \`
	`bs=1G \`
	`seek=5 \`
	`count=0`
	
	`ls -lrt tmptst.dat`	

Mulai DataSync tugas.	Masukkan perintah berikut untuk mereplikasi file sementara dari Wilayah utama ke Wilayah sekunder, di mana `<ARN-task>` adalah Nama Sumber Daya Amazon (ARN) tugas Anda. DataSync
	
	`aws datasync start-tas`
	`k-execution \`
	`--task-arn <ARN-`
	`task>`
	
	Perintah mengembalikan ARN dari eksekusi tugas dalam format berikut.
	
	`arn:aws:datasync:<`
	`region>:<account-I`
	`D>:task/task-execu`
	`tion/<exec-ID>`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa status transfer data.	Masukkan perintah berikut untuk menggambarkan tugas DataSync eksekusi, di `<ARN-task-execution>` mana ARN dari eksekusi tugas. `aws datasync describe-task-execution \ --task-execution-arn <ARN-task-execution>` DataSync Tugas selesai ketika`PrepareStatus` ,`TransferStatus` , dan `VerifyStatus` semua memiliki nilai`SUCCESS`.	AWS DevOps
Masuk ke EC2 instance di Wilayah sekunder.	1. Menggunakan Session Manager, kemampuan AWS Systems Manager, masuk ke EC2 instans di Wilayah sekunder. Untuk petunjuk, lihat Connect ke instans Linux Anda dengan AWS Systems Manager Session Manager. 2. Ubah direktori ke jalur pemasangan Amazon EFS. `cd /mnt/efs`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi replikasi.	Masukkan perintah berikut untuk memverifikasi bahwa file sementara ada di sistem file Amazon EFS. `ls -lrt tmptst.dat`	AWS DevOps

Sumber daya terkait

Dokumentasi AWS

- AWS CDK API Referensi
- Mengkonfigurasi DataSync transfer AWS dengan Amazon EFS
- Memecahkan masalah dengan transfer AWS DataSync

Sumber daya AWS lainnya

- AWS DataSync FAQs

Uji AWS infrastruktur dengan menggunakan LocalStack dan Tes Terraform

Ivan Girardi dan Ioannis Kalyvas, Amazon Web Services

Ringkasan

Pola ini membantu Anda menguji infrastruktur sebagai kode (IAc) secara lokal AWS di Terraform tanpa perlu menyediakan infrastruktur di lingkungan Anda. AWS Ini mengintegrasikan kerangka Terraform Tests dengan. LocalStack Kontainer LocalStack Docker menyediakan lingkungan pengembangan lokal yang mengemulasi berbagai. Layanan AWS Ini membantu Anda menguji dan mengulangi penerapan infrastruktur tanpa menimbulkan biaya di. AWS Cloud

Solusi ini memberikan manfaat sebagai berikut:

- Optimalisasi biaya — Menjalankan tes terhadap LocalStack menghilangkan kebutuhan untuk menggunakan Layanan AWS. Ini mencegah Anda mengeluarkan biaya yang terkait dengan pembuatan, pengoperasian, dan modifikasi sumber daya tersebut. AWS
- Kecepatan dan efisiensi — Pengujian secara lokal juga biasanya lebih cepat daripada menyebarkan sumber daya. AWS Loop umpan balik yang cepat ini mempercepat pengembangan dan debugging. Karena LocalStack berjalan secara lokal, Anda dapat mengembangkan dan menguji file konfigurasi Terraform Anda tanpa koneksi internet. Anda dapat men-debug file konfigurasi Terraform secara lokal dan menerima umpan balik langsung, yang merampingkan proses pengembangan.
- Konsistensi dan reproduktifitas — LocalStack menyediakan lingkungan yang konsisten untuk pengujian. Konsistensi ini membantu memastikan bahwa pengujian menghasilkan hasil yang sama, terlepas dari AWS perubahan eksternal atau masalah jaringan.
- Isolasi — Pengujian dengan LocalStack mencegah Anda memengaruhi AWS sumber daya hidup atau lingkungan produksi secara tidak sengaja. Isolasi ini membuatnya aman untuk bereksperimen dan menguji berbagai konfigurasi.
- Otomasi - Integrasi dengan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) membantu Anda menguji file konfigurasi Terraform secara otomatis. Pipeline menguji IAc secara menyeluruh sebelum penerapan.
- Fleksibilitas — Anda dapat mensimulasikan berbagai konfigurasi Wilayah AWS Akun AWS, dan layanan agar sesuai dengan lingkungan produksi Anda lebih dekat.

Prasyarat dan batasan

Prasyarat

- Instal Docker
- Aktifkan akses ke soket Docker default (`/var/run/docker.sock`). Lihat informasi yang lebih lengkap dalam dokumentasi LocalStack .
- Instal Docker Compose
- Instal Terraform versi 1.6.0 atau yang lebih baru
- Instal Terraform CLI
- Konfigurasikan Penyedia Terraform AWS
- (Opsional) Instal dan konfigurasikan AWS Command Line Interface (AWS CLI). Untuk contoh cara menggunakan AWS CLI with LocalStack, lihat AWS Infrastruktur GitHub pengujian menggunakan LocalStack dan repositori Terraform Tests.

Batasan

- Pola ini memberikan contoh eksplisit untuk menguji sumber daya Amazon Simple Storage Service (Amazon S3),,, AWS Lambda dan AWS Step Functions Amazon DynamoDB. Namun, Anda dapat memperluas solusi ini untuk memasukkan AWS sumber daya tambahan.
- Pola ini memberikan instruksi untuk menjalankan Terraform Tests secara lokal, tetapi dapatkah Anda dapat mengintegrasikan pengujian ke dalam pipeline apa pun. CI/CD
- Pola ini memberikan instruksi untuk menggunakan gambar LocalStack Komunitas. Jika Anda menggunakan gambar LocalStack Pro, lihat dokumentasi LocalStack Pro.
- LocalStack menyediakan layanan emulasi untuk yang berbeda AWS APIs. Untuk daftar lengkapnya, lihat Cakupan Fitur AWS Layanan. Beberapa fitur canggih mungkin memerlukan langganan untuk LocalStack Pro.

Arsitektur

Diagram berikut menunjukkan arsitektur untuk solusi ini. Komponen utamanya adalah repositori kode sumber, CI/CD pipeline, dan wadah LocalStack Docker. LocalStack Docker Container meng-host yang berikut ini Layanan AWS secara lokal:

- Bucket Amazon S3 untuk menyimpan file

- Amazon CloudWatch untuk pemantauan dan pencatatan
- AWS Lambda Fungsi untuk menjalankan kode tanpa server
- Mesin AWS Step Functions negara untuk mengatur alur kerja multi-langkah
- Tabel Amazon DynamoDB untuk menyimpan data NoSQL

Diagram menunjukkan alur kerja berikut:

1. Anda menambahkan dan mengkomit file konfigurasi Terraform ke repositori kode sumber.
2. CI/CD Pipeline mendeteksi perubahan dan memulai proses pembuatan untuk analisis kode Terraform statis. Pipeline membangun dan menjalankan wadah LocalStack Docker. Kemudian pipa memulai proses pengujian.
3. Pipeline mengunggah objek ke bucket Amazon S3 yang di-host di container Docker LocalStack .
4. Mengunggah objek memanggil fungsi. AWS Lambda
5. Fungsi Lambda menyimpan pemberitahuan peristiwa Amazon S3 dalam log. CloudWatch
6. Fungsi Lambda memulai mesin AWS Step Functions status.
7. Mesin status menulis nama objek Amazon S3 ke dalam tabel DynamoDB.
8. Proses pengujian dalam CI/CD pipeline memverifikasi bahwa nama objek yang diunggah cocok dengan entri dalam tabel DynamoDB. Ini juga memverifikasi bahwa bucket S3 diterapkan dengan nama yang ditentukan dan bahwa AWS Lambda fungsi tersebut telah berhasil diterapkan.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- Docker Compose adalah alat untuk mendefinisikan dan menjalankan aplikasi multi-container.
- LocalStackadalah emulator layanan cloud yang berjalan dalam satu wadah. Dengan menggunakan LocalStack, Anda dapat menjalankan beban kerja pada mesin lokal Anda yang menggunakan Layanan AWS, tanpa menghubungkan ke AWS Cloud.
- Terraform adalah alat IAc HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.
- Tes Terraform membantu Anda memvalidasi pembaruan konfigurasi modul Terraform melalui pengujian yang analog dengan integrasi atau pengujian unit.

Repositori kode

Kode untuk pola ini tersedia di AWS infrastruktur GitHub Uji menggunakan LocalStack dan repositori Terraform Tests.

Praktik terbaik

- Solusi ini menguji AWS infrastruktur yang ditentukan dalam file konfigurasi Terraform, dan tidak menyebarkan sumber daya tersebut di file. AWS Cloud Jika Anda ingin menerapkan sumber daya, ikuti prinsip hak istimewa paling sedikit (dokumentasi IAM) dan konfigurasikan backend Terraform (dokumentasi Terraform) dengan benar.
- Saat mengintegrasikan LocalStack dalam CI/CD pipeline, kami menyarankan Anda untuk tidak menjalankan wadah LocalStack Docker dalam mode hak istimewa. Untuk informasi selengkapnya, lihat hak istimewa Runtime dan kemampuan Linux (dokumentasi Docker) dan Keamanan untuk pelari yang dikelola sendiri (dokumentasi). GitLab

Epik

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Dalam shell bash, masukkan perintah berikut. Ini mengkloning AWS infrastruktur Test menggunakan LocalStack dan repositori Terraform Tests dari: GitHub `git clone https://github.com/aws-samples/localstack-terraform-test.git`	DevOps insinyur

Jalankan LocalStack wadah.	1. Masukkan perintah berikut untuk menavigasi ke repositori kloning:
	`cd localstack-terraform-test`
	2. Masukkan perintah berikut untuk memulai wadah LocalStack Docker dalam mode terpisah:
	`docker-compose up -d`
	3. Tunggu hingga wadah LocalStack Docker beroperasi.
		DevOps insinyur

Inisialisasi Terraform.	Masukkan perintah berikut untuk menginisialisasi Terraform:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`terraform init`	

Jalankan Tes Terraform.	1. Masukkan perintah berikut untuk menjalankan Tes Terraform:
	`terraform test`
	2. Verifikasi bahwa semua tes berhasil diselesai kan. Output harus serupa dengan yang berikut ini:
	`Success! 3 passed, 0 failed.`
		DevOps insinyur

Pembersihan sumber daya	Masukkan perintah berikut untuk menghancurkan LocalStack wadah: `docker-compose down`	DevOps insinyur

Pemecahan masalah

Isu	Solusi
`Error: reading DynamoDB Table Item (Files|README.md): empty` hasil saat menjalankan `terraform test` perintah.	1. Masukkan kembali `terraform test` perintah. 2. Jika itu tidak menyelesaikan kesalahan, edit file.tf utama untuk meningkatkan batas waktu tidur ke nilai lebih dari 15 detik: `resource "time_sleep" "wait" { create_duration = "15s"`

Isu	Solusi

	<pre> triggers = {
	    s3_object = local.key_json
	  }
	}</pre>

Sumber daya terkait

- Memulai dengan Terraform: Panduan untuk AWS CDK dan para AWS CloudFormation ahli (PanduanAWS Preskriptif)
- Praktik terbaik untuk menggunakan AWS Penyedia Terraform (Panduan AWS Preskriptif)
- Terraform CI/CD dan pengujian AWS dengan Terraform Test Framework baru (posting blog)AWS
- Mempercepat pengiriman perangkat lunak menggunakan LocalStack Cloud Emulator dari AWS Marketplace (posting AWS blog)

Informasi tambahan

Integrasi dengan GitHub Tindakan

Anda dapat mengintegrasikan LocalStack dan Pengujian Terraform dalam CI/CD pipeline dengan menggunakan GitHub Tindakan. Untuk informasi selengkapnya, lihat dokumentasi GitHub Tindakan. Berikut ini adalah contoh file konfigurasi GitHub Tindakan:

```
name: LocalStack Terraform Test

on:
  push:
    branches:
      - '**'

  workflow_dispatch: {}

jobs:
  localstack-terraform-test:
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v4
```

```
    - name: Build and Start LocalStack Container
      run: |
        docker compose up -d

    - name: Setup Terraform
      uses: hashicorp/setup-terraform@v3
      with:
        terraform_version: latest

    - name: Run Terraform Init and Validation
      run: |
        terraform init
        terraform validate
        terraform fmt --recursive --check
        terraform plan
        terraform show

    - name: Run Terraform Test
      run: |
        terraform test

    - name: Stop and Delete LocalStack Container
      if: always()
      run: docker compose down
```

Tingkatkan cluster SAP Pacemaker dari ke ENSA1 ENSA2

Gergely Cserdi dan Balazs Sandor Skublics, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah dan pertimbangan untuk memutakhirkan cluster SAP Pacemaker yang didasarkan pada Standalone Enqueue Server () ke. ENSA1 ENSA2 Informasi dalam pola ini berlaku untuk sistem operasi SUSE Linux Enterprise Server (SLES) dan Red Hat Enterprise Linux (RHEL).

Cluster alat pacu jantung pada SAP NetWeaver 7.52 atau S/4HANA 1709 dan versi sebelumnya berjalan pada arsitektur dan dikonfigurasi khusus untuk. ENSA1 ENSA1 Jika Anda menjalankan beban kerja SAP Anda di Amazon Web Services (AWS) dan Anda tertarik untuk pindah ke ENSA2, Anda mungkin menemukan bahwa dokumentasi SAP, SUSE, dan RHEL tidak memberikan informasi yang komprehensif. Pola ini menjelaskan langkah-langkah teknis yang diperlukan untuk mengkonfigurasi ulang parameter SAP dan cluster Pacemaker untuk meningkatkan dari ke. ENSA1 ENSA2 Ini memberikan contoh sistem SUSE, tetapi konsepnya sama untuk cluster RHEL.

> **Note**
>
> ENSA1 dan ENSA2 merupakan konsep yang hanya berkaitan dengan aplikasi SAP, sehingga informasi dalam pola ini tidak berlaku untuk SAP HANA atau jenis cluster lainnya.

> **Note**
>
> Secara teknis, ENSA2 dapat digunakan dengan atau tanpa Enqueue Replicator 2. Namun, ketersediaan tinggi (HA) dan otomatisasi failover (melalui solusi cluster) memerlukan Enqueue Replicator 2. Pola ini menggunakan istilah ENSA2 cluster untuk merujuk ke cluster dengan Standalone Enqueue Server 2 dan Enqueue Replicator 2.

Prasyarat dan batasan

Prasyarat

- Cluster ENSA1 berbasis kerja yang menggunakan Pacemaker dan Corosync pada SLES atau RHEL.

- Setidaknya dua instans Amazon Elastic Compute Cloud (Amazon EC2) di mana instans (ABAP) SAP Central Services (ASCS/SCS) dan Enqueue Replication Server (ERS) berjalan.
- Pengetahuan tentang mengelola aplikasi dan cluster SAP.
- Akses ke lingkungan Linux sebagai pengguna root.

Keterbatasan

- ENSA1cluster berbasis hanya mendukung arsitektur dua simpul.
- ENSA2cluster berbasis tidak dapat digunakan ke NetWeaver versi SAP sebelum 7.52.
- EC2 instance dalam cluster harus berada di AWS Availability Zones yang berbeda.

Versi produk

- SAP NetWeaver versi 7.52 atau yang lebih baru
- Dimulai dengan S/4HANA 2020, hanya ENSA2 cluster yang didukung
- Kernel 7.53 atau yang lebih baru, yang mendukung ENSA2 dan Enqueue Replicator 2
- SLES untuk Aplikasi SAP versi 12 atau yang lebih baru
- RHEL untuk SAP dengan High Availability (HA) versi 7.9 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- SAP NetWeaver 7.52 dengan SAP Kernel 7.53 atau yang lebih baru
- Sistem operasi SLES atau RHEL

Tumpukan teknologi target

- SAP NetWeaver 7.52 dengan SAP Kernel 7.53 atau lebih baru, termasuk S/4HANA 2020 dengan platform ABAP
- Sistem operasi SLES atau RHEL

Arsitektur target

Diagram berikut menunjukkan konfigurasi HA ASCS/SCS dan instans ERS berdasarkan ENSA2 cluster.

Perbandingan ENSA1 dan ENSA2 cluster

SAP diperkenalkan ENSA2 sebagai penerus. ENSA1 Cluster ENSA1 berbasis mendukung arsitektur dua simpul di mana ASCS/SCS instance gagal ke ERS ketika terjadi kesalahan. Keterbatasan ini berasal dari bagaimana ASCS/SCS instance mendapatkan kembali informasi tabel kunci dari memori bersama node ERS setelah failover. ENSA2cluster berbasis dengan Enqueue Replicator 2 menghilangkan batasan ini, karena ASCS/SCS instance dapat mengumpulkan informasi kunci dari instance ERS melalui jaringan. ENSA2cluster berbasis dapat memiliki lebih dari dua node, karena ASCS/SCS instance tidak lagi diperlukan untuk gagal ke node ERS. (Namun, dalam lingkungan ENSA2 cluster dua simpul, ASCS/SCS instance masih akan gagal ke node ERS karena tidak ada node lain di cluster yang gagal.) ENSA2 didukung dimulai dengan SAP Kernel 7.50 dengan beberapa batasan. Untuk pengaturan HA yang mendukung Enqueue Replicator 2, persyaratan minimum adalah NetWeaver 7.52 (lihat SAP OSS Note 2630416). S/4HANA 1809 hadir dengan ENSA2 arsitektur yang direkomendasikan secara default, sedangkan S/4HANA hanya ENSA2 mendukung dimulai dengan versi 2020.

Otomatisasi dan skala

Cluster HA dalam arsitektur target membuat ASCS gagal ke node lain secara otomatis.

Skenario untuk pindah ke cluster ENSA2 berbasis

Ada dua skenario utama untuk meningkatkan ke cluster ENSA2 berbasis:

- Skenario 1: Anda memilih untuk meningkatkan ke ENSA2 tanpa upgrade SAP atau konversi S/4HANA yang menyertainya, dengan asumsi bahwa rilis SAP dan versi Kernel Anda mendukung. ENSA2
- Skenario 2: Anda pindah ke ENSA2 sebagai bagian dari upgrade atau konversi (misalnya, ke S/4HANA 1809 atau lebih baru) dengan menggunakan SUM.

Bagian Epik mencakup langkah-langkah untuk dua skenario ini. Skenario pertama mengharuskan Anda untuk mengatur parameter terkait SAP secara manual sebelum Anda mengubah konfigurasi cluster untuk. ENSA2 Dalam skenario kedua, binari dan parameter terkait SAP diterapkan oleh SUM, dan satu-satunya tugas Anda yang tersisa adalah memperbarui konfigurasi cluster untuk HA. Kami

masih menyarankan Anda memvalidasi parameter SAP setelah Anda menggunakan SUM. Dalam kebanyakan kasus, konversi S/4HANA adalah alasan utama untuk upgrade cluster.

Alat

- Untuk manajer paket OS, kami merekomendasikan alat Zypper (untuk SLES) atau YUM (untuk RHEL).
- Untuk manajemen cluster, kami merekomendasikan shell crm (untuk SLES) atau pcs (untuk RHEL).
- Alat manajemen instans SAP seperti SAPControl.
- (Opsional) Alat SUM untuk peningkatan konversi S/4HANA.

Praktik terbaik

- Untuk praktik terbaik dalam menggunakan beban kerja SAP di AWS, lihat Lensa SAP untuk AWS Well-Architected Framework.
- Pertimbangkan jumlah node cluster (ganjil atau genap) dalam arsitektur ENSA2 multi-node Anda.
- Siapkan ENSA2 cluster untuk SLES 15 selaras dengan standar sertifikasi SAP S/4-HA-CLU 1.0.
- Selalu simpan atau buat cadangan klaster dan status aplikasi yang ada sebelum memutakhirkan ke ENSA2.

Epik

Konfigurasikan parameter SAP secara manual untuk ENSA2 (skenario 1 saja)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan parameter di profil default.	Jika Anda ingin meningkat kan ke ENSA2 sambil tetap pada rilis SAP yang sama atau jika rilis target Anda default ENSA1, atur parameter di profil default (file DEFAULT.P FL) ke nilai berikut. `enq/enable=TRUE`	SAP

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>enq/serverhost=sapas
	csvirt
	enq/serverinst=10
	    (instance number of
	 ASCS/SCS instance)
	enque/process_loc
	ation=REMOTESA
	enq/replicatorhost=sa
	persvirt
	enq/replicatorinst=11
	    (instance number of
	 ERS instance)</pre>
	di mana `sapascsvirt` adalah nama host virtual untuk instance ASCS, dan `sapersvirt` merupakan nama host virtual untuk instance ERS. Anda dapat mengubah ini agar sesuai dengan lingkungan target Anda.
	
	Note
	Untuk menggunakan opsi upgrade ini, rilis SAP dan versi Kernel Anda harus mendukung ENSA2 dan Enqueue Replicator 2.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan profil ASCS/SCS instance.	Jika Anda ingin memutakhirkan ke ENSA2 sambil tetap menggunakan rilis SAP yang sama atau jika rilis target Anda default ENSA1, atur parameter berikut di profil instance. ASCS/SCS Bagian profil di mana ENSA1 didefinisikan terlihat seperti berikut ini. `#------------------ ------------------ ------------------ --------- Start SAP enqueue server #------------------- ------------------ ------------------ ------- _EN = en.sap$(S APSYSTEMNAME)$(INS TANCE_NAME) Execute_04 = local rm - f $(_EN) Execute_05 = local ln - s -f $(DIR_EXECUTABLE)/ enserver$(FT_EXE) $(_EN) Start_Program_01 = local $(_EN) pf=$(_PF)` Untuk mengkonfigurasi ulang bagian ini untuk ENSA2:	SAP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Ubah awalan _EN program menjadi _ENQ berdasarkan informasi terbaru dari SAP (OSS Note 2501860; memerlukan akun pengguna SAP ONE Support Launchpad). 2. Ubah biner untuk enqueue server dari `enserver` ke. `enq_server` 3. Atur parameter baru `enq/server/replication/enable` ke`TRUE`. 4. Pastikan itu`Autostart = 0`. Bagian profil ini akan terlihat seperti berikut setelah perubahan Anda. `#------------------ ------------------ ------------------ --------- Start SAP enqueue server #------------------- ------------------ ------------------ ------- _ENQ = enq.sap$( SAPSYSTEMNAME)$(IN STANCE_NAME) Execute_04 = local rm - f $(_ENQ) Execute_05 = local ln - s -f $(DIR_EXECUTABLE)/`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`enq_server$(FT_EXE) $(_ENQ)`
	`Start_Program_01 = local $(_ENQ) pf= $(_PF)`
	`...`
	`enq/server/replication/enable = TRUE`
	`Autostart = 0`
	
	Important
	_ENQtidak boleh mengaktifkan opsi restart. Jika `RestartProgram_01` diatur untuk_ENQ, ubah menjadi`StartProgram_01` . Ini mencegah SAP memulai ulang layanan atau mengganggu sumber daya yang dikelola cluster.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan profil ERS.	Jika Anda ingin meningkatkan ke ENSA2 sementara tetap pada rilis SAP yang sama atau jika rilis target Anda default ENSA1, atur parameter berikut di profil instans ERS. Temukan bagian di mana replikator enqueue didefinis ikan. Ini akan mirip dengan yang berikut ini. `#----------------- ----------------- ------------------ Start enqueue replicati on server #------------------ ----------------- ----------------- _ER = er.sap$(S APSYSTEMNAME)$(INS TANCE_NAME) Execute_03 = local rm - f $(_ER) Execute_04 = local ln - s -f $(DIR_EXECUTABLE)/ enrepserver$(FT_EXE) $(_ER) Start_Program_00 = local $(_ER) pf=$(_PF) NR=$(SCSID)` Untuk mengkonfigurasi ulang bagian ini untuk Enqueue Replicator 2:	SAP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Ubah awalan _ER program menjadi _ENQR berdasarkan catatan terbaru dari SAP (OSS Note 2501860; membutuhkan akun pengguna SAP ONE Support Launchpad). 2. Ubah biner untuk replikator enqueue menjadi `enq_replicator` bukan. `enrepserver` 3. Pastikan itu`Autostart = 0`. Bagian profil ini akan terlihat seperti berikut setelah perubahan Anda. `#------------------ ------------------ ------------------- Start enqueue replication server #------------------- ------------------ ------------------ _ENQR = enqr.sap$(SAPSYSTEMNAME)$(INSTANCE_NAME) Execute_01 = local rm -f $(_ENQR) Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/enq_replicator$(FT_EXE) $(_ENQR)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Start_Program_00 = local $(_ENQR) pf= $(_PF) NR=$(SCSID) … Autostart = 0` **Important** _ENQRtidak boleh mengaktifkan opsi restart. Jika `RestartProgram_01` diatur untuk_ENQR, ubah menjadi`StartProgram_01` . Ini mencegah SAP memulai ulang layanan atau mengganggu layanan yang dikelola cluster.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulai ulang Layanan Mulai SAP.	Setelah Anda mengubah profil yang dijelaskan sebelumny a dalam epik ini, restart SAP Start Services untuk keduanya ASCS/SCS dan ERS.
	`sapcontrol -nr 10 -function RestartSe rvice SCT`
	`sapcontrol -nr 11 -function RestartSe rvice SCT`
	di mana `SCT` mengacu pada ID sistem SAP, dan dengan asumsi bahwa 10 dan 11 adalah nomor instance untuk ASCS/SCS dan instans ERS, masing-masing.
		SAP

Konfigurasi ulang cluster untuk ENSA2 (diperlukan untuk kedua skenario)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi nomor versi di agen sumber daya SAP.	Ketika Anda menggunak an SUM untuk meng-upgr ade SAP ke S/4HANA 1809 atau yang lebih baru, SUM menangani perubahan parameter dalam profil SAP. Hanya cluster yang membutuhkan penyesuai an manual. Namun, kami menyarankan Anda memverifi	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kasi pengaturan parameter sebelum Anda membuat perubahan apa pun pada cluster. **Note** Contoh dalam epik ini mengasums ikan bahwa Anda menggunakan sistem operasi SUSE. Jika Anda menggunakan RHEL, Anda harus menggunakan alat seperti YUM dan shell pcs alih-alih Zypper dan crm. Periksa kedua node dalam arsitektur untuk mengonfir masi bahwa `resource-agents` paket tersebut cocok dengan versi minimum yang direkomendasikan oleh SAP. Untuk SLES, periksa SAP OSS Note 2641019. Untuk RHEL, periksa SAP OSS Note 2641322. (SAP Notes memerlukan akun pengguna SAP ONE Support Launchpad .)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`sapers:sctadm 23> zypper search -s -i resource-agents`
	`Loading repository data...`
	`Reading installed packages...`
	`S | Name | Type | Version | Arch | Repository`
	`--+-----------------+---------+-----------------+---------------+------+-----------------`
	`i | resource-agents | package | 4.8.0+git30.d0077df0-150300.8.28.1 | x86_64 | SLE-Product-HA15-SP3-Updates`
	
	Perbarui `resource-agents` versi jika perlu.	

Cadangkan konfigurasi cluster.	Cadangkan konfigurasi cluster CRM sebagai berikut. `crm configure show > /tmp/cluster_config_backup.txt`	Administrator sistem AWS
Atur mode pemeliharaan.	Atur cluster ke mode pemeliharaan. `crm configure property maintenance-mode="true"`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa konfigurasi cluster.	Periksa konfigurasi cluster saat ini.
	`crm configure show`
	Berikut adalah kutipan dari output penuh:
	`node 1: sapascs`
	`node 2: sapers`
	`...`
	`primitive rsc_sap_S`
	`CT_ASCS10 SAPInstance`
	` \`
	`operations $id=rsc_s`
	`ap_SCT_ASCS10-oper`
	`ations \`
	`op monitor interval=120`
	` timeout=60 on-fail=r`
	`estart \`
	`params InstanceN`
	`ame=SCT_ASCS10_sap`
	`ascsvirt START_PRO`
	`FILE="/sapmnt/SCT/`
	`profile/SCT_ASCS10`
	`_sapascsvirt" \`
	`AUTOMATIC_RECOVER=`
	`false \`
	`meta resource-stickines`
	`s=5000 failure-t`
	`imeout=60 migration-`
	`threshold=1 priority=`
	`10`
	`primitive rsc_sap_S`
	`CT_ERS11 SAPInstance \`
	`operations $id=rsc_s`
	`ap_SCT_ERS11-opera`
	`tions \`
	`op monitor interval=120`
	` timeout=60 on-fail=r`
	`estart \`
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>params InstanceN
	ame=SCT_ERS11_sape
	rsvirt START_PRO
	FILE="/sapmnt/SCT/
	profile/SCT_ERS11_
	sapersvirt" \
	   AUTOMATIC_RECOVER=
	false IS_ERS=true \
	meta priority=1000
	...
	colocation col_sap_S
	CT_no_both -5000:
	 grp_SCT_ERS11
	 grp_SCT_ASCS10
	location loc_sap_S
	CT_failover_to_ers
	 rsc_sap_SCT_ASCS10 \
	rule 2000: runs_ers_SCT
	 eq 1
	order ord_sap_S
	CT_first_start_asc
	s Optional: rsc_sap_S
	CT_ASCS10:start
	 rsc_sap_SCT_ERS11:
	stop symmetrical=false
	...</pre>
	di mana `sapascsvirt` mengacu pada nama host virtual untuk instance ASCS, `sapersvirt` mengacu pada nama host virtual untuk instans ERS, dan `SCT` mengacu pada ID sistem SAP.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus kendala kolokasi failover.	Dalam contoh sebelumnya, batasan lokasi `loc_sap_SCT_failover_to_ers` menentukan bahwa ENSA1 fitur ASCS harus selalu mengikuti instance ERS pada failover. Dengan ENSA2, ASCS harus dapat gagal secara bebas ke node yang berpartisipasi, sehingga Anda dapat menghapus kendala ini. `crm configure delete loc_sap_SCT_failover_to_ers`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Sesuaikan primitif.	Anda juga perlu membuat perubahan kecil pada SAPInstance primitif ASCS dan ERS.
	
	Berikut adalah contoh SAPInstance primitif ASCS yang dikonfigurasi untuk. ENSA1
	
	`primitive rsc_sap_SCT_ASCS10 SAPInstance \`
	`operations $id=rsc_sap_SCT_ASCS10-operations \`
	`op monitor interval=120 timeout=60 on-fail=restart \`
	`params InstanceName=SCT_ASCS10_sapascsvirt START_PROFILE="/sapmnt/SCT/profile/SCT_ASCS10_sapascsvirt" \`
	`AUTOMATIC_RECOVER=false \`
	`meta resource-stickiness=5000 failure-timeout=60 migration-threshold=1 priority=10`
	
	Untuk memutakhirkan ke ENSA2, ubah konfigurasi ini menjadi yang berikut.
		Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`primitive rsc_sap_S CT_ASCS10 SAPInstance \ operations $id=rsc_s ap_SCT_ASCS10-oper ations \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceN ame=SCT_ASCS10_sap ascsvirt START_PRO FILE="/sapmnt/SCT/ profile/SCT_ASCS10 _sapascsvirt" \ AUTOMATIC_RECOVER= false \ meta resource-stickines s=3000`  Ini adalah contoh SAPInstance primitif ERS yang dikonfigurasi untuk ENSA1.  `primitive rsc_sap_S CT_ERS11 SAPInstance \ operations $id=rsc_s ap_SCT_ERS11-opera tions \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceN ame=SCT_ERS11_sape rsvirt START_PRO FILE="/sapmnt/SCT/ profile/SCT_ERS11_ sapersvirt" \ AUTOMATIC_RECOVER= false IS_ERS=true \`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`meta priority=1000`
	
	Untuk memutakhirkan ke ENSA2, ubah konfigurasi ini menjadi yang berikut.
	
	`primitive rsc_sap_SCT_ERS11 SAPInstance \ operations $id=rsc_sap_SCT_ERS11-operations \ op monitor interval=120 timeout=60 on-fail=restart \ params InstanceName=SCT_ERS11_sapersvirt START_PROFILE="/sapmnt/SCT/profile/SCT_ERS11_sapersvirt" \ AUTOMATIC_RECOVER=false IS_ERS=true`
	
	Anda dapat mengubah primitif dengan berbagai cara. Misalnya, Anda dapat merevisinya di editor seperti vi, seperti pada contoh berikut.
	
	`crm configure edit rsc_sap_SCT_ERS11`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Nonaktifkan mode pemeliharaan.	Nonaktifkan mode pemeliharaan pada cluster. `crm configure property maintenance-mode="false"` Ketika cluster keluar dari mode pemeliharaan, ia mencoba untuk membawa instans ASCS dan ERS online dengan pengaturan baru ENSA2 .	Administrator sistem AWS

(Opsional) Tambahkan node cluster

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau praktik terbaik.	Sebelum Anda menambahkan lebih banyak node, pastikan untuk memahami praktik terbaik seperti apakah akan menggunakan jumlah node ganjil atau genap.	Administrator sistem AWS
Tambahkan node.	Menambahkan lebih banyak node melibatkan serangkaian tugas, seperti memperbarui sistem operasi, menginstal paket perangkat lunak yang cocok dengan node yang ada, dan membuat mount tersedia. Anda dapat menggunakan opsi Siapkan Host Tambahan di SAP Software Provision	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ing Manager (SWPM) untuk membuat baseline khusus SAP dari host. Untuk informasi selengkapnya, lihat panduan SAP yang tercantum di bagian selanjutnya.	

Sumber daya terkait

Referensi SAP dan SUSE

Untuk mengakses SAP Notes, Anda harus memiliki akun pengguna SAP ONE Support Launchpad. Untuk informasi selengkapnya, lihat situs web SAP Support.

- SAP Note 2501860 - Dokumentasi untuk Server NetWeaver Aplikasi SAP untuk ABAP 7.52
- SAP Note 2641019 - Instalasi ENSA2 dan pembaruan dari ENSA1 ke ENSA2 dalam lingkungan SUSE HA
- SAP Note 2641322 - Instalasi ENSA2 dan perbarui dari ENSA1 hingga ENSA2 saat menggunakan solusi Red Hat HA untuk SAP
- SAP Note 2711036 - Penggunaan Standalone Enqueue Server 2 di Lingkungan HA
- Server Enqueue Mandiri 2 (dokumentasi SAP)
- SAP S/4 HANA - Enqueue Replication 2 High Availability Cluster - Panduan Pengaturan (dokumentasi SUSE)

Referensi AWS

- SAP HANA on AWS: Panduan Konfigurasi Ketersediaan Tinggi untuk SLES dan RHEL
- SAP Lens - AWS Well-Architected Framework

Gunakan Availability Zone yang konsisten VPCs di berbagai akun AWS

Adam Spicer, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, Availability Zone memiliki nama yang dapat bervariasi antara akun AWS Anda dan ID Availability Zone (ID AZ) yang mengidentifikasi lokasinya. Jika Anda menggunakan AWS CloudFormation untuk membuat virtual private cloud (VPCs), Anda harus menentukan nama atau ID Availability Zone saat membuat subnet. Jika Anda membuat VPCs di beberapa akun, nama Availability Zone diacak, yang berarti subnet menggunakan Availability Zone yang berbeda di setiap akun.

Untuk menggunakan Availability Zone yang sama di seluruh akun Anda, Anda harus memetakan nama Availability Zone di setiap akun ke ID AZ yang sama. Misalnya, diagram berikut menunjukkan bahwa ID `use1-az6` AZ diberi nama `us-east-1a` di akun AWS A dan `us-east-1c` di akun AWS Z.

Pola ini membantu memastikan konsistensi zona dengan menyediakan solusi lintas akun yang dapat diskalakan untuk menggunakan Availability Zone yang sama di subnet Anda. Konsistensi zona memastikan bahwa lalu lintas jaringan lintas akun Anda menghindari jalur jaringan Zona Ketersediaan Lintas, yang membantu mengurangi biaya transfer data dan menurunkan latensi jaringan di antara beban kerja Anda.

Pola ini merupakan pendekatan alternatif untuk CloudFormation AvailabilityZoneId properti AWS.

Prasyarat dan batasan

Prasyarat

- Setidaknya dua akun AWS aktif di Wilayah AWS yang sama.
- Evaluasi berapa banyak Availability Zone yang diperlukan untuk mendukung persyaratan VPC Anda di Region.
- Identifikasi dan catat ID AZ untuk setiap Availability Zone yang perlu Anda dukung. Untuk informasi selengkapnya tentang hal ini, lihat Availability Zone IDs untuk sumber daya AWS Anda di dokumentasi AWS Resource Access Manager.

- Daftar AZ Anda yang dipesan dan dipisahkan koma. IDs Misalnya, Availability Zone pertama pada daftar Anda dipetakan sebagai`az1`, Availbility Zone kedua dipetakan sebagai`az2`, dan struktur pemetaan ini berlanjut hingga daftar yang dipisahkan koma Anda sepenuhnya dipetakan. Tidak ada jumlah maksimum AZ IDs yang dapat dipetakan.
- `az-mapping.yaml`File dari repositori pemetaan Zona Ketersediaan GitHub Multi-akun, disalin ke mesin lokal Anda

Arsitektur

Diagram berikut menunjukkan arsitektur yang diterapkan di akun dan yang menciptakan nilai AWS Systems Manager Parameter Store. Nilai Parameter Store ini dikonsumsi saat Anda membuat VPC di akun.

Diagram menunjukkan alur kerja berikut:

1. Solusi pola ini diterapkan ke semua akun yang memerlukan konsistensi zona untuk VPC.
2. Solusi ini membuat nilai Parameter Store untuk setiap ID AZ dan menyimpan nama Availability Zone yang baru.
3. CloudFormation Template AWS menggunakan nama Availability Zone yang disimpan di setiap nilai Parameter Store dan ini memastikan konsistensi zona.

Diagram berikut menunjukkan alur kerja untuk membuat VPC dengan solusi pola ini.

Diagram menunjukkan alur kerja berikut:

1. Kirim template untuk membuat VPC ke AWS. CloudFormation
2. AWS CloudFormation menyelesaikan nilai Parameter Store untuk setiap Availability Zone dan mengembalikan nama Availability Zone untuk setiap ID AZ.
3. VPC dibuat dengan AZ yang benar yang IDs diperlukan untuk konsistensi zona.

Setelah menerapkan solusi pola ini, Anda dapat membuat subnet yang mereferensikan nilai Parameter Store. Jika Anda menggunakan AWS CloudFormation, Anda dapat mereferensikan nilai parameter pemetaan Availability Zone dari kode sampel berformat YAML berikut:

```
Resources:
    PrivateSubnet1AZ1:
        Type: AWS::EC2::Subnet
        Properties:
            VpcId: !Ref VPC
            CidrBlock: !Ref PrivateSubnetAZ1CIDR
            AvailabilityZone:
                !Join
                    - ''
                    - - '{{resolve:ssm:/az-mapping/az1:1}}'
```

Kode contoh ini terdapat dalam `vpc-example.yaml` file dari repositori pemetaan Zona Ketersediaan GitHub Multi-akun. Ini menunjukkan cara membuat VPC dan subnet yang sejajar dengan nilai Parameter Store untuk konsistensi zona.

Tumpukan teknologi

- AWS CloudFormation
- AWS Lambda
- AWS Systems Manager Parameter Store

Otomatisasi dan skala

Anda dapat menerapkan pola ini ke semua akun AWS Anda dengan menggunakan AWS CloudFormation StackSets atau solusi Kustomisasi untuk AWS Control Tower. Untuk informasi selengkapnya, lihat Bekerja dengan AWS CloudFormation StackSets dalam dokumentasi AWS Cloudformation dan Kustomisasi untuk AWS Control Tower di AWS Solutions Library.

Setelah menerapkan CloudFormation template AWS, Anda dapat memperbaruinya untuk menggunakan nilai Parameter Store dan menerapkan VPCs di pipeline atau sesuai dengan kebutuhan Anda.

Alat

Layanan AWS

- AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-

alih mengelola sumber daya secara individual. Anda dapat mengelola dan menyediakan tumpukan di beberapa akun AWS dan Wilayah AWS.

- AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.
- AWS Systems Manager Parameter Store adalah kemampuan AWS Systems Manager. Ini menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.

Kode

Kode untuk pola ini disediakan di repositori pemetaan Zona Ketersediaan GitHub Multi-akun.

Epik

Terapkan file az-mapping.yaml

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan Availability Zone yang diperlukan untuk Wilayah tersebut.	1. Tentukan AZ IDs yang harus digunakan secara konsisten di Wilayah Anda. 2. Rekam AZ ini IDs dalam daftar yang dipisahkan koma dan dalam urutan yang Anda inginkan untuk diterapkan. Misalnya, Availability Zone pertama pada daftar Anda dipetakan sebagai `az1` dan yang kedua dipetakan sebagai. `az2` Tidak ada jumlah maksimum AZ IDs yang dapat dipetakan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan file az-mappin g.yaml.	Gunakan `az-mappin g.yaml` file untuk membuat CloudFormation tumpukan AWS di semua akun AWS yang diperlukan. Dalam `AZIds` parameter, gunakan daftar dipisahkan koma yang Anda buat sebelumnya. Kami menyarankan Anda menggunakan AWS CloudFormation StackSets atau Kustomisasi untuk AWS Control Tower Solution.	Arsitek awan

Terapkan VPCs di akun Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kustomisasi CloudFormation templat AWS.	Saat Anda membuat subnet menggunakan AWS CloudFormation, sesuaikan templat untuk menggunakan nilai Parameter Store yang Anda buat sebelumnya. Untuk contoh templat, lihat `vpc-example.yaml` file di repositori pemetaan Zona Ketersediaan GitHub Multi-aku n.	Arsitek awan
Menyebarkan. VPCs	Terapkan CloudFormation templat AWS yang disesuaik an ke akun Anda. Setiap VPC	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di Wilayah kemudian memiliki konsistensi zona di Availabil ity Zone yang digunakan untuk subnet	

Sumber daya terkait

- Availability Zone IDs untuk sumber daya AWS Anda (dokumentasi AWS Resource Access Manager)
- AWS::EC2::Subnet(CloudFormation Dokumentasi AWS)

Gunakan pengguna IDs dalam kebijakan IAM untuk kontrol akses dan otomatisasi

Srinivas Ananda Babu dan Ram Kandaswamy, Amazon Web Services

Ringkasan

Pola ini menjelaskan potensi jebakan menggunakan kebijakan berbasis nama pengguna di AWS Identity and Access Management (IAM), manfaat menggunakan pengguna IDs, dan bagaimana mengintegrasikan pendekatan ini dengan otomatisasi. AWS CloudFormation

Di dalamnya AWS Cloud, layanan IAM membantu Anda mengelola identitas pengguna dan kontrol akses dengan presisi. Namun, ketergantungan pada nama pengguna dalam pembuatan kebijakan IAM dapat menyebabkan risiko keamanan yang tidak terduga dan masalah kontrol akses. Misalnya, pertimbangkan skenario ini: Karyawan baru, John Doe, bergabung dengan tim Anda, dan Anda membuat akun pengguna IAM dengan nama pengguna`j.doe`, yang memberi mereka izin melalui kebijakan IAM yang mereferensikan nama pengguna. Ketika John meninggalkan perusahaan, akun tersebut dihapus. Masalahnya dimulai ketika karyawan baru, Jane Doe, bergabung dengan tim Anda, dan `j.doe` nama pengguna dibuat ulang. Kebijakan yang ada sekarang memberi Jane Doe izin yang sama dengan John Doe. Ini menciptakan mimpi buruk keamanan dan kepatuhan potensial.

Memperbarui setiap kebijakan secara manual untuk mencerminkan detail pengguna baru adalah proses yang memakan waktu dan rawan kesalahan, terutama seiring pertumbuhan organisasi Anda. Solusinya adalah dengan menggunakan ID pengguna yang unik dan tidak dapat diubah. Saat Anda membuat akun pengguna IAM, AWS berikan ID pengguna unik (atau ID utama) kepada pengguna IAM. Anda dapat menggunakan pengguna ini IDs dalam kebijakan IAM Anda untuk memastikan kontrol akses yang konsisten dan andal yang tidak terpengaruh oleh perubahan nama pengguna atau penggunaan kembali.

Misalnya, kebijakan IAM yang menggunakan ID pengguna mungkin terlihat seperti ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::example-bucket",
            "Principal": { "AWS": "arn:aws:iam::123456789012:user/abcdef01234567890" }
        }
```

```
        ]
}
```

Manfaat menggunakan pengguna IDs dalam kebijakan IAM meliputi:

- Keunikan. Pengguna IDs unik di semua Akun AWS, sehingga mereka memberikan aplikasi izin yang benar dan konsisten.
- Kekekalan. Pengguna IDs tidak dapat diubah, sehingga mereka menyediakan pengenal yang stabil untuk merujuk pengguna dalam kebijakan.
- Audit dan kepatuhan. Layanan AWS sering menyertakan pengguna IDs dalam log dan jejak audit, yang membuatnya mudah untuk melacak tindakan kembali ke pengguna tertentu.
- Otomasi dan integrasi. Menggunakan pengguna IDs dalam AWS APIs, SDKs, atau skrip otomatisasi memastikan bahwa proses tetap tidak terpengaruh oleh perubahan nama pengguna.
- Pemeriksaan masa depan. Menggunakan kebijakan pengguna IDs sejak awal dapat mencegah potensi masalah kontrol akses atau pembaruan kebijakan yang ekstensif.

Otomatisasi

Saat Anda menggunakan alat infrastruktur sebagai kode (IAc) seperti AWS CloudFormation, jebakan kebijakan IAM berbasis nama pengguna masih dapat menyebabkan masalah. Sumber daya pengguna IAM mengembalikan nama pengguna saat Anda memanggil fungsi `Ref` intrinsik. Seiring perkembangan infrastruktur organisasi Anda, siklus pembuatan dan penghapusan sumber daya, termasuk akun pengguna IAM, dapat menyebabkan masalah kontrol akses yang tidak diinginkan jika Anda menggunakan kembali nama pengguna.

Untuk mengatasi masalah ini, kami sarankan Anda memasukkan pengguna IDs ke dalam CloudFormation template Anda. Namun, mendapatkan pengguna IDs untuk tujuan ini bisa menjadi tantangan. Di sinilah sumber daya khusus dapat membantu. Anda dapat menggunakan sumber daya CloudFormation khusus untuk memperluas fungsionalitas layanan dengan mengintegrasikan dengan AWS APIs atau layanan eksternal. Dengan membuat sumber daya kustom yang mengambil ID pengguna untuk pengguna IAM tertentu, Anda dapat membuat ID pengguna tersedia dalam template Anda CloudFormation . Pendekatan ini merampingkan proses referensi pengguna IDs dan memastikan bahwa alur kerja otomatisasi Anda tetap kuat dan tahan masa depan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Peran IAM untuk administrator cloud untuk menjalankan template CloudFormation

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Arsitektur target

Diagram berikut menunjukkan bagaimana CloudFormation menggunakan sumber daya kustom yang didukung oleh AWS Lambda untuk mengambil ID pengguna IAM.

Otomatisasi dan skala

Anda dapat menggunakan CloudFormation template beberapa kali untuk berbagai Wilayah AWS dan akun. Anda hanya perlu menjalankannya sekali di setiap Wilayah atau akun.

Alat

Layanan AWS

- IAM — AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke sumber daya dengan aman. AWS Anda menggunakan IAM untuk mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan sumber daya.
- AWS CloudFormation— AWS CloudFormation membantu Anda memodelkan dan mengatur AWS sumber daya Anda sehingga Anda dapat menghabiskan lebih sedikit waktu mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi Anda yang berjalan AWS. Anda membuat template yang menjelaskan AWS sumber daya yang Anda inginkan, dan CloudFormation mengurus penyediaan dan konfigurasi sumber daya tersebut untuk Anda.
- AWS Lambda— AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat

diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.

Praktik terbaik

Jika Anda memulai dari awal atau merencanakan penerapan greenfield, kami sangat menyarankan Anda menggunakannya AWS IAM Identity Centeruntuk manajemen pengguna terpusat. IAM Identity Center terintegrasi dengan penyedia identitas Anda yang ada (seperti Active Directory atau Okta) untuk menyatukan identitas pengguna AWS, yang menghilangkan kebutuhan untuk membuat dan mengelola pengguna IAM secara langsung. Pendekatan ini tidak hanya memastikan kontrol akses yang konsisten tetapi juga menyederhanakan manajemen siklus hidup pengguna dan membantu meningkatkan keamanan dan kepatuhan di seluruh lingkungan Anda. AWS

Epik

Validasi izin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi peran Anda Akun AWS dan IAM.	Konfirmasikan bahwa Anda memiliki peran IAM dengan izin untuk menerapkan CloudFormation templat di situs Anda. Akun AWS Jika Anda berencana menggunakan AWS CLI alih-alih CloudFormation konsol untuk menerapka n template pada langkah terakhir prosedur ini, Anda juga harus menyiapkan kredensyal sementara untuk menjalankan perintah. AWS CLI Untuk petunjuk, lihat dokumentasi IAM.	Arsitek awan

Membangun CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation template.	1. Buat CloudFormation template dengan mengikuti instruksi dalam CloudFormation dokumentasi. Anda dapat menggunakan format JSON atau YAMM. Pola ini mengasumsikan bahwa Anda menggunakan format YAMM. 2. Simpan template dengan nama `get_unique_user_id.yaml` .	AWS DevOps, arsitek Cloud

Tambahkan parameter input untuk nama pengguna.	Tambahkan kode berikut ke `Parameters` bagian CloudFormation template:
	`Parameters:`
	`  NewIamUserName:`
	`  Type: String`
	`  Description: Unique username for the new IAM user`
	Parameter ini meminta pengguna untuk nama pengguna.
		AWS DevOps, arsitek Cloud

Tambahkan sumber daya khusus untuk membuat pengguna IAM.	Tambahkan kode berikut ke `Resources` bagian CloudFormation template:
	`Resources:`
	`  rNewIamUser:`
		AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> Type: 'AWS::IAM
	::User'
	    Properties:
	      UserName: !Ref
	 NewIamUserName</pre>
	Kode ini menambahkan CloudFormation sumber daya yang menciptakan pengguna IAM dengan nama yang disediakan oleh `NewIamUserName` parameter.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan peran eksekusi untuk fungsi Lambda.	Pada langkah ini, Anda membuat peran IAM yang memberikan izin AWS Lambda fungsi untuk mendapatkan IAM. `UserId` Tentukan izin minimum yang diperlukan berikut untuk dijalankan Lambda: • `logs:CreateLogStream` • `logs:PutLogEvents` • `CreateLogGroup` • `iam:GetUser` • `AssumeRole` untuk `lambda.amazonaws.com` Untuk petunjuk tentang cara membuat peran eksekusi, lihat dokumentasi Lambda. Anda akan mereferensikan peran ini di langkah berikutnya, saat Anda membuat fungsi Lambda.	Administrator AWS, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan fungsi Lambda untuk mendapatkan IAM yang unik. `UserId`	Pada langkah ini, Anda mendefinisikan fungsi Lambda dengan runtime Python untuk mendapatkan IAM yang unik. `UserId` Untuk melakukan ini, tambahkan kode berikut ke `Resources` bagian CloudFormation templat. Ganti `<<ROLENAME>>` dengan nama peran eksekusi yang Anda buat di langkah terakhir.
	
	`GetUserLambdaFunction:`
	`Type: 'AWS::Lambda::Function'`
	`Properties:`
	`Handler: index.handler`
	`Role: <<ROLENAME>>`
	`Timeout: 30`
	`Runtime: python3.11`
	`Code:`
	`ZipFile: |`
	`import cfnresponse, boto3`
	`def handler(event, context):`
	`try:`
	`print(event)`
	`user = boto3.client('iam').get_user(UserName=event['ResourceProperties']['NewIamUserName'])['User']`
		AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>cfnrespon
	se.send(event,
	 context, cfnrespon
	se.SUCCESS, {'NewIamU
	serId': user['Use
	rId'], 'NewIamUs
	erPath': user['Pat
	h'], 'NewIamUserArn':
	 user['Arn']})
	except
	 Exception as e:
	cfnrespon
	se.send(event,
	 context, cfnrespon
	se.FAILED, {'NewIamU
	ser': str(e)})</pre>	

Tambahkan sumber daya khusus.	Tambahkan kode berikut ke `Resources` bagian CloudFormation template: <pre> rCustomGetUniqueUs erId: Type: 'Custom:: rCustomGetUniqueUs erIdWithLambda' Properties: ServiceToken: ! GetAtt GetUserLa mbdaFunction.Arn NewIamUserName: ! Ref NewIamUserName</pre> Sumber daya khusus ini memanggil fungsi Lambda untuk mendapatkan IAM. `UserID`	AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tentukan CloudFormation output.	Tambahkan kode berikut ke `Outputs` bagian CloudForm ation template:
	
	`Outputs:`
	`  NewIamUserId:`
	`  Value: !GetAtt`
	`  rCustomGetUniqueUs`
	`erId.NewIamUserId`
	
	Ini menampilkan IAM `UserID` untuk pengguna IAM baru.
		AWS DevOps, arsitek Cloud

Simpan template.	Simpan perubahan Anda ke CloudFormation template.	AWS DevOps, arsitek Cloud

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan CloudFormation template.	Untuk menerapkan `get_unique_user_id .yaml` template dengan menggunakan CloudForm ation konsol, ikuti instruksi dalam CloudFormation dokumentasi.
	
	Atau, Anda dapat menjalank an AWS CLI perintah berikut untuk menyebarkan template:
	
	`aws cloudformation`
	`  create-stack \`
		AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--stack-name DemoNewUser \`
	`--template-body file://get_unique_user_id.yaml \`
	`--parameters ParameterKey=NewIamUserName,ParameterValue=demouser \`
	`--capabilities CAPABILITY_NAMED_IAM`	

Sumber daya terkait

- Buat tumpukan dari CloudFormation konsol (CloudFormation dokumentasi)
- Sumber daya kustom yang didukung Lambda (dokumentasi) CloudFormation
- Pengidentifikasi unik (dokumentasi IAM)
- Gunakan kredensyal sementara dengan AWS sumber daya (dokumentasi IAM)

Validasi Account Factory untuk kode Terraform (AFT) secara lokal

Alexandru Pop dan Michal Gorniak, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menguji kode HashiCorp Terraform secara lokal yang dikelola oleh AWS Control Tower Account Factory for Terraform (AFT). Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud. AFT menyiapkan pipeline Terraform yang membantu Anda menyediakan dan menyesuaikan beberapa Akun AWS di. AWS Control Tower

Selama pengembangan kode, akan sangat membantu untuk menguji infrastruktur Terraform Anda sebagai kode (IAc) secara lokal, di luar pipa AFT. Pola ini menunjukkan bagaimana melakukan hal berikut:

- Ambil salinan lokal kode Terraform yang disimpan di AWS CodeCommit repositori di akun manajemen AFT Anda.
- Simulasikan pipeline AFT secara lokal dengan menggunakan kode yang diambil.

Prosedur ini juga dapat digunakan untuk menjalankan perintah Terraform yang bukan merupakan bagian dari pipeline AFT normal. Misalnya, Anda dapat menggunakan metode ini untuk menjalankan perintah seperti`terraform validate`,`terraform plan`,`terraform destroy`, dan`terraform import`.

Prasyarat dan batasan

Prasyarat

- Lingkungan AWS multi-akun aktif yang menggunakan AWS Control Tower
- Lingkungan AFT yang sepenuhnya digunakan
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- AWS CLI pembantu kredensi untuk AWS CodeCommit, diinstal dan dikonfigurasi
- Python 3.x
- Git, diinstal dan dikonfigurasi pada mesin lokal Anda
- `git-remote-commit`utilitas, diinstal dan dikonfigurasi

- Terraform, diinstal dan dikonfigurasi (versi paket Terraform lokal harus cocok dengan versi yang digunakan dalam penerapan AFT)

Batasan

- Pola ini tidak mencakup langkah-langkah penerapan yang diperlukan untuk AWS Control Tower, AFT, atau modul Terraform tertentu.
- Output yang dihasilkan secara lokal selama prosedur ini tidak disimpan di log runtime pipeline AFT.

Arsitektur

Tumpukan teknologi target

- Infrastruktur AFT dikerahkan dalam penerapan AWS Control Tower
- Terraform
- Git
- AWS CLI versi 2

Otomatisasi dan skala

Pola ini menunjukkan cara memanggil kode Terraform secara lokal untuk penyesuaian akun global AFT dalam satu yang dikelola AFT. Akun AWS Setelah kode Terraform Anda divalidasi, Anda dapat menerapkannya ke akun yang tersisa di lingkungan multi-akun Anda. Untuk informasi selengkapnya, lihat Memanggil ulang penyesuaian dalam dokumentasi. AWS Control Tower

Anda juga dapat menggunakan proses serupa untuk menjalankan kustomisasi akun AFT di terminal lokal. Untuk memanggil kode Terraform secara lokal dari penyesuaian akun AFT, kloning repositori alih-alih aft-account-customizationsrepositori dari akun manajemen AFT Anda. aft-global-account-customizations CodeCommit

Alat

Layanan AWS

- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif.

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

Layanan lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- Git adalah sistem kontrol versi terdistribusi open-source.

Kode

Berikut ini adalah contoh skrip bash yang dapat digunakan untuk menjalankan kode Terraform secara lokal yang dikelola oleh AFT. Untuk menggunakan skrip, ikuti instruksi di bagian Epik dari pola ini.

```
#! /bin/bash
# Version: 1.1 2022-06-24 Unsetting AWS_PROFILE since, when set, it interferes with
 script operation
#          1.0 2022-02-02 Initial Version
#
# Purpose: For use with AFT: This script runs the local copy of TF code as if it were
 running within AFT pipeline.
#        * Facilitates testing of what the AFT pipline will do
#           * Provides the ability to run terraform with custom arguments (like 'plan'
 or 'move') which are currently not supported within the pipeline.
#
# © 2021 Amazon Web Services, Inc. or its affiliates. All Rights Reserved.
# This AWS Content is provided subject to the terms of the AWS Customer Agreement
# available at http://aws.amazon.com/agreement or other written agreement between
# Customer and either Amazon Web Services, Inc. or Amazon Web Services EMEA SARL or
 both.
#
# Note: Arguments to this script are passed directly to 'terraform' without parsing nor
 validation by this script.
#
# Prerequisites:
#    1. local copy of ct GIT repositories
#    2. local backend.tf and aft-providers.tf filled with data for the target account
 on which terraform is to be run
#       Hint: The contents of above files can be obtain from the logs of a previous
 execution of the AFT pipeline for the target account.
#    3. 'terraform' binary is available in local PATH
```

```
#    4. Recommended: .gitignore file containing 'backend.tf', 'aft_providers.tf' so the
 local copy of these files are not pushed back to git

readonly credentials=$(aws sts assume-role \
    --role-arn arn:aws:iam::$(aws sts get-caller-identity --query "Account" --output
 text ):role/AWSAFTAdmin \
    --role-session-name AWSAFT-Session \
    --query Credentials )

unset AWS_PROFILE
export AWS_ACCESS_KEY_ID=$(echo $credentials | jq -r '.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo $credentials | jq -r '.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo $credentials | jq -r '.SessionToken')
terraform "$@"
```

Epik

Simpan kode contoh sebagai file lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Simpan kode contoh sebagai file lokal.	1. Salin contoh skrip bash yang ada di bagian Kode dari pola ini dan tempelkan ke editor kode. 2. Beri nama file `ct_terraform.sh`, lalu simpan file secara lokal di dalam folder khusus, seperti `~/scripts` atau `~/bin`.	Administrator AWS
Jadikan kode contoh dapat dijalankan.	Buka jendela terminal dan autentikasi ke akun manajemen AWS AFT Anda dengan melakukan salah satu hal berikut: • Gunakan AWS CLI profil yang sudah ada yang dikonfigurasi dengan	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	izin yang diperlukan untuk mengakses akun manajemen AFT. Untuk menggunakan profil, Anda dapat menjalankan perintah berikut: `export AWS_PROFILE=<aft account profile name>` • Jika organisasi Anda menggunakan SSO untuk mengakses AWS, masukkan kredensil untuk akun manajemen AFT Anda di halaman SSO organisasi Anda. **Note** Organisasi Anda mungkin juga memiliki alat khusus untuk memberikan kredensil otentikasi ke lingkungan Anda. AWS	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi akses ke akun manajemen AFT dengan benar Wilayah AWS.	**⚠ Important**
	Pastikan Anda menggunakan sesi terminal yang sama dengan yang Anda autentikasi ke akun manajemen AFT Anda.
	
	1. Arahkan ke penerapan AFT Anda Wilayah AWS dengan menjalankan perintah berikut:
	`export AWS_REGIO N=<aft_region>`
	2. Pastikan Anda berada di akun yang benar.
	a. Jalankan perintah berikut:
	`aws code-commit list-repositories`
	b. Verifikasi bahwa repositori yang tercantum dalam output cocok dengan nama repositor i yang ada di akun manajemen AFT Anda.
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat direktori lokal baru untuk menyimpan kode repositori AFT.	Dalam sesi terminal yang sama, jalankan perintah berikut:
	`mkdir my_aft`
	`cd my_aft`
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning kode repositori AFT jarak jauh.	1. Di terminal lokal Anda, jalankan perintah berikut: `git clone codecommit::$AWS_REGION://aft-global-customizations` **Note** Untuk mempermudah, prosedur ini dan AFT hanya menggunakan cabang kode utama. Untuk menggunakan percabangan kode, Anda dapat memasukkan perintah percabangan kode di sini juga. Namun, setiap perubahan yang diterapkan dari cabang non-utama akan dibatalkan ketika otomatisasi AFT menerapkan kode dari cabang utama. 2. Arahkan ke direktori kloning:	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`cd aft-global-customizations/terraform`	

Buat file konfigurasi Terraform yang diperlukan agar pipeline AFT berjalan secara lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka pipeline AFT yang dijalankan sebelumnya dan salin file konfigurasi Terraform ke folder lokal.	**Note** File `backend.tf` dan `aft-providers.tf` konfigurasi yang dibuat dalam epik ini diperlukan agar pipeline AFT dapat berjalan secara lokal. File-file ini dibuat secara otomatis dalam pipeline AFT berbasis cloud, tetapi harus dibuat secara manual agar pipeline dapat berjalan secara lokal. Menjalankan pipeline AFT secara lokal memerlukan satu set file yang mewakili menjalankan pipeline dalam satu Akun AWS.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Menggunakan kredensil akun AWS Control Tower manajemen Anda, masuk ke Konsol Manajemen AWS, dan buka konsol.AWS CodePipeline Pastikan Anda berada di Wilayah AWS tempat yang sama di mana Anda menggunakan AFT. 2. Di panel navigasi kiri, pilih Pipelines. 3. Pilih ##########-penyesuaian-pipa. (########## adalah Akun AWS ID yang Anda gunakan untuk menjalankan kode Terraform secara lokal.) 4. Pastikan bahwa Most Recent Execution Marked menunjukkan nilai Succeeded. Jika nilainya berbeda, Anda harus memanggil kembali penyesuaian Anda di pipeline AFT. Untuk informasi selengkapnya, lihat Memanggil ulang penyesuaian dalam dokumentasi. AWS Control Tower 5. Pilih runtime terbaru untuk memunculkan detailnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	6. Di bagian Apply-AFT-Global-Customizations, temukan tahap Apply-Terraform.
	7. Pilih bagian Detail dari tahap Terapkan-Terraform.
	8. Temukan log runtime untuk tahap Apply-Terraform.
	9. Di log runtime, cari bagian yang dimulai dan diakhiri dengan baris berikut:
	`"\n\n aft-providers.tf ... "\n \n backend.tf"`
	10. Salin output antara dua label ini dan simpan sebagai file lokal bernama `aft-providers.tf` dalam folder Terraform lokal (direktori kerja sesi terminal Anda saat ini).
	Contoh pernyataan providers.tf yang dihasilkan secara otomatis
	`## Autogenerated providers.tf ## ## Updated on: 2022-05-31 16:27:45 ## provider "aws" { region = "us-east-2"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

```
  assume_role {
    role_arn    =
 "arn:aws:iam::####
########:role/AWSA
FTExecution"
  }
  default_tags {
    tags = {
      managed_by
             = "AFT"
    }
  }
}
```

11. Di log runtime, cari bagian yang dimulai dan diakhiri dengan baris berikut:

```
"\n\n tf
...
"\n \n backend.tf"
```

12. Salin output antara dua label ini dan simpan sebagai file lokal bernama `tf` dalam folder Terraform lokal (direktori kerja sesi terminal Anda saat ini).

Contoh pernyataan backend.tf yang dibuat secara otomatis

```
## Autogenerated
 backend.tf ##
## Updated on: 2022-05-3
1 16:27:45 ##
terraform {
  required_version =
 ">= 0.15.0"
```

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> backend "s3" {
	    region =
	 "us-east-2"
	    bucket =
	 "aft-backend-#####
	#######-primary-re
	gion"
	    key =
	 "############-aft-
	global-customizati
	ons/terraform.tfst
	ate"
	    dynamodb_table =
	 "aft-backend-#####
	#######"
	    encrypt =
	 "true"
	    kms_key_id =
	 "########-####-####-
	####-############"
	    role_arn =
	 "arn:aws:iam::####
	#########:role/AWS
	AFTExecution"
	  }
	}</pre>
	Note
	aft-providers.tf File backend.tf dan terkait dengan penyebaran AFT Akun AWS, dan folder tertentu. File-file ini juga berbeda, tergantung pada apakah	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mereka berada di aft-account-customizationsrepositori aft-global-customizationsdan dalam penerapan AFT yang sama. Pastikan Anda membuat kedua file dari daftar runtime yang sama.	

Jalankan pipeline AFT secara lokal dengan menggunakan contoh skrip bash

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan perubahan konfigurasi Terraform yang ingin Anda validasi.	1. Arahkan ke aft-global-customizationsrepositori kloning dengan menjalankan perintah berikut: `cd aft-global-customizations/terraform` **Note** File `backend.tf` dan `aft-providers.tf` berada di direktori ini. Direktori ini juga berisi file Terraform dari repositori. aft-	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	global-customi zations 2. Gabungkan perubahan kode Terraform yang ingin Anda uji secara lokal ke dalam file konfigurasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan `ct_terraform.sh` skrip dan tinjau outputnya.	1. Arahkan ke folder lokal yang berisi skrip sh.
	2. Untuk memvalidasi kode Terraform Anda yang dimodifikasi, jalankan `ct_terraform.sh` skrip dengan menjalankan perintah berikut:
	`~/scripts/ct_terraform.sh apply`
	`terraform --help`
	Note
	Anda dapat menjalankan perintah Terraform apa pun selama langkah ini. Untuk melihat daftar lengkap perintah Terraform, jalankan perintah berikut:
	3. Tinjau output perintah, lalu debug perubahan kode secara lokal sebelum melakukan dan mendorong perubahan kembali ke repositori AFT.
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important • Setiap perubahan yang dilakukan secara lokal dan tidak didorong kembali ke repositori jarak jauh bersifat sementara dan dapat dibatalkan kapan saja oleh otomatisasi pipa AFT yang sedang berjalan. • Otomatisasi AFT dapat berjalan kapan saja, karena dapat dipanggil oleh pengguna lain dan pemicu otomatisasi AFT. • AFT akan selalu menerapkan kode dari cabang utama repositori, membatalkan perubahan yang tidak berkomitmen.	

Dorong perubahan kode lokal Anda kembali ke repositori AFT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan referensi ke `aft-providers.tf` file `backend.tf` dan file ke `.gitignore` file.	Tambahkan `backend.tf` dan `aft-providers.tf` file yang Anda buat ke `.gitignore` file dengan menjalankan perintah berikut:	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>echo backend.tf
	 >> .gitignore
	echo aft-providers.tf
	 >>.gitignore</pre>
	Note
	Memindahkan file ke `.gitignore` file memastikan bahwa mereka tidak dikomit dan didorong kembali ke repositori AFT jarak jauh.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Komit dan dorong perubahan kode Anda ke repositori AFT jarak jauh.	1. Untuk menambahkan file konfigurasi Terraform baru ke repositori, jalankan perintah berikut:
	`git add <filename>`
	2. Untuk melakukan perubahan dan mendorong nya ke repositori AFT jarak jauh CodeCommitt, jalankan perintah berikut:
	`git commit -a`
	`git push`
	Important
	Perubahan kode yang Anda perkenalkan dengan mengikuti prosedur ini hingga titik ini diterapkan Akun AWS hanya untuk satu.
		Administrator AWS

Meluncurkan perubahan ke beberapa akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan perubahan ke semua akun Anda yang dikelola oleh AFT.	Untuk meluncurkan perubahan ke beberapa Akun AWS yang dikelola oleh AFT,	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ikuti petunjuk di Re-invoke kustomisasi dalam dokumenta si. AWS Control Tower	

Lebih banyak pola

- Tambahkan HA ke Oracle PeopleSoft di Amazon RDS Custom dengan menggunakan replika baca
- Secara otomatis mengaudit grup AWS keamanan yang memungkinkan akses dari alamat IP publik
- Mengotomatiskan pembuatan akun dengan menggunakan Akselerator Zona Landing di AWS
- Otomatis menambahkan atau memperbarui entri registri Windows menggunakan AWS Systems Manager
- Mengotomatiskan penilaian sumber daya AWS
- Mengotomatiskan portofolio AWS Service Catalog dan penerapan produk dengan menggunakan AWS CDK
- Otomatiskan failover dan failback lintas wilayah dengan menggunakan DR Orchestrator Framework
- Mengotomatiskan penghapusan AWS CloudFormation tumpukan dan sumber daya terkait
- Otomatiskan konsumsi dan visualisasi metrik kustom Amazon MWAA di Amazon Managed Grafana dengan menggunakan Terraform
- Otomatiskan konfigurasi RabbitMQ di Amazon MQ
- Otomatiskan penerapan Rantai Pasokan AWS data lake dalam pengaturan multi-repositori dengan menggunakan GitHub Actions, Artifactory, dan Terraform
- Otomatiskan replikasi instans Amazon RDS di seluruh Akun AWS
- Secara otomatis melampirkan kebijakan terkelola AWS untuk Systems Manager ke profil EC2 instans menggunakan Cloud Custodian dan AWS CDK
- Membuat CI/CD pipeline dan kluster Amazon ECS secara otomatis untuk layanan mikro menggunakan AWS CDK
- Secara otomatis mendeteksi perubahan dan memulai CodePipeline jaringan pipa yang berbeda untuk monorepo di CodeCommit
- Membangun pipeline data untuk menyerap, mengubah, dan menganalisis data Google Analytics menggunakan AWS DataOps Development Kit
- Bangun PAC Server Perusahaan Fokus Mikro dengan Amazon EC2 Auto Scaling dan Systems Manager
- Buat dan dorong gambar Docker ke Amazon ECR menggunakan GitHub Actions dan Terraform
- Bangun AWS landing zone yang mencakup MongoDB Atlas
- Memusatkan manajemen kunci akses IAM di AWS Organizations dengan menggunakan Terraform

- Memusatkan distribusi paket perangkat lunak di AWS Organizations dengan menggunakan Terraform
- Konfigurasikan pencatatan pemanggilan model di Amazon Bedrock dengan menggunakan AWS CloudFormation
- Konfigurasikan perutean hanya-baca dalam grup ketersediaan Selalu Aktif di SQL Server di AWS
- Buat portal untuk frontend mikro dengan menggunakan AWS Amplify, Angular, dan Federasi Modul
- Buat kerangka orkestrasi sumber daya berbasis API menggunakan Actions dan Terragrunt GitHub
- Membuat EventBridge koneksi Amazon lintas akun di suatu organisasi
- Buat pipeline CI dinamis untuk proyek Java dan Python secara otomatis
- Menerapkan API Amazon API Gateway di situs web internal menggunakan endpoint pribadi dan Application Load Balancer
- Menyebarkan dan mengelola AWS Control Tower kontrol dengan menggunakan AWS CDK dan CloudFormation
- Terapkan dan kelola AWS Control Tower kontrol dengan menggunakan Terraform
- Menyebarkan kenari CloudWatch Synthetics dengan menggunakan Terraform
- Menerapkan cluster CockroachDB di Amazon EKS dengan menggunakan Terraform
- Menerapkan sistem file Lustre untuk pemrosesan data berkinerja tinggi dengan menggunakan Terraform dan DRA
- Terapkan kasus penggunaan RAG AWS dengan menggunakan Terraform dan Amazon Bedrock
- Menerapkan sumber daya di AWS Wavelength Zona dengan menggunakan Terraform
- Menerapkan instance cluster failover SQL Server di Amazon EC2 dan Amazon FSx dengan menggunakan Terraform
- Terapkan Otomatisasi Keamanan untuk AWS WAF solusi dengan menggunakan Terraform
- Mendeteksi instans database Amazon RDS dan Aurora yang memiliki sertifikat CA kedaluwarsa
- Dokumentasikan desain AWS landing zone
- Ekspor laporan AWS Backup dari seluruh organisasi di AWS Organizations sebagai file CSV
- Buat rekomendasi yang dipersonalisasi dan diberi peringkat ulang menggunakan Amazon Personalize
- Mengatur set izin untuk beberapa akun dengan menggunakan Account Factory untuk Terraform
- Identifikasi dan beri tahu kapan sumber daya Amazon Data Firehose tidak dienkripsi dengan kunci AWS KMS
- Implementasikan Account Factory for Terraform (AFT) dengan menggunakan pipeline bootstrap

- Menerapkan versi API berbasis jalur dengan menggunakan domain khusus di Amazon API Gateway
- Instal Agen SSM di node pekerja Amazon EKS dengan menggunakan Kubernetes DaemonSet
- Instal Agen dan CloudWatch agen SSM di node pekerja Amazon EKS menggunakan preBootstrapCommands
- Kelola set AWS IAM Identity Center izin sebagai kode dengan menggunakan AWS CodePipeline
- Kelola set AWS izin secara dinamis dengan menggunakan Terraform
- Kelola produk AWS Service Catalog di beberapa akun AWS dan Wilayah AWS
- Mengelola aplikasi kontainer lokal dengan menyiapkan Amazon ECS Anywhere dengan AWS CDK
- Mengelola AWS Organizations kebijakan sebagai kode dengan menggunakan AWS CodePipeline dan Amazon Bedrock
- Migrasikan data DNS secara massal ke zona host pribadi Amazon Route 53
- Migrasi Oracle ke PeopleSoft Amazon RDS Kustom
- Migrasikan sistem RHEL BYOL ke instans AWS License-Included dengan menggunakan AWS MGN
- Siapkan ruang data minimum yang layak untuk berbagi data antar organisasi
- Pantau ElastiCache kluster Amazon untuk enkripsi saat istirahat
- Memantau aktivitas aplikasi dengan menggunakan Wawasan CloudWatch Log
- Pantau cluster SAP RHEL Pacemaker dengan menggunakan layanan AWS
- Buat arsitektur IPAM Multi-wilayah yang hierarkis dengan menggunakan Terraform AWS
- Optimalkan penerapan tanpa server multi-akun dengan menggunakan alur kerja dan Tindakan AWS CDK GitHub
- Menyediakan AWS Service Catalog produk berdasarkan AWS CloudFormation template dengan menggunakan GitHub Actions
- Menyediakan peran IAM dengan hak istimewa paling sedikit dengan menerapkan solusi mesin penjual otomatis peran
- Hapus EC2 entri Amazon di Akun AWS seberang AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi
- Hapus EC2 entri Amazon dalam hal yang sama Akun AWS dari AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi
- Amankan transfer file dengan menggunakan Transfer Family, Amazon Cognito, dan GuardDuty
- Kirim pemberitahuan saat pengguna IAM dibuat

- Siapkan router sel tanpa server untuk arsitektur berbasis sel
- Siapkan CI/CD pipeline untuk beban kerja hybrid di Amazon ECS Anywhere dengan menggunakan AWS CDK dan GitLab
- Menyiapkan HA/DR arsitektur untuk Oracle E-Business Suite di Amazon RDS Custom dengan database siaga aktif
- Mengatur resolusi DNS untuk jaringan hybrid di lingkungan multi-akun AWS
- Menyiapkan infrastruktur Multi-AZ untuk SQL Server Always On FCI dengan menggunakan Amazon FSx
- Siapkan fungsionalitas Oracle UTL_FILE pada Aurora PostgreSQL yang kompatibel
- Sederhanakan otentikasi aplikasi dengan TLS timbal balik di Amazon ECS dengan menggunakan Application Load Balancer
- Sederhanakan manajemen sertifikat pribadi dengan menggunakan AWS Private CA dan AWS RAM
- Merampingkan alur kerja pembelajaran mesin dari pengembangan lokal hingga eksperimen yang dapat diskalakan dengan menggunakan SageMaker AI dan Hydra
- Tandai lampiran Transit Gateway secara otomatis menggunakan AWS Organizations
- Peran transisi untuk PeopleSoft aplikasi Oracle di Amazon RDS Custom for Oracle
- Gunakan Amazon Q Developer sebagai asisten pengkodean untuk meningkatkan produktivitas Anda

Aplikasi web & seluler

Topik

- Mengautentikasi pengguna aplikasi React yang ada dengan menggunakan Amazon Cognito dan AWS Amplify UI
- Buat aplikasi React dengan menggunakan AWS Amplify dan tambahkan autentikasi dengan Amazon Cognito
- Buat portal untuk frontend mikro dengan menggunakan AWS Amplify, Angular, dan Federasi Modul
- Menerapkan aplikasi satu halaman berbasis React ke Amazon S3 dan CloudFront
- Menerapkan API Amazon API Gateway di situs web internal menggunakan endpoint pribadi dan Application Load Balancer
- Sematkan komponen visual Amazon Quick Sight ke dalam aplikasi web dengan menggunakan otomatisasi Amazon Cognito dan IAc
- Jelajahi pengembangan aplikasi web cloud-native full-stack dengan Green Boost
- Jalankan pengujian unit untuk aplikasi Node.js dari GitHub dengan menggunakan AWS CodeBuild
- Struktur proyek Python dalam arsitektur heksagonal menggunakan AWS Lambda
- Lebih banyak pola

Mengautentikasi pengguna aplikasi React yang ada dengan menggunakan Amazon Cognito dan AWS Amplify UI

Daniel Kozhemyako, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menambahkan kemampuan otentikasi ke aplikasi React frontend yang ada dengan menggunakan pustaka AWS Amplify UI dan kumpulan pengguna Amazon Cognito.

Pola ini menggunakan Amazon Cognito untuk menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi. Ini juga menggunakan komponen dari Amplify UI, perpustakaan sumber terbuka yang memperluas kemampuan pengembangan AWS Amplify antarmuka pengguna (UI). Komponen Authenticator UI mengelola sesi login dan menjalankan alur kerja yang terhubung dengan cloud yang mengautentikasi pengguna melalui Amazon Cognito.

Setelah Anda menerapkan pola ini, pengguna dapat masuk dengan menggunakan salah satu kredensi berikut:

- Nama pengguna dan kata sandi
- Penyedia identitas sosial, seperti Apple, Facebook, Google, dan Amazon
- Penyedia identitas perusahaan yang kompatibel dengan SAFL 2.0 atau kompatibel dengan OpenID Connect (OIDC)

> **Note**
>
> Untuk membuat komponen UI otentikasi kustom, Anda dapat menjalankan komponen Authenticator UI dalam mode headless.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- React 18.2.0 atau aplikasi web yang lebih baru
- Node.js dan npm 6.14.4 atau yang lebih baru, diinstal

Batasan

- Pola ini hanya berlaku untuk aplikasi web React.
- Pola ini menggunakan komponen Amplify UI bawaan. Solusinya tidak mencakup langkah-langkah yang diperlukan untuk mengimplementasikan komponen UI khusus.

Versi produk

- Amplify UI 6.1.3 atau yang lebih baru (Gen 1)
- Amplify 6.0.16 atau yang lebih baru (Gen 1)

Arsitektur

Arsitektur target

Diagram berikut menunjukkan arsitektur yang menggunakan Amazon Cognito untuk mengautentikasi pengguna untuk aplikasi web React.

Alat

Layanan AWS

- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.

Alat-alat lainnya

- Amplify UI adalah pustaka UI open-source yang menyediakan komponen yang dapat disesuaikan yang dapat Anda sambungkan ke cloud.
- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.
- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.

Praktik terbaik

Jika Anda sedang membangun aplikasi baru, kami sarankan Anda menggunakan Amplify Gen 2.

Epik

Buat kumpulan pengguna Amazon Cognito

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat pengguna.	Buat kumpulan pengguna Amazon Cognito. Konfigura sikan opsi masuk dan persyaratan keamanan kumpulan pengguna agar sesuai dengan kasus penggunaan Anda.	Pengembang aplikasi
Tambahkan klien aplikasi.	Konfigurasikan klien aplikasi kumpulan pengguna. Klien ini diperlukan agar aplikasi Anda berinteraksi dengan kumpulan pengguna Amazon Cognito.	Pengembang aplikasi

Integrasikan kumpulan pengguna Amazon Cognito Anda dengan komponen UI Authenticator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dependensi.	Untuk menginstal `aws-amplify` dan `@aws-amplify/ui-react` paket, jalankan perintah berikut dari direktori root aplikasi Anda: `npm i @aws-amplify/ui-react aws-amplify`	Pengembang aplikasi
Konfigurasikan kumpulan pengguna.	Berdasarkan contoh berikut, buat `aws-exports.js` file dan simpan di `src` folder. File	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	harus menyertakan informasi berikut:
	• Wilayah AWS tempat kumpulan pengguna Amazon Cognito Anda berada
	• ID kumpulan pengguna Amazon Cognito
	• ID klien aplikasi
	
	`// replace the user pool region, id, and app client id details`
	`const awsmobile = {`
	`"aws_project_region": "put_your_region_here",`
	`"aws_cognito_region": "put_your_region_here",`
	`"aws_user_pools_id": "put_your_user_pool_id_here",`
	`"aws_user_pools_web_client_id": "put_your_user_pool_app_id_here"`
	`}`
	
	`export default awsmobile;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Impor dan konfigurasikan layanan Amplify.	1. Dalam file titik masuk aplikasi (misalnya,`App.js`), impor dan muat `aws-exports.js` file dengan memasukkan baris kode berikut:
	`import { Amplify } from 'aws-amplify'; import awsExports from './aws-exports';`
	2. Berdasarkan contoh berikut, konfigurasikan klien Amplify dengan menggunakan file: `aws-exports.js`
	`// Configure Amplify in index file or root file Amplify.configure({ ...awsExports });`
	Untuk informasi selengkapnya, lihat Mengonfigurasi kategori Amplify di dokumentasi Amplify.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan komponen Authenticator UI.	Untuk menampilkan komponen `Authenticator` UI, tambahkan baris kode berikut ke file titik masuk aplikasi (`App.js`): `import { Authentic ator } from '@aws-amp lify/ui-react'; import '@aws-amplify/ui-react/styles.css';` **Note** Contoh cuplikan kode mengimpor komponen `Authenticator` UI dan file styles.cs s Amplify UI, yang diperlukan saat menggunakan tema bawaan komponen. Komponen `Authenticator` UI menyediakan dua nilai kembali: • Detail pengguna • Fungsi yang dapat dipanggil untuk mengeluarkan pengguna Lihat contoh komponen berikut:	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>function App() {
	    return (
	        <Authenticator>
	            {({ signOut,
	 user }) => (
	
	 Welcome {user.use
	rname}
	
	  <button onClick={
	signOut}>Sign out</
	button>
	
	            )}
	        </Authent
	icator>
	    );
	}</pre>
	Note
	Untuk `App.js` file contoh, lihat bagian Informasi tambahan dari pola ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Ambil informasi sesi.	Setelah pengguna diautenti kasi, Anda dapat mengambil data dari klien Amplify tentang sesi mereka. Misalnya, Anda dapat mengambil token web JSON (JWT) dari sesi pengguna sehingga Anda dapat mengautentikasi permintaan dari sesi mereka ke API backend. Lihat contoh header permintaa n berikut yang menyertakan JWT: `import { fetchAuth Session } from 'aws- amplify/auth'; (await fetchAuth Session()).tokens? .idToken?.toString();`	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Pengguna baru tidak dapat mendaftar untuk aplikasi.	Sebagai berikut, pastikan kumpulan pengguna Amazon Cognito Anda dikonfigurasi untuk memungkinkan pengguna mendaftar sendiri untuk berada di kumpulan pengguna: • Masuk ke Konsol Manajemen AWS, lalu buka konsol Amazon Cognito. • Di panel navigasi kiri, pilih Kumpulan pengguna.

Isu	Solusi
	• Pilih kumpulan pengguna Anda dari daftar. • Di bawah Pengaturan Umum, pilih Kebijakan. • Pilih Izinkan pengguna untuk mendaftar sendiri.
Komponen autentikasi berhenti bekerja setelah memutakhirkan dari v5 ke v6.	`Auth`Kategori telah pindah ke pendekata n fungsional dan menamai parameter di Amplify v6. Anda sekarang harus mengimpor fungsional APIs langsung dari `aws-ampli fy/auth` jalur. Untuk informasi selengkapnya, lihat Memigrasi dari v5 ke v6 di dokumentasi Amplify.

Sumber daya terkait

- Memulai dengan Amazon Cognito (situs web)AWS
- Buat aplikasi React baru (dokumentasi React)
- Apa itu Amazon Cognito? (Dokumentasi Amazon Cognito)
- Amplify pustaka UI (Amplify dokumentasi)

Informasi tambahan

`App.js`File harus berisi kode berikut:

```
import './App.css';
import { Amplify } from 'aws-amplify';
import awsExports from './aws-exports';
import { fetchAuthSession } from 'aws-amplify/auth';
import { Authenticator } from '@aws-amplify/ui-react';
import '@aws-amplify/ui-react/styles.css';
Amplify.configure({ ...awsExports });
let token = (await fetchAuthSession()).tokens?.idToken?.toString();
function App() {
  return (
      <Authenticator>
        {({ signOut, user }) => (
```

```
            <div>
              <p>Welcome {user.username}</p>
                <p>Your token is: {token}</p>
              <button onClick={signOut}>Sign out</button>
            </div>
        )}
      </Authenticator>
  );
}

export default App;
```

Buat aplikasi React dengan menggunakan AWS Amplify dan tambahkan autentikasi dengan Amazon Cognito

Rishi Singla, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menggunakan AWS Amplify untuk membuat aplikasi berbasis React dan cara menambahkan otentikasi ke frontend dengan menggunakan Amazon Cognito. AWS Amplify terdiri dari seperangkat alat (kerangka open source, lingkungan pengembangan visual, konsol) dan layanan (aplikasi web dan hosting situs web statis) untuk mempercepat pengembangan aplikasi seluler dan web di AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Node.js dan npm diinstal pada mesin Anda

Versi produk

- Node.js versi 10.x atau yang lebih baru (untuk memverifikasi versi Anda, jalankan `node -v` di jendela terminal)
- npm versi 6.x atau yang lebih baru (untuk memverifikasi versi Anda, jalankan `npm -v` di jendela terminal)

Arsitektur

Tumpukan teknologi target

- AWS Amplify
- Amazon Cognito

Alat

- Amplify Antarmuka Baris Perintah (CLI)

- Amplify Libraries (pustaka klien open source)
- Amplify Studio (antarmuka visual)

Epik

Instal AWS Amplify CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal Amplify CLI.	Amplify CLI adalah toolchain terpadu untuk membuat layanan AWS Cloud untuk aplikasi React Anda. Untuk menginstal Amplify CLI, jalankan:
	`npm install -g @aws-amplify/cli`
	npm akan memberi tahu Anda jika versi utama baru tersedia. Jika demikian, gunakan perintah berikut untuk memutakhirkan versi npm Anda:
	`npm install -g npm@9.8.0`
	di mana 9.8.0 mengacu pada versi yang ingin Anda instal.
		Pengembang aplikasi

Buat aplikasi React

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat aplikasi React.	Untuk membuat aplikasi React baru, gunakan perintah:
	`npx create-react-app amplify-react-application`
	di `ampify-react-application` mana nama aplikasinya.
	Ketika aplikasi telah berhasil dibuat, Anda akan melihat pesan:
	`Success! Created amplify-react-application`
	Direktori dengan berbagai subfolder akan dibuat untuk aplikasi React.
		Pengembang aplikasi

Luncurkan aplikasi di mesin lokal Anda.	Buka direktori `amplify-react-application` yang dibuat pada langkah sebelumnya dan jalankan perintah: `amplify-react-application% npm start` Ini meluncurkan aplikasi React di mesin lokal Anda.	Pengembang aplikasi

Konfigurasikan Amplify CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Amplify untuk terhubung ke akun AWS Anda.	Konfigurasikan Amplify dengan menjalankan perintah: `amplify-react-application % amplify configure` Amplify CLI meminta Anda untuk mengikuti langkah-langkah ini untuk mengatur akses ke akun AWS Anda: 1. Masuk ke akun administrator AWS Anda. 2. Tentukan Wilayah AWS yang ingin Anda gunakan. 3. Buat pengguna AWS Identity and Access Management (IAM) dengan akses terprogram, dan lampirkan kebijakan `AdministratorAccess-Amplify` izin ke pengguna. 4. Buat dan kemudian salin ID kunci akses dan kunci akses rahasia. 5. Masukkan detail ini di terminal. 6. Buat nama profil atau gunakan profil default.	AWS Umum, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Warning Skenario ini mengharuskan pengguna IAM dengan akses terprogram dan kredensil jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan. Kunci akses dapat diperbarui jika perlu. Untuk informasi selengkapnya, lihat Memperbarui kunci akses di Panduan Pengguna IAM. Langkah-langkah ini muncul di terminal sebagai berikut.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`Follow these steps to set up access to your AWS account:`
	`Sign in to your AWS administrator account: https://console.aws.amazon.com/`
	`Press Enter to continue`
	`Specify the AWS Region`
	`? region:  us-east-1`
	`Follow the instructions at https://docs.amplify.aws/cli/start/install/#configure-the-amplify-cli`
	`to complete the user creation in the AWS console`
	`https://console.aws.amazon.com/iamv2/home#/users/create`
	`Press Enter to continue`
	`Enter the access key of the newly created user:`
	`? accessKeyId: ********************`
	`? secretAccessKey: ********************`
	`This would update/create the AWS Profile in your local machine`
	`? Profile Name:  new`
	
	`Successfully set up the new user.`
	
	Untuk informasi selengkapnya tentang langkah-langkah ini,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lihat dokumentasi di Amplify Dev Center.	

Inisialisasi Amplify

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Inisialisasi Amplify.	1. Untuk menginisialisasi Amplify di direktori baru, jalankan:
	`amplify init`
	Amplify meminta Anda untuk nama proyek dan parameter konfigurasi
	2. Tentukan semua parameter , lalu tekan Y untuk menginisialisasi proyek dengan konfigurasi yang ditentukan.
	`Project information`
	`| Name: amplifyre`
	`actproject`
	`| Environment: dev`
	`| Default editor:`
	` Visual Studio Code`
	`| App type: javascrip`
	`t`
	`| Javascript`
	` framework: react`
		Pengembang aplikasi, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>| Source Directory
	Path: src
	
	| Distribution
	Directory Path: build
	
	| Build Command: npm
	run-script build
	
	| Start Command: npm
	run-script start</pre>
	3. Pilih profil yang Anda buat pada langkah sebelumnya. Sumber daya akan diterapkan ke `dev` lingkungan dalam proyek Amplify yang Anda buat.
	4. Untuk mengonfirmasi bahwa sumber daya telah dibuat, Anda dapat membuka konsol AWS Amplify dan melihat CloudFormation templat AWS yang digunakan untuk membuat sumber daya dan detailnya.
	
	<pre>Deploying root stack
	amplifyreactproject
	[==========
	==========
	----] 2/4
	amplify-amplif
	yreactproject-d…
	AWS::CloudFormatio
	n::Stack
	CREATE_IN_PROGRESS</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`UnauthRole AWS::IAM::Role CREATE_COMPLETE`
	
	`DeploymentBucket AWS::S3::Bucket CREATE_IN_PROGRESS`
	
	`AuthRole AWS::IAM::Role CREATE_COMPLETE`	

Tambahkan otentikasi ke frontend

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menambahkan otentikasi.	Anda dapat menggunakan `amplify add <category>` perintah untuk menambahkan fitur seperti login pengguna atau API backend. Pada langkah ini Anda akan menggunakan perintah untuk menambahkan otentikasi. Amplify menyediakan layanan otentikasi backend dengan Amazon Cognito, pustaka frontend, dan komponen UI Authenticator drop-in. Fitur termasuk pendaftaran	Pengembang aplikasi, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengguna, masuk pengguna, otentikasi multi-faktor, masuk pengguna, dan masuk tanpa kata sandi. Anda juga dapat mengautentikasi pengguna dengan mengintegrasikan dengan penyedia identitas federasi seperti Amazon, Google, dan Facebook. Kategori otentikasi Amplify terintegrasi secara mulus dengan kategori Amplify lainnya seperti API, analitik, dan penyimpanan, sehingga Anda dapat menentuka n aturan otorisasi untuk pengguna yang diautentikasi dan tidak diautentikasi. 1. Untuk mengonfigurasi autentikasi aplikasi React Anda, jalankan perintah: `amplify-react-appl ication1 % amplify add auth` Ini menampilkan informasi dan petunjuk berikut. Anda dapat memilih konfigura si yang sesuai tergantung pada kebutuhan bisnis dan keamanan Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>Using service:
	 Cognito, provided by:
	 awscloudformation
	 The current configure
	d provider is Amazon
	 Cognito.
	Do you want to use the
	 default authentic
	ation and security
	 configuration? (Use
	 arrow keys)
	
	# Default configura
	tion
	
	 Default configura
	tion with Social
	 Provider (Federati
	on)
	
	 Manual configura
	tion
	
	 I want to learn
	 more.</pre>	

	2. Untuk contoh sederhana, pilih konfigurasi default dan kemudian pilih mekanisme masuk untuk pengguna (dalam hal ini, email):	

	<pre>How do you want users
	 to be able to sign
	 in?
	 Username
	
	# Email
	
	 Phone Number</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> Email or Phone Number
	
	  I want to learn more.</pre>
	3. Lewati pengaturan lanjutan untuk menyelesaikan penambahan sumber daya otentikasi:
	<pre>Do you want to configure advanced settings? (Use arrow keys)
	# No, I am done.
	
	  Yes, I want to make some additional changes.</pre>
	4. Bangun sumber daya backend lokal Anda dan sediakan di cloud:
	<pre>amplify-react-appl ication1 % amplify push</pre>
	Perintah ini membuat perubahan yang sesuai pada kumpulan pengguna Congito di akun Anda.
	5. Tekan Y untuk mengkonfi gurasi `auth` sumber daya dengan menggunakan CloudFormation.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Ini mengonfigurasi sumber daya berikut:
	
	`UserPool AWS::Cognito::UserPool CREATE_COMPLETE`
	`UserPoolClientWeb AWS::Cognito::UserPoolClient CREATE_COMPLETE`
	
	`UserPoolClientWeb AWS::Cognito::UserPoolClient CREATE_COMPLETE`
	
	`UserPoolClientRole AWS::IAM::Role CREATE_COMPLETE`
	
	`UserPoolClientLambda AWS::Lambda::Function CREATE_COMPLETE`
	`UserPoolClientLambdaPolicy AWS::IAM::Policy CREATE_COMPLETE`
	
	`UserPoolClientLogPolicy AWS::IAM::Policy CREATE_IN_PROGRESS`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda juga dapat menggunakan konsol AWS Cognito untuk melihat sumber daya ini (cari kumpulan pengguna Cognito dan kumpulan identitas). Langkah ini memperbar ui `aws-exports.js` file di `src` folder untuk aplikasi React Anda dengan kumpulan pengguna Cognito dan konfigurasi kumpulan identitas.	

Ubah file App.js

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah file App.js.	Di `src` folder, buka dan revisi `App.js` file. File yang dimodifikasi akan terlihat seperti ini: `{  App.Js File after modifications: import React from 'react'; import logo from './logo.svg'; import './App.css'; import { Amplify } from 'aws-amplify'; import { withAuthenticator, Button,`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>Heading } from '@aws-amplify/ui-react';
	import awsconfig from './aws-exports';
	Amplify.configure(awsconfig);
	function App({ signOut }) {
	return (
	
	 Thankyou for doing verification
	 My Content
	<button onClick={signOut}>Sign out</button>
	
	);
	}
	export default withAuthenticator(App);</pre>	

Impor paket React.	`App.js`File mengimpor dua paket React. Instal paket-paket ini dengan menggunakan perintah: <pre>amplify-react-application1 % npm install --save aws-amplify @aws-amplify/ui-react</pre>	Pengembang aplikasi

Luncurkan aplikasi React dan periksa otentikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan aplikasi.	Luncurkan aplikasi React di mesin lokal Anda: `amplify-react-appl ication1 % npm start`	Pengembang aplikasi, AWS Umum
Periksa otentikasi.	Periksa apakah aplikasi meminta parameter otentikas i. (Dalam contoh kami, kami telah mengonfigurasi email sebagai metode masuk.) UI frontend harus meminta Anda untuk kredensi masuk dan memberikan opsi untuk membuat akun. Anda juga dapat mengonfig urasi proses build Amplify untuk menambahkan backend sebagai bagian dari alur kerja penerapan berkelanj utan. Namun, pola ini tidak mencakup opsi itu.	Pengembang aplikasi, AWS Umum

Sumber daya terkait

- Memulai (dokumentasi npm)
- Buat akun AWS mandiri (dokumentasi AWS Account Management)
- Dokumentasi AWS Amplify
- Dokumentasi Amazon Cognito

Buat portal untuk frontend mikro dengan menggunakan AWS Amplify, Angular, dan Federasi Modul

Milena Godau dan Pedro Garcia, Amazon Web Services

Ringkasan

Arsitektur micro-frontend memungkinkan beberapa tim untuk bekerja pada bagian yang berbeda dari aplikasi frontend secara independen. Setiap tim dapat mengembangkan, membangun, dan menyebarkan fragmen frontend tanpa mengganggu bagian lain dari aplikasi. Dari perspektif pengguna akhir, tampaknya menjadi aplikasi tunggal yang kohesif. Namun, mereka berinteraksi dengan beberapa aplikasi independen yang diterbitkan oleh tim yang berbeda.

Dokumen ini menjelaskan cara membuat arsitektur micro-frontend dengan menggunakan AWS Amplify, framework frontend Angular, dan Module Federation. Dalam pola ini, frontend mikro digabungkan di sisi klien oleh aplikasi shell (atau induk). Aplikasi shell bertindak sebagai wadah yang mengambil, menampilkan, dan mengintegrasikan mikro-frontend. Aplikasi shell menangani perutean global, yang memuat frontend mikro yang berbeda. Plugin @angular -architects/module-federation mengintegrasikan Federasi Modul dengan Angular. Anda menerapkan aplikasi shell dan micro-frontend dengan menggunakan. AWS Amplify Pengguna akhir mengakses aplikasi melalui portal berbasis web.

Portal dibagi secara vertikal. Ini berarti bahwa frontend mikro adalah keseluruhan tampilan atau kelompok tampilan, bukan bagian dari tampilan yang sama. Oleh karena itu aplikasi shell hanya memuat satu micro-frontend pada satu waktu.

Micro-frontend diimplementasikan sebagai modul jarak jauh. Aplikasi shell dengan malas memuat modul jarak jauh ini, yang menunda inisialisasi mikro-frontend hingga diperlukan. Pendekatan ini mengoptimalkan kinerja aplikasi dengan hanya memuat modul yang diperlukan. Ini mengurangi waktu muat awal dan meningkatkan pengalaman pengguna secara keseluruhan. Selain itu, Anda berbagi dependensi umum di seluruh modul melalui file konfigurasi webpack (webpack.config.js). Praktik ini mempromosikan penggunaan kembali kode, mengurangi duplikasi, dan merampingkan proses bundling.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS

- Node.js dan npm, diinstal
- Amplify CLI, diinstal
- CLI sudut, dipasang
- Izin untuk digunakan AWS Amplify
- Keakraban dengan Angular

Versi produk

- Angular CLI versi 13.1.2 atau yang lebih baru
- @angular -architects/module-federation versi 14.0.1 atau yang lebih baru
- webpack versi 5.4.0 atau yang lebih baru
- AWS Amplify Gen 1

Batasan

Arsitektur micro-frontend adalah pendekatan yang ampuh untuk membangun aplikasi web yang terukur dan tangguh. Namun, penting untuk memahami tantangan potensial berikut sebelum mengadopsi pendekatan ini:

- Integrasi — Salah satu tantangan utama adalah potensi peningkatan kompleksitas dibandingkan dengan frontend monolitik. Mengatur beberapa frontend mikro, menangani komunikasi di antara mereka, dan mengelola dependensi bersama bisa lebih rumit. Selain itu, mungkin ada overhead kinerja yang terkait dengan komunikasi antara frontend mikro. Komunikasi ini dapat meningkatkan latensi dan mengurangi kinerja. Ini perlu ditangani melalui mekanisme pesan yang efisien dan strategi berbagi data.
- Duplikasi kode — Karena setiap micro-frontend dikembangkan secara independen, ada risiko duplikasi kode untuk fungsionalitas umum atau pustaka bersama. Ini dapat meningkatkan ukuran aplikasi secara keseluruhan dan memperkenalkan tantangan pemeliharaan.
- Koordinasi dan manajemen - Mengkoordinasikan proses pengembangan dan penyebaran di beberapa frontend mikro dapat menjadi tantangan. Memastikan pembuatan versi yang konsisten, mengelola dependensi, dan menjaga kompatibilitas antar komponen menjadi lebih penting dalam arsitektur terdistribusi. Menetapkan tata kelola yang jelas, pedoman, dan jalur pipa pengujian dan penyebaran otomatis sangat penting untuk kolaborasi dan pengiriman yang mulus.
- Pengujian - Menguji arsitektur mikro-frontend bisa lebih kompleks daripada menguji frontend monolitik. Ini membutuhkan upaya tambahan dan strategi pengujian khusus untuk melakukan

pengujian dan pengujian integrasi lintas komponen, dan end-to-end untuk memvalidasi pengalaman pengguna yang konsisten di beberapa frontend mikro.

Sebelum berkomitmen pada pendekatan micro-frontend, kami sarankan Anda meninjau Memahami dan menerapkan micro-frontend pada. AWS

Arsitektur

Dalam arsitektur micro-frontend, setiap tim mengembangkan dan menyebarkan fitur secara independen. Gambar berikut menunjukkan bagaimana beberapa DevOps tim bekerja sama. Tim portal mengembangkan aplikasi shell. Aplikasi shell bertindak sebagai wadah. Ini mengambil, menampilkan, dan mengintegrasikan aplikasi mikro-frontend yang diterbitkan oleh tim lain. DevOps Anda gunakan AWS Amplify untuk mempublikasikan aplikasi shell dan aplikasi micro-frontend.

Diagram arsitektur menunjukkan alur kerja berikut:

1. Tim portal mengembangkan dan memelihara aplikasi shell. Aplikasi shell mengatur integrasi dan rendering mikro-frontend untuk menyusun portal secara keseluruhan.
2. Tim A dan B mengembangkan dan memelihara satu atau lebih mikro-frontend atau fitur yang terintegrasi ke dalam portal. Setiap tim dapat bekerja secara independen pada frontend mikro masing-masing.
3. Pengguna akhir mengautentikasi dengan menggunakan Amazon Cognito.
4. Pengguna akhir mengakses portal, dan aplikasi shell dimuat. Saat pengguna menavigasi, aplikasi shell menangani perutean dan mengambil micro-frontend yang diminta, memuat bundelnya.

Alat

Layanan AWS

- AWS Amplifyadalah seperangkat alat dan fitur yang dibuat khusus yang membantu pengembang web dan seluler frontend dengan cepat membangun aplikasi full-stack. AWS Dalam pola ini, Anda menggunakan Amplify CLI untuk menerapkan aplikasi micro-frontend Amplify.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

Alat lainnya

- @angular -architects/module-federation adalah plugin yang mengintegrasikan Angular dengan Federasi Modul.
- Angular adalah kerangka kerja aplikasi web open source untuk membangun aplikasi satu halaman yang modern, terukur, dan dapat diuji. Ini mengikuti arsitektur modular dan berbasis komponen yang mempromosikan penggunaan kembali dan pemeliharaan kode.
- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.
- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.
- Federasi Modul Webpack membantu Anda memuat kode yang dikompilasi dan digunakan secara independen, seperti mikro-frontend atau plugin, ke dalam aplikasi.

Repositori kode

Kode untuk pola ini tersedia di portal Micro-frontend menggunakan repositori Angular dan Module Federation. GitHub Repositori ini berisi dua folder berikut:

- `shell-app`berisi kode untuk aplikasi shell.
- `feature1-app`berisi sampel micro-frontend. Aplikasi shell mengambil micro-frontend ini dan menampilkannya sebagai halaman dalam aplikasi portal.

Praktik terbaik

Arsitektur mikro-frontend menawarkan banyak keuntungan, tetapi mereka juga memperkenalkan kompleksitas. Berikut ini adalah beberapa praktik terbaik untuk pengembangan yang lancar, kode berkualitas tinggi, dan pengalaman pengguna yang hebat:

- Perencanaan dan komunikasi — Untuk merampingkan kolaborasi, berinvestasi dalam perencanaan awal, desain, dan saluran komunikasi yang jelas.
- Konsistensi desain - Menerapkan gaya visual yang konsisten di seluruh frontend mikro dengan menggunakan sistem desain, panduan gaya, dan pustaka komponen. Ini memberikan pengalaman pengguna yang kohesif dan mempercepat pengembangan.
- Manajemen ketergantungan — Karena frontend mikro berkembang secara independen, mengadopsi kontrak standar dan strategi pembuatan versi untuk mengelola dependensi secara efektif dan mencegah masalah kompatibilitas.

- Arsitektur micro-frontend — Untuk memungkinkan pengembangan dan penyebaran independen, setiap frontend mikro harus memiliki tanggung jawab yang jelas dan terdefinisi dengan baik untuk fungsionalitas yang dienkapsulasi.
- Integrasi dan komunikasi — Untuk memfasilitasi kelancaran integrasi dan meminimalkan konflik, tentukan kontrak yang jelas dan protokol komunikasi antara frontend mikro, termasuk, peristiwa APIs, dan model data bersama.
- Pengujian dan jaminan kualitas - Menerapkan otomatisasi pengujian dan jaringan pipa integrasi berkelanjutan untuk frontend mikro. Ini meningkatkan kualitas keseluruhan, mengurangi upaya pengujian manual, dan memvalidasi fungsionalitas antara interaksi mikro-frontend.
- Optimalisasi kinerja — Terus memantau metrik kinerja dan melacak dependensi antara frontend mikro. Ini membantu Anda mengidentifikasi kemacetan dan mempertahankan kinerja aplikasi yang optimal. Gunakan alat pemantauan kinerja dan analisis ketergantungan untuk tujuan ini.
- Pengalaman pengembang - Fokus pada pengalaman pengembang dengan memberikan dokumentasi, perkakas, dan contoh yang jelas. Ini membantu Anda merampingkan pengembangan dan memasukkan anggota tim baru.

Epik

Buat aplikasi shell

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat aplikasi shell.	1. Di CLI Angular, masukkan perintah berikut:
	`ng new shell --routing`
	2. Masukkan perintah berikut untuk menavigasi ke folder proyek:
	`cd shell`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Folder dan struktur proyek untuk aplikasi shell dan micro-frontend dapat sepenuhnya independen. Mereka dapat ditangani sebagai aplikasi Angular independen.	
Instal plugin.	Di CLI Angular, masukkan perintah berikut untuk menginstal plugin @angular-architects/module-federation: `ng add @angular-architects/module-federation --project shell --port 4200`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan URL micro-frontend sebagai variabel lingkungan.	1. Buka file environment.ts. 2. Tambahkan `mfe1URL: 'http://localhost:5000'` ke `environment` objek: `export const environment = { production: false, mfe1URL: 'http://localhost:5000', };` 3. Simpan dan tutup file environment.ts.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tentukan routing.	1. Buka file app-routing.module.ts.
	2. Di CLI Angular, masukkan perintah berikut untuk mengimpor `loadRemoteModule` modul dari plugin @angular -architects/module-federation:
	`import { loadRemoteModule } from '@angular-architects/module-federation';`
	3. Tetapkan rute default sebagai berikut:
	`{ path: '', pathMatch: 'full', redirectTo: 'mfe1' },`
	4. Mengatur rute untuk micro-frontend:
	`{ path: 'mfe1', loadChildren: () => loadRemoteModule({ type: 'module', remoteEntry: `${environment.mfe`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>1URL}/remoteEntry.
	js`,
	exposedMo
	dule: './Module'
	})
	.then(m =>
	m.Mfe1Module)
	},</pre>
	5. Simpan dan tutup file app-routing.module.ts.	

Deklarasikan modul. `mfe1`	1. Di `src` folder, buat file baru bernama decl.d.ts.
	2. Buka file decl.d.ts.
	3. Tambahkan yang berikut ini ke file:
	<pre>declare module 'mfe1/
	Module';</pre>
	4. Simpan dan tutup file decl.d.ts.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan preloading untuk micro-frontend.	Memuat pramuat micro-fro ntend membantu webpack menegosiasikan pustaka dan paket bersama dengan benar.
	1. Buka file main.ts.
	2. Ganti konten dengan yang berikut ini:
	`import { loadRemot eEntry } from '@angular-architec ts/module-federati on';`
	`Promise.all([ loadRemoteEntry(` ${environment.mfe1U RL}/remoteEntry.js `, 'mfe1'), ]) .catch(err => console.error('Err or loading remote entries', err)) .then(() => import('. /bootstrap')) .catch(err => console.error(err) );`
	3. Simpan dan tutup file main.ts.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan konten HTML.	1. Buka file app.compo nent.html. 2. Ganti konten dengan yang berikut ini: `<h1>Shell application is running!</h1> <router-outlet></ router-outlet>` 3. Simpan dan tutup file app.component.html.	Pengembang aplikasi

Buat aplikasi micro-frontend

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat micro-frontend.	1. Di CLI Angular, masukkan perintah berikut:
	`ng new mfe1 --routing`
	2. Masukkan perintah berikut untuk menavigasi ke folder proyek:
	`cd mfe1`
		Pengembang aplikasi

Instal plugin.	Masukkan perintah berikut untuk menginstal plugin @angular -architects/module-federation: `ng add @angular-architects/module-`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`federation --project mfe1 --port 5000`	

Buat modul dan komponen.	Masukkan perintah berikut untuk membuat modul dan komponen dan mengekspornya sebagai modul entri jarak jauh:
	`ng g module mfe1 --routing`
	`ng g c mfe1`
		Pengembang aplikasi

Mengatur jalur routing default.	1. Buka file mfe-routing.module.ts. 2. Tetapkan rute default sebagai berikut: `{ path: '', component: Mfe1Component },` 3. Simpan dan tutup file mfe-routing.module.ts.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan `mfe1` rute.	1. Buka file app-routing.module.ts.
	2. Tetapkan rute default sebagai berikut:
	`{ path: '', pathMatch: 'full', redirectTo: 'mfe1' },`
	3. Tambahkan `mfe1` rute berikut:
	`{ path: 'mfe1', loadChildren: () => import('./mfe1/mfe1.module').then((m) => m.Mfe1Module), },`
	4. Simpan dan tutup file app-routing.module.ts.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Edit file webpack.config.js.	1. Buka file webpack.config.js.
	2. Edit `For remotes` bagian agar sesuai dengan yang berikut:
	`// For remotes (please adjust)`
	`name: "mfe1",`
	`filename: "remoteEntry.js",`
	`exposes: {`
	`'./Module': './src/app/mfe1/mfe1.module.ts',`
	`},`
	3. Di `shared` bagian ini, tambahkan dependensi apa pun yang dibagikan `mfe1` aplikasi dengan aplikasi shell:
	`shared: share({`
	`"@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },`
	`"@angular/common": { singleton: true, strictVersion: true, requiredVersion: 'auto' },`
	`"@angular/common/http": { singleton: true, strictVersion: true, requiredVersion: 'auto' },`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"@angular/router": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
	
	...sharedMappings.getDescriptors()
	})</pre>
	4. Simpan dan tutup file webpack.config.js.	

Sesuaikan konten HTML.	1. Buka file app.component.html. 2. Ganti konten dengan yang berikut ini: <pre><router-outlet></router-outlet></pre> 3. Simpan dan tutup file app.component.html.	Pengembang aplikasi

Jalankan aplikasi secara lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan `mfe1` aplikasi.	1. Masukkan perintah berikut untuk memulai `mfe1` aplikasi: <pre>npm start</pre>	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di browser web, akses`http://localhost:5000`. 3. Verifikasi bahwa micro-frontend dapat dijalankan secara mandiri. `mfe1`Aplikasi harus dirender dengan benar tanpa kesalahan.	
Jalankan aplikasi shell.	1. Masukkan perintah berikut untuk memulai aplikasi shell: `npm start` 2. Di browser web, akses`http://localhost:4200/mfe1`. 3. Verifikasi bahwa `mfe1` micro-frontend tertanam dalam aplikasi shell. Aplikasi portal harus dirender dengan benar tanpa kesalahan, dan `mfe1` aplikasi harus disematkan di dalamnya.	Pengembang aplikasi

Memfaktorkan ulang aplikasi shell untuk menangani kesalahan pemuatan frontend mikro

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat modul dan komponen.	Di folder root aplikasi shell, masukkan perintah berikut untuk membuat modul dan	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	komponen untuk halaman kesalahan:
	`ng g module error-page --routing`
	`ng g c error-page`	

Sesuaikan konten HTML.	1. Buka file error-page.component.html. 2. Ganti konten dengan yang berikut ini: `<p>Sorry, this page is not available.</p>` 3. Simpan dan tutup file error-page.component.html.	Pengembang aplikasi
Mengatur jalur routing default.	1. Buka error-page-routing file.module.ts. 2. Tetapkan rute default sebagai berikut: `{ path: '', component: ErrorPageComponent },` 3. Simpan dan tutup error-page-routingfile.module.ts.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat fungsi untuk memuat mikro-frontend.	1. Buka file app-routing.module.ts.
	2. Buat fungsi berikut:
	`function loadMFE(url: string) { return loadRemoteModule({ type: 'module', remoteEntry: `${url}/remoteEntry.js`, exposedModule: './Module' }) .then(m => m.Mfe1Module) .catch( () => import('./error-page/error-page.module').then(m => m.ErrorPageModule) ); }`
	3. Ubah `mfe1` rute menjadi sebagai berikut:
	`{ path: 'mfe1', loadChildren: () => loadMFE(environment.mfe1URL) },`
	4. Simpan dan tutup file app-routing.module.ts.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji penanganan kesalahan.	1. Jika belum berjalan, masukkan perintah berikut untuk memulai aplikasi shell:
	`npm start`
	2. Di browser web, akses`http://localhost:4200/mfe1` .
	3. Verifikasi bahwa halaman kesalahan dirender. Anda akan melihat teks berikut:
	`Sorry, this page is not available.`
		Pengembang aplikasi

Menyebarkan aplikasi dengan menggunakan AWS Amplify

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan micro-frontend.	1. Di Amplify CLI, navigasikan ke folder root aplikasi micro-frontend. 2. Masukkan perintah berikut untuk menginisialisasi Amplify: `amplify init` 3. Ketika Anda diminta untuk memasukkan nama untuk proyek Amplify Anda, tekan Enter. Ini menggunakan	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kembali nama dari file package.json. 4. Ketika Anda diminta untuk menginisialisasi proyek dengan konfigurasi di atas, masukkan. `Yes` 5. Ketika Anda diminta untuk memilih metode otentikasi, pilih. `AWS Profile` 6. Pilih profil yang ingin Anda gunakan. 7. Tunggu Amplify untuk menginisialisasi proyek. Ketika proses ini selesai, Anda menerima pesan konfirmasi di terminal. 8. Masukkan perintah berikut untuk menambahkan kategori Amplify hosting ke micro-frontend: `amplify add hosting` 9. Ketika Anda diminta untuk memilih modul plugin, pilih`Hosting with Amplify Console`. 10. Ketika Anda diminta untuk memilih jenis, pilih`Manual deployment` . 11. Instal dependensi proyek npm dengan memasukkan perintah berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`npm install`
	12. Publikasikan aplikasi ke konsol Amplify dengan memasukkan perintah berikut:
	`amplify publish -y`
	Saat penerbitan selesai, Amplify mengembalikan URL micro-frontend.
	13. Salin URL. Anda memerlukan nilai ini untuk memperbarui aplikasi shell.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan aplikasi shell.	1. Di src/app/environmentsfolder, buka file environments.prod.ts.
	2. Ganti `mfe1URL` nilainya dengan URL micro-frontend yang digunakan:
	`export const environment = { production: true, mfe1URL: 'https://<env>.<Amplify-app-ID>.amplifyapp.com' };`
	3. Simpan dan tutup file environments.prod.ts.
	4. Di Amplify CLI, arahkan ke folder root aplikasi shell.
	5. Masukkan perintah berikut untuk menginisialisasi Amplify:
	`amplify init`
	6. Ketika Anda diminta untuk memasukkan nama untuk proyek Amplify Anda, tekan Enter. Ini menggunakan kembali nama dari file package.json.
	7. Ketika Anda diminta untuk menginisialisasi proyek dengan konfigurasi di atas, masukkan. `Yes`
		Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	8. Ketika Anda diminta untuk memilih metode otentikasi, pilih. `AWS Profile`
	9. Pilih profil yang ingin Anda gunakan.
	10. Tunggu Amplify untuk menginisialisasi proyek. Ketika proses ini selesai, Anda menerima pesan konfirmasi di terminal.
	11. Tambahkan kategori hosting Amplify ke aplikasi shell:
	`amplify add hosting`
	12. Ketika Anda diminta untuk memilih modul plugin, pilih`Hosting with Amplify Console`.
	13. Ketika Anda diminta untuk memilih jenis, pilih`Manual deployment` .
	14. Instal dependensi proyek npm dengan memasukkan perintah berikut:
	`npm install`
	15. Publikasikan aplikasi shell ke konsol Amplify dengan memasukkan perintah berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`amplify publish -y` Saat penerbitan selesai, Amplify mengembalikan URL aplikasi shell yang digunakan. 16. Catat URL untuk aplikasi shell.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan CORS.	Karena aplikasi shell dan micro-frontend di-host secara independen pada domain yang berbeda, Anda harus mengaktifkan cross-origin resource sharing (CORS) di micro-frontend. Ini memungkinkan aplikasi shell memuat konten dari asal yang berbeda. Untuk mengaktifkan CORS, Anda menambahkan header khusus. 1. Di Amplify CLI, navigasikan ke folder root dari micro-frontend. 2. Masukkan perintah berikut: `amplify configure hosting` 3. Saat Anda diminta untuk mengonfigurasi pengaturan khusus, masukkanY. 4. Masuk ke Konsol Manajemen AWS, lalu buka konsol Amplify. 5. Pilih micro-frontend. 6. Di panel navigasi, pilih Hosting, lalu pilih Header khusus. 7. Pilih Edit.	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	8. Di jendela Edit header khusus, masukkan yang berikut ini:
	
	`customHeaders:`
	`  - pattern: '*.js'`
	`  headers:`
	`  - key: Access-Control-Allow-Origin`
	`  value: '*'`
	`  - key: Access-Control-Allow-Methods`
	`  value: 'GET, OPTIONS'`
	`  - key: Access-Control-Allow-Headers`
	`  value: '*'`
	
	9. Pilih Simpan.
	10. Menerapkan ulang micro-frontend untuk menerapkan header kustom baru.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan penulisan ulang pada aplikasi shell.	Aplikasi shell Angular dikonfigurasi untuk menggunakan HTML5 routing. Jika pengguna melakukan penyegaran keras, Amplify mencoba memuat halaman dari URL saat ini. Ini menghasilkan kesalahan 403. Untuk menghindari hal ini, Anda menambahkan aturan penulisan ulang di konsol Amplify. Untuk membuat aturan penulisan ulang, ikuti langkah-langkah ini: 1. Di Amplify CLI, arahkan ke folder root aplikasi shell. 2. Masukkan perintah berikut: `amplify configure hosting` 3. Saat Anda diminta untuk mengonfigurasi pengaturan khusus, masukkanY. 4. Buka konsol Amplify. 5. Pilih aplikasi shell. 6. Di panel navigasi, pilih Hosting, lalu pilih Menulis ulang dan mengarahkan ulang. 7. Pada halaman Menulis ulang dan mengarahkan	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ulang, pilih Kelola pengalihan. 8. Pilih Buka editor teks. 9. Di editor JSON, masukkan pengalihan berikut: `[ { "source": "/<*>", "target": "/index.html", "status": "404-200", "condition": null } ]` 10. Pilih Simpan.	
Uji portal web.	1. Di browser web, masukkan URL aplikasi shell yang digunakan. 2. Verifikasi bahwa aplikasi shell dan micro-frontend dimuat dengan benar.	Pengembang aplikasi

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus aplikasi.	Jika Anda tidak lagi membutuhkan aplikasi shell dan micro-frontend, hapuslah. Ini membantu mencegah biaya	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk sumber daya yang tidak Anda gunakan. 1. Masuk ke Konsol Manajemen AWS, lalu buka konsol Amplify. 2. Pilih micro-frontend. 3. Di panel navigasi, pilih Pengaturan aplikasi, lalu pilih Pengaturan umum. 4. Pilih Hapus aplikasi. 5. Di jendela konfirmasi, masukkan`delete`, lalu pilih Hapus aplikasi. 6. Ulangi langkah-langkah ini untuk menghapus aplikasi shell.	

Pemecahan masalah

Isu	Solusi
Tidak ada AWS profil yang tersedia saat menjalankan `amplify init` perintah	Jika Anda tidak memiliki AWS profil yang dikonfigurasi, Anda masih dapat melanjutk an dengan `amplify init` perintah. Namun, Anda harus memilih `AWS access keys` opsi ketika Anda diminta untuk metode otentikasi. Sediakan kunci AWS akses dan kunci rahasia Anda. Atau, Anda dapat mengonfigurasi profil bernama untuk file AWS CLI. Untuk petunjuk,

Isu	Solusi
	lihat Pengaturan konfigurasi dan file kredensi dalam AWS CLI dokumentasi.
Kesalahan saat memuat entri jarak jauh	Jika Anda mengalami kesalahan saat memuat entri jarak jauh di file main.ts dari aplikasi shell, pastikan `environment.mfe1URL` variabel disetel dengan benar. Nilai variabel ini harus URL dari micro-frontend.
Kesalahan 404 saat mengakses micro-frontend	Jika Anda mendapatkan kesalahan 404 saat mencoba mengakses micro-frontend lokal, seperti at`http://localhost:4200/mfe1` , periksa hal berikut: • Untuk aplikasi shell, pastikan bahwa konfigurasi routing dalam file app-routi ng.module.ts diatur dengan benar, dan pastikan bahwa fungsi tersebut memanggil micro-frontend dengan benar. `loadRemot eModule` • Untuk micro-frontend, verifikasi bahwa file webpack.config.js memiliki **`exposes`** konfigurasi yang benar, dan pastikan bahwa file remoteEntry.js sedang dibuat dengan benar.

Informasi tambahan

AWS dokumentasi

- Memahami dan menerapkan frontend mikro pada AWS(Panduan Preskriptif)AWS
- Amplify CLI (Amplify dokumentasi)
- Amplify Hosting (Amplify dokumentasi)

Referensi lainnya

- Federasi Modul
- Node.js
- Sudut
- @angular -architects/modul-federasi

Menerapkan aplikasi satu halaman berbasis React ke Amazon S3 dan CloudFront

Jean-Baptiste Guillois, Amazon Web Services

Ringkasan

Aplikasi satu halaman (SPA) adalah situs web atau aplikasi web yang secara dinamis memperbarui konten halaman web yang ditampilkan dengan menggunakan. JavaScript APIs Pendekatan ini meningkatkan pengalaman pengguna dan kinerja situs web karena hanya memperbarui data baru alih-alih memuat ulang seluruh halaman web dari server.

Pola ini memberikan step-by-step pendekatan untuk mengkodekan dan menghosting SPA yang ditulis dalam React di Amazon Simple Storage Service (Amazon S3) dan Amazon. CloudFront SPA dalam pola ini menggunakan REST API yang dikonfigurasi di Amazon API Gateway dan diekspos melalui CloudFront distribusi Amazon untuk menyederhanakan manajemen berbagi sumber daya lintas asal (CORS).

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Node.js dan`npm`, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat bagian Unduhan pada dokumentasi Node.js.
- Benang, dipasang dan dikonfigurasi. Untuk informasi selengkapnya, lihat dokumentasi Yarn.
- Git, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat dokumentasi Git.

Arsitektur

Arsitektur ini secara otomatis digunakan dengan menggunakan AWS CloudFormation (infrastruktur sebagai kode). Ini menggunakan layanan Regional seperti Amazon S3 untuk menyimpan aset statis dan Amazon CloudFront dengan Amazon API Gateway untuk mengekspos titik akhir API Regional (REST). Log aplikasi dikumpulkan dengan menggunakan Amazon CloudWatch. Semua panggilan AWS API diaudit di AWS CloudTrail. Semua konfigurasi keamanan (misalnya, identitas dan izin) dikelola di AWS Identity and Access Management (IAM). Konten statis dikirimkan melalui jaringan pengiriman CloudFront konten Amazon (CDN), dan kueri DNS ditangani oleh Amazon Route 53.

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- Amazon CloudFront mempercepat distribusi konten web Anda dengan mengirimkannya melalui jaringan pusat data di seluruh dunia, yang menurunkan latensi dan meningkatkan kinerja.
- AWS CloudTrailmembantu Anda mengaudit tata kelola, kepatuhan, dan risiko operasional Anda Akun AWS.
- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Kode

Contoh kode aplikasi pola ini tersedia di repositori aplikasi satu halaman CORS GitHub berbasis React.

Praktik terbaik

Dengan menggunakan penyimpanan objek Amazon S3, Anda dapat menyimpan aset statis aplikasi Anda dengan cara yang aman, sangat tangguh, berkinerja, dan hemat biaya. Tidak perlu menggunakan wadah khusus atau instans Amazon Elastic Compute Cloud (Amazon EC2) untuk tugas ini.

Dengan menggunakan jaringan pengiriman CloudFront konten Amazon, Anda dapat mengurangi latensi yang mungkin dialami pengguna saat mereka mengakses aplikasi Anda. Anda juga dapat melampirkan firewall aplikasi web (AWS WAF) untuk melindungi aset Anda dari serangan berbahaya.

Epik

Membangun dan menerapkan aplikasi Anda secara lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Jalankan perintah berikut untuk mengkloning repositori aplikasi sampel: `git clone https://github.com/aws-samples/react-cors-spa react-cors-spa && cd react-cors-spa`	Pengembang aplikasi, AWS DevOps
Menyebarkan aplikasi secara lokal.	1. Dalam direktori proyek, jalankan `npm install` perintah untuk memulai dependensi aplikasi. 2. Jalankan `yarn dev` perintah untuk memulai aplikasi secara lokal.	Pengembang aplikasi, AWS DevOps
Akses aplikasi secara lokal.	Buka jendela browser dan masukkan `http://localhost:3000` URL untuk mengakses aplikasi.	Pengembang aplikasi, AWS DevOps

Deploy aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan AWS CloudFormation template.	1. Masuk ke Konsol Manajemen AWS, lalu buka AWS CloudFormation konsol.	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pilih Buat Tumpukan, lalu pilih Dengan sumber daya baru (standar). 3. Pilih Mengunggah file template. 4. Pilih Pilih file, pilih **`react-cors-spa-stack.yaml`** file dari repositori kloning, lalu pilih Berikutnya. 5. Masukkan nama untuk tumpukan Anda, lalu pilih Berikutnya. 6. Simpan semua opsi default, lalu pilih Berikutnya. 7. Tinjau pengaturan akhir untuk tumpukan Anda, lalu pilih Buat tumpukan.	
Sesuaikan file sumber aplikasi Anda.	1. Setelah tumpukan Anda dikerahkan, buka tab Output dan identifikasi `Bucket` nama dan `APIDomain` nilainya. 2. Salin domain CloudFront distribusi untuk REST API. 3. Arahkan ke`<project_root>/src/pages/index.tsx`, lalu masukkan atau tempel domain ini ke nilai `APIEndPoint` variabel pada baris 13 `index.tsx` file.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bangun paket aplikasi.	Di direktori proyek Anda, jalankan `yarn build` perintah untuk membangun paket aplikasi.	Pengembang aplikasi
Menyebarkan paket aplikasi.	1. Buka konsol Amazon S3. 2. Identifikasi dan pilih bucket S3 yang dibuat sebelumnya oleh CloudFormation tumpukan. 3. Pilih Unggah, lalu pilih Tambahkan file. 4. Pilih konten `out` folder Anda. 5. Pilih Tambah folder, lalu pilih `_next` direktori. **Important** Pilih `_next` direktori, bukan isinya. 6. Pilih Unggah untuk mengunggah file dan direktori ke bucket S3 Anda.	Pengembang aplikasi, AWS DevOps

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses dan uji aplikasi.	Buka jendela browser, lalu tempel domain CloudFront distribusi (`SPADomain` output dari CloudForm	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ation tumpukan yang Anda gunakan sebelumnya) untuk mengakses aplikasi.	

Bersihkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus isi bucket S3.	1. Buka konsol Amazon S3 dan pilih bucket yang dibuat sebelumnya oleh tumpukan (bucket pertama yang namanya dimulai dengan`react-cors-spa-`). 2. Pilih Kosong untuk menghapus isi bucket. 3. Pilih ember kedua yang dibuat sebelumnya oleh tumpukan (ember kedua yang namanya dimulai dengan `react-cors-spa-` dan diakhiri dengan`-logs`). 4. Pilih Kosong untuk menghapus isi bucket.	AWS DevOps, Pengembang aplikasi
Hapus CloudFormation tumpukan.	1. Buka CloudFormation konsol dan pilih tumpukan yang Anda buat sebelumnya.	AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pilih Hapus untuk menghapus tumpukan dan semua sumber daya terkait.	

Sumber daya terkait

Untuk menyebarkan dan meng-host aplikasi web Anda, Anda juga dapat menggunakan AWS Amplify Hosting, yang menyediakan alur kerja berbasis Git untuk hosting aplikasi web full-stack, tanpa server dengan penerapan berkelanjutan. Amplify Hosting adalah bagian dari AWS Amplify, yang menyediakan seperangkat alat dan fitur yang dibuat khusus yang memungkinkan pengembang web dan seluler frontend untuk membangun aplikasi full-stack dengan cepat dan mudah. AWS

Informasi tambahan

Untuk menangani URLs permintaan yang tidak valid oleh pengguna yang mungkin menghasilkan 403 kesalahan, halaman kesalahan kustom yang dikonfigurasi dalam CloudFront distribusi menangkap 403 kesalahan dan mengarahkannya ke titik masuk aplikasi (). `index.html`

Untuk menyederhanakan pengelolaan CORS, REST API diekspos melalui distribusi. CloudFront

Menerapkan API Amazon API Gateway di situs web internal menggunakan endpoint pribadi dan Application Load Balancer

Saurabh Kothari, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menerapkan API Amazon API Gateway di situs web internal yang dapat diakses dari jaringan lokal. Anda belajar membuat nama domain khusus untuk API pribadi dengan menggunakan arsitektur yang dirancang dengan titik akhir pribadi, Application Load Balancer, PrivateLink AWS, dan Amazon Route 53. Arsitektur ini mencegah konsekuensi yang tidak diinginkan dari penggunaan nama domain khusus dan server proxy untuk membantu perutean berbasis domain pada API. Misalnya, jika Anda menerapkan titik akhir virtual private cloud (VPC) di subnet yang tidak dapat dirutekan, jaringan Anda tidak dapat menjangkau API Gateway. Solusi umum adalah menggunakan nama domain khusus dan kemudian menerapkan API di subnet yang dapat dirutekan, tetapi ini dapat merusak situs internal lainnya ketika konfigurasi proxy meneruskan traffic (`execute-api.{region}.vpce.amazonaws.com`) ke AWS Direct Connect. Terakhir, pola ini dapat membantu Anda memenuhi persyaratan organisasi untuk menggunakan API pribadi yang tidak dapat dijangkau dari internet dan nama domain khusus.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sertifikat Indikasi Nama Server (SNI) untuk situs web dan API Anda
- Sambungan dari lingkungan lokal ke akun AWS yang disiapkan menggunakan AWS Direct Connect atau AWS VPN Site-to-Site
- Zona host pribadi dengan domain yang sesuai (misalnya, domain.com) yang diselesaikan dari jaringan lokal dan meneruskan kueri DNS ke Route 53
- Subnet pribadi yang dapat dirutekan yang dapat dijangkau dari jaringan lokal

Batasan

Untuk informasi selengkapnya tentang kuota (sebelumnya disebut sebagai limit) untuk penyeimbang beban, aturan, dan sumber daya lainnya, lihat Kuota untuk Application Load Balancers Anda dalam dokumentasi Elastic Load Balancing.

Arsitektur

Tumpukan teknologi

- Amazon API Gateway
- Amazon Route 53
- Penyeimbang Beban Aplikasi
- AWS Certificate Manager
- AWS PrivateLink

Arsitektur target

Diagram berikut menunjukkan bagaimana Application Load Balancer digunakan dalam VPC yang mengarahkan lalu lintas web ke grup target situs web atau grup target API Gateway berdasarkan aturan pendengar Application Load Balancer. Grup target API Gateway adalah daftar alamat IP untuk titik akhir VPC di API Gateway. API Gateway dikonfigurasi untuk membuat API pribadi dengan kebijakan sumber dayanya. Kebijakan ini menyangkal semua panggilan yang bukan berasal dari titik akhir VPC tertentu. Nama domain khusus di gateway API diperbarui untuk menggunakan api.domain.com untuk API dan tahapannya. Aturan Application Load Balancer ditambahkan ke lalu lintas rute berdasarkan nama host.

Diagram menunjukkan alur kerja berikut:

1. Pengguna dari jaringan lokal mencoba mengakses situs web internal. Permintaan dikirim ke ui.domain.com dan api.domain.com. Kemudian, permintaan diselesaikan ke internal Application Load Balancer dari subnet privat routable. SSL dihentikan di Application Load Balancer untuk ui.domain.com dan api.domain.com.
2. Aturan listener, yang dikonfigurasi pada Application Load Balancer, periksa header host.

   a. Jika header host adalah api.domain.com, permintaan diteruskan ke grup target API Gateway. Application Load Balancer memulai koneksi baru ke API Gateway melalui port 443.

   b. Jika header host adalah ui.domain.com, permintaan diteruskan ke grup target situs web.
3. Saat permintaan mencapai API Gateway, pemetaan domain kustom yang dikonfigurasi di API Gateway menentukan nama host dan API mana yang akan dijalankan.

Otomatisasi dan skala

Langkah-langkah dalam pola ini dapat diotomatisasi dengan menggunakan AWS CloudFormation atau AWS Cloud Development Kit (AWS CDK). Untuk mengonfigurasi grup target panggilan API Gateway, Anda harus menggunakan sumber daya khusus untuk mengambil alamat IP titik akhir VPC. API memanggil describe-vpc-endpointsdan describe-network-interfacesmengembalikan alamat IP dan grup keamanan, yang dapat digunakan untuk membuat grup target API dari alamat IP.

Alat

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- Amazon Route 53 adalah layanan web DNS yang sangat tersedia dan dapat diskalakan.
- AWS Certificate Manager (ACM) membantu Anda membuat, menyimpan, dan memperbarui sertifikat dan kunci SSL/TLS X.509 publik dan pribadi yang melindungi situs web dan aplikasi AWS Anda.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS PrivateLink membantu Anda membuat koneksi pribadi searah dari layanan Anda VPCs ke layanan di luar VPC.

Epik

Buat sertifikat SNI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sertifikat SNI dan impor sertifikat ke ACM.	1. Buat sertifikat SNI untuk ui.domain.com dan api.domain.com. Untuk informasi selengkap nya, lihat Memilih cara CloudFront menyajikan permintaan HTTPS dalam CloudFront dokumentasi Amazon.	Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Impor sertifikat SNI ke AWS Certificate Manager (ACM). Untuk informasi selengkapnya, lihat Mengimpor sertifikat ke AWS Certificate Manager di dokumentasi ACM.	

Menerapkan titik akhir VPC di subnet pribadi yang tidak dapat dirutekan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir VPC antarmuka di API Gateway.	Untuk membuat titik akhir VPC antarmuka, ikuti petunjuk dari Akses layanan AWS menggunakan titik akhir VPC antarmuka dalam dokumentasi Amazon Virtual Private Cloud (Amazon VPC).	Administrator awan

Konfigurasikan Application Load Balancer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup target untuk aplikasi Anda.	Buat grup target untuk sumber daya UI aplikasi Anda.	Administrator awan
Buat grup target untuk titik akhir API Gateway.	1. Buat grup target dengan tipe alamat IP, lalu tambahkan alamat IP titik akhir VPC untuk titik akhir API Gateway ke grup target.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Konfigurasikan pemeriksaan kesehatan untuk grup target Anda dengan kode sukses 403. 403 diperlukan karena titik akhir VPC untuk API Gateway mengembalikan kode 403 saat dipanggil tanpa header apa pun oleh pemeriksaan kesehatan grup target.	
Buat Application Load Balancer.	1. Buat Application Load Balancer (internal) di subnet pribadi yang dapat dirutekan. 2. Tambahkan 443 listener ke Application Load Balancer, dan pilih sertifikat dari ACM.	Administrator awan
Buat aturan pendengar.	Buat aturan listener untuk melakukan hal berikut: 1. Teruskan host api.domain.com ke grup target API Gateway 2. Teruskan host ui.domain.com ke grup target untuk sumber daya UI	Administrator awan

Konfigurasikan Route 53

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat zona host pribadi.	Buat zona host pribadi untuk domain.com.	Administrator awan
Buat catatan domain.	Buat catatan CNAME untuk hal berikut: • API dengan nilai yang disetel ke nama DNS Application Load Balancer • UI dengan nilai yang disetel ke nama DNS Application Load Balancer	Administrator awan

Buat titik akhir API pribadi di API Gateway

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan titik akhir API pribadi.	1. Untuk membuat endpoint API pribadi, ikuti petunjuk dari Membuat API pribadi di Amazon API Gateway dalam dokumentasi API Gateway. 2. Konfigurasikan kebijakan sumber daya untuk mengizinkan panggilan hanya ke API dari titik akhir VPC. Untuk informasi selengkapnya, lihat Mengontrol akses ke API dengan kebijakan sumber	Pengembang aplikasi, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	daya API Gateway dalam dokumentasi API Gateway.	
Buat nama domain khusus.	1. Buat nama domain khusus untuk api.domain.com. Untuk informasi selengkapnya, lihat Menyiapkan nama domain khusus untuk REST APIs di dokumentasi API Gateway. 2. Pilih API dan panggung yang dibuat. Untuk informasi selengkapnya, lihat Bekerja dengan pemetaan API untuk REST APIs di dokumentasi API Gateway.	Administrator awan

Sumber daya terkait

- Amazon API Gateway
- Amazon Route 53
- Penyeimbang Beban Aplikasi
- AWS PrivateLink
- AWS Certificate Manager

Sematkan komponen visual Amazon Quick Sight ke dalam aplikasi web dengan menggunakan otomatisasi Amazon Cognito dan IAc

Ishita Gupta, Saurabh Singh, dan Srishti Wadhwa, Amazon Web Services

Ringkasan

Pola ini memberikan pendekatan khusus untuk menyematkan komponen visual Amazon Quick Sight ke dalam aplikasi React dengan menggunakan penyematan pengguna terdaftar dengan otentikasi Amazon Cognito yang efisien. Sumber daya ini kemudian digunakan melalui template infrastruktur sebagai kode (IAc). Tidak seperti penyematan dasbor tradisional, solusi ini mengisolasi bagan dan grafik tertentu untuk integrasi langsung ke dalam aplikasi React, yang secara dramatis meningkatkan kinerja dan pengalaman pengguna.

Arsitektur menetapkan alur otentikasi yang efisien antara manajemen pengguna Amazon Cognito dan izin Quick Sight: Pengguna mengautentikasi melalui Amazon Cognito dan mengakses visualisasi resmi mereka berdasarkan aturan berbagi dasbor di Quick Sight. Pendekatan yang disederhanakan ini menghilangkan kebutuhan akan akses konsol Quick Sight langsung sambil mempertahankan kontrol keamanan yang kuat.

Lingkungan lengkap digunakan melalui satu AWS CloudFormation templat yang menyediakan semua komponen infrastruktur yang diperlukan, termasuk:

- Backend tanpa server yang menggunakan AWS Lambda dan Amazon API Gateway
- Hosting frontend aman melalui Amazon CloudFront, Amazon Simple Storage Service (Amazon S3), dan AWS WAF
- Manajemen identitas dengan menggunakan Amazon Cognito

Semua komponen dikonfigurasi dengan mengikuti praktik terbaik keamanan dengan kebijakan, perlindungan, dan enkripsi dengan hak istimewa paling rendah AWS Identity and Access Management (IAM). AWS WAF end-to-end

Solusi ini sangat ideal untuk tim pengembangan dan organisasi yang ingin mengintegrasikan analitik interaktif yang aman ke dalam aplikasi mereka sambil mempertahankan kontrol halus atas akses pengguna. Solusi ini menggunakan layanan AWS terkelola dan otomatisasi untuk menyederhanakan proses penyematan, meningkatkan keamanan, dan memastikan skalabilitas untuk memenuhi kebutuhan bisnis yang terus berkembang.

Target audiens dan kasus penggunaan:

- Pengembang frontend yang ingin menyematkan analitik ke dalam aplikasi React
- Tim produk perangkat lunak sebagai layanan (SaaS) yang ingin menawarkan visualisasi data per pengguna atau berbasis peran
- Arsitek solusi yang tertarik untuk mengintegrasikan AWS analitik ke portal khusus
- Pengembang Business Intelligence (BI) yang ingin mengekspos visual ke pengguna yang diautentikasi tanpa memerlukan akses dasbor penuh
- Tim perusahaan yang ingin menyematkan bagan Quick Sight interaktif dalam alat internal

Prasyarat dan batasan

Prasyarat

Agar berhasil menerapkan pola ini, pastikan bahwa berikut ini sudah ada:

- Aktif Akun AWS — An Akun AWS dengan izin untuk menerapkan CloudFormation tumpukan dan membuat sumber daya Lambda, API Gateway, Amazon Cognito, CloudFront, dan Amazon S3.
- Akun Amazon Quick Sight — Akun Quick Sight aktif dengan setidaknya satu dasbor yang berisi visual. Untuk petunjuk penyiapan, lihat Tutorial: Membuat dasbor Amazon Quick Sight menggunakan data sampel dalam dokumentasi Amazon Quick Suite.
- Lingkungan pengembangan yang terdiri dari:
  - Node.js (versi 16 atau yang lebih baru)
  - npm atau benang terpasang
  - Vite sebagai alat build React
  - Bereaksi (versi 19.1.1)
- Berbagi dasbor - Dasbor harus dibagikan di Quick Sight dan pelaksana harus masuk untuk mengakses visual atau dasbor yang disematkan.

Batasan

- Pola ini menggunakan metode embedding pengguna terdaftar, yang mengharuskan pelaksana untuk memiliki akun Quick Sight yang aktif.
- Akses dibatasi pada dasbor dan visual yang secara eksplisit dibagikan dengan pengguna Quick Sight yang diautentikasi yang menerapkan pola ini. Jika pelaksana tidak memiliki hak akses yang benar, pembuatan URL yang disematkan akan gagal dan visual tidak akan dimuat.

- CloudFormation Tumpukan harus diterapkan di Wilayah AWS tempat Quick Sight, API Gateway, dan Amazon Cognito didukung. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah.

Versi produk

- Quick Sight Embedding SDK versi 2.10.1
- Bereaksi versi 19.1.1
- Node.js versi 16 atau yang lebih baru untuk memastikan kompatibilitas dengan versi React dan Vite terbaru yang digunakan dalam solusi ini

Arsitektur

Arsitektur target

Diagram berikut menunjukkan arsitektur dan alur kerja untuk pola ini.

Dalam alur kerja ini:

1. Pengguna mengakses aplikasi. Pengguna membuka aplikasi web React dengan menggunakan browser. Permintaan diarahkan ke CloudFront distribusi, yang bertindak sebagai jaringan pengiriman konten untuk aplikasi.
2. AWS WAF memfilter permintaan berbahaya. Sebelum permintaan mencapai CloudFront, itu lewat AWS WAF. AWS WAF memeriksa lalu lintas dan memblokir permintaan berbahaya atau mencurigakan berdasarkan aturan keamanan.
3. Amazon S3 menyajikan file statis. Jika permintaan bersih, CloudFront ambil aset frontend statis (HTML, JS, CSS) dari bucket S3 pribadi dengan menggunakan kontrol akses asal (OAC) dan mengirimkannya ke browser.
4. Pengguna masuk. Setelah aplikasi dimuat, pengguna masuk melalui Amazon Cognito, yang mengautentikasi pengguna dan mengembalikan token web JSON (JWT) yang aman untuk akses API resmi.
5. Aplikasi membuat permintaan API. Setelah login, aplikasi React membuat panggilan aman ke `/get-embed-url` titik akhir pada API Gateway, dan meneruskan token JWT di header permintaan untuk otentikasi.

6. Token divalidasi. API Gateway memvalidasi token dengan menggunakan otorisasi Amazon Cognito. Jika token valid, permintaan dilanjutkan; jika tidak, ditolak dengan respons 401 (tidak sah).
7. Permintaan diarahkan ke Lambda untuk diproses. Permintaan yang divalidasi kemudian diteruskan ke fungsi Lambda backend. Fungsi ini bertanggung jawab untuk menghasilkan URL embed untuk visual Quick Sight yang diminta.
8. Lambda menghasilkan URL embed dari Quick Sight. IAM menggunakan peran IAM dengan izin yang sesuai untuk memanggil Quick Sight `GenerateEmbedUrlForRegisteredUser` API untuk menghasilkan URL visual yang aman dan tercakup pengguna.
9. Lambda mengembalikan URL embed ke API Gateway. Lambda mengirimkan URL embed yang dihasilkan kembali ke API Gateway sebagai bagian dari respons JSON. Respons ini kemudian disiapkan untuk pengiriman ke frontend.
10. URL embed dikirim ke browser. URL embed dikembalikan ke browser sebagai respons API.
11. Visual ditampilkan kepada pengguna. Aplikasi React menerima respons dan menggunakan Quick Sight Embedding SDK untuk merender visual tertentu kepada pengguna.

Otomatisasi dan skala

Penerapan backend dan frontend sepenuhnya otomatis dengan menggunakan CloudFormation, yang menyediakan semua sumber daya yang diperlukan, AWS termasuk Amazon Cognito, Lambda, API Gateway, Amazon S3,,, peran IAM CloudFront AWS WAF, dan Amazon dalam satu penerapan. CloudWatch

Otomatisasi ini memastikan infrastruktur yang konsisten dan berulang di semua lingkungan. Semua komponen menskalakan secara otomatis: Lambda menyesuaikan dengan pemanggilan fungsi, CloudFront menyajikan konten yang di-cache secara global, dan skala API Gateway dengan permintaan yang masuk.

Alat

Layanan AWS

- Amazon Quick Sight adalah layanan intelijen bisnis cloud-native yang membantu Anda membuat, mengelola, dan menyematkan dasbor dan visual interaktif.
- Amazon API Gateway mengelola APIs yang bertindak sebagai jembatan antara aplikasi React dan layanan backend.

- AWS Lambdaadalah layanan komputasi tanpa server yang digunakan pola ini untuk menghasilkan penyematan Quick Sight yang aman URLs secara dinamis, dan menskalakan secara otomatis berdasarkan permintaan.
- Amazon Cognito menyediakan otentikasi dan otorisasi bagi pengguna, dan mengeluarkan token aman untuk akses API.
- Amazon S3 menghosting aset frontend statis untuk pola ini, dan menyajikannya dengan aman. CloudFront
- Amazon CloudFront menghadirkan konten frontend secara global dengan latensi rendah dan terintegrasi dengan AWS WAF untuk pemfilteran lalu lintas.
- AWS WAFmelindungi aplikasi web dari lalu lintas berbahaya dengan menerapkan aturan keamanan dan pembatasan tarif.
- AWS CloudFormationmengotomatiskan penyediaan dan konfigurasi semua sumber daya aplikasi dalam satu penerapan.
- Amazon CloudWatch mengumpulkan log dan metrik dari Lambda, API Gateway, dan AWS WAF untuk pemantauan dan pemecahan masalah.

Alat pengembangan

- React JS adalah kerangka kerja frontend yang digunakan pola ini untuk membangun aplikasi web dan mengintegrasikan visual Quick Sight yang disematkan.
- Vite adalah alat build yang digunakan untuk pengembangan cepat dan pembuatan produksi aplikasi React yang dioptimalkan.
- Quick Sight Embedding SDK memfasilitasi penyematan visual Quick Sight ke dalam aplikasi React dan memungkinkan interaksi yang mulus antara aplikasi dan visual.

Repositori kode

Kode untuk pola ini tersedia di GitHub Amazon Quick Sight Visual Embedding di repositori React.

Praktik terbaik

Pola ini secara otomatis menerapkan praktik terbaik keamanan berikut:

- Menggunakan kumpulan pengguna Amazon Cognito untuk otentikasi berbasis JWT, dengan otentikasi multi-faktor opsional (MFA).

- Mengamankan APIs dengan otorisasi Amazon Cognito dan memberlakukan kebijakan IAM dengan hak istimewa terkecil di semua layanan.
- Menerapkan penyematan pengguna terdaftar Quick Sight dan penyediaan otomatis pengguna dengan peran pembaca.
- Menerapkan enkripsi dalam perjalanan yang mendukung TLS 1.2 dan versi yang lebih baru melalui CloudFront dan HTTPS.
- Mengenkripsi data saat istirahat dengan menggunakan AES-256 untuk Amazon S3 dengan versi dan OAC.
- Mengonfigurasi paket penggunaan API Gateway dengan pembatasan dan kuota.
- Mengamankan Lambda dengan konkurensi cadangan dan perlindungan variabel lingkungan.
- Mengaktifkan pencatatan untuk Amazon S3,, CloudFront Lambda, dan API Gateway; memantau layanan dengan menggunakan. CloudWatch
- Mengenkripsi log, menerapkan kontrol akses, dan memberlakukan kebijakan penolakan untuk unggahan non-HTTPS atau tidak terenkripsi.

Selain itu, kami merekomendasikan yang berikut:

- Gunakan CloudFormation untuk mengotomatiskan penerapan dan mempertahankan konfigurasi yang konsisten di seluruh lingkungan.
- Pastikan setiap pengguna memiliki izin Quick Sight yang benar untuk mengakses visual yang disematkan.
- Lindungi titik akhir API Gateway dengan otorisasi Amazon Cognito dan terapkan prinsip hak istimewa paling sedikit untuk semua peran IAM.
- Simpan informasi sensitif seperti Amazon Resource Names (ARNs) dan IDs dalam variabel lingkungan alih-alih melakukan hardcoding.
- Optimalkan fungsi Lambda dengan mengurangi dependensi dan meningkatkan kinerja cold-start. Untuk informasi lebih lanjut, lihat posting AWS blog Mengoptimalkan kinerja start dingin AWS Lambda menggunakan strategi priming lanjutan dengan. SnapStart
- Tambahkan CloudFront domain Anda ke daftar izin Quick Sight untuk mengaktifkan penyematan visual yang aman.
- Pantau kinerja dan keamanan dengan menggunakan CloudWatch dan AWS WAF untuk pencatatan, peringatan, dan perlindungan lalu lintas.

Praktik terbaik lainnya yang direkomendasikan

- Gunakan domain khusus dengan sertifikat SSL AWS Certificate Manager untuk memberikan pengalaman pengguna yang aman dan bermerek.
- Enkripsi data CloudWatch dan log Amazon S3 dengan menggunakan kunci AWS Key Management Service terkelola pelanggan AWS KMS() untuk kontrol yang lebih besar atas enkripsi.
- Perluas AWS WAF aturan dengan pemblokiran geografis, injeksi SQL (SQLi), perlindungan skrip lintas situs (XSS), dan filter khusus untuk pencegahan ancaman yang ditingkatkan.
- Aktifkan CloudWatch alarm, AWS Config, dan AWS CloudTrail untuk pemantauan waktu nyata, audit, dan kepatuhan konfigurasi.
- Terapkan kebijakan IAM terperinci, terapkan rotasi kunci API, dan izinkan akses lintas akun hanya jika benar-benar diperlukan.
- Lakukan penilaian keamanan rutin untuk memastikan keselarasan dengan kerangka kepatuhan seperti Kontrol Sistem dan Organisasi 2 (SOC 2), Peraturan Perlindungan Data Umum (GDPR), dan Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA).

Epik

Siapkan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori.	Kloning GitHub repositori untuk solusi ini ke sistem lokal Anda dan arahkan ke direktori proyek:
	`git clone https://github.com/aws-samples/sample-quicksight-visual-embedding.git`
	`cd sample-quicksight-visual-embedding`
	Repositori ini berisi CloudFormation template dan kode
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber React yang diperlukan untuk menyebarkan solusi.	

Menyebarkan tumpukan CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan template.	1. Masuk ke Konsol Manajemen AWS dan buka AWS CloudFormation konsol. 2. Pada halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar) . 3. Pada halaman Buat tumpukan, untuk Prasyarat - Siapkan template, pilih Pilih template yang ada. 4. Untuk Tentukan templat, pilih Unggah file templat. 5. Unggah **`template.yaml`** file dari GitHub repositor i lokal Anda, lalu pilih Berikutnya. 6. Pada halaman Tentukan detail tumpukan, masukkan nama tumpukan (misalnya ,`quicksight-embeddi ng-stack`). 7. Pada halaman Configure stack options, pertahank	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an pengaturan default, dan centang kotak untuk mengakui pembuatan sumber daya IAM. Pilih Berikutnya. 8. Pada halaman Tinjau dan buat, pilih Kirim. Untuk informasi selengkap nya, lihat Membuat dan mengelola tumpukan dalam CloudFormation dokumentasi.	
Pantau pembuatan tumpukan.	Pantau tumpukan di tab Peristiwa hingga statusnya CREATE_COMPLETE.	Administrator AWS
Ambil output tumpukan.	1. Saat penerapan selesai, pilih tab Output. 2. Perhatikan output kunci:`ApiGatewayUrl` ,,`UserPoolId` , `UserPoolClientId` `CloudFrontDomainName` , dan`S3BucketName` . Anda akan menggunakan informasi ini dalam langkah-langkah selanjutnya.	Administrator AWS

Konfigurasikan lingkungan frontend

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ambil pengidentifikasi visual Quick Sight.	1. Buka dasbor Quick Sight. 2. Perhatikan ID dasbor, ID lembar, dan ID visual dari setiap visual yang Anda rencanakan untuk disematkan dalam aplikasi web Anda.	Administrator Penglihatan Cepat
Konfigurasikan lingkungan React lokal Anda.	Untuk menyiapkan lingkungan React lokal Anda dan menautkannya ke AWS sumber daya, buat `.env` file di `my-app/` folder GitHub repositori lokal Anda. Isi file dengan: • Anda Wilayah AWS • Informasi kumpulan Amazon Cognito (dari output CloudFormation tumpukan) • Titik akhir API Gateway (dari output CloudFormation tumpukan) • Penglihatan Cepat visual IDs (`DashboardId` ,`SheetId`,`VisualId`) Berikut adalah contoh `.env` file: `VITE_AWS_REGION=us-east-1`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`# Cognito Configura tion (from CloudForm ation outputs) VITE_USER_POOL_ID= us-east-1_xxxxxxxx x VITE_USER_POOL_WEB _CLIENT_ID=xxxxxxx xxxxxxxxxxxxxxxxxx`
	
	`# API Configuration (from CloudFormation outputs) VITE_API_URL=https :/your-api-id.exec ute-api.us-east-1. amazonaws.com/prod`
	
	`# QuickSight Visual Configuration VITE_DASHBOARD_ID=your -dashboard-id VITE_SHEET_ID=your- sheet-id VITE_VISUAL_ID=your- visual-id`	

Mengatur otentikasi pengguna

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat atau Kelola Pengguna di Cognito	Untuk mengaktifkan akses pengguna yang diautentikasi ke visual Quick Sight yang disematkan, Anda terlebih dahulu membuat pengguna di Amazon Cognito:	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Di konsol Amazon Cognito, pilih Kumpulan pengguna, lalu pilih kumpulan yang cocok `UserPoolId` (dari CloudFormation output). 2. Tambahkan pengguna ke kolam renang. Anda dapat memilih dari dua opsi: • Di Amazon Cognito, buat pengguna secara manual dengan alamat email dan kata sandi sementara (untuk informasi selengkap nya, lihat dokumentasi Amazon Cognito) — atau — • Aktifkan pendaftaran mandiri di UI dengan menyetel `hideSignUp={false}` di file. `my-app/src/auth/AuthWrapper.jsx`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan akses dasbor Quick Sight	Untuk menyediakan akses ke visual Quick Sight, berikan akses izin Viewer ke pengguna yang diautentikasi: 1. Pada konsol Quick Sight, pilih Dasbor. 2. Pilih dasbor yang ingin Anda bagikan dengan memilih namanya. 3. Di sudut kanan atas halaman dasbor, pilih Bagikan. 4. Di panel berbagi, pilih Bagikan dengan pengguna tertentu. 5. Untuk menambahkan pengguna baru, pilih Undang pengguna atau ikon plus (+). 6. Di bidang email, masukkan alamat email lengkap penerima. (Email ini harus sama persis dengan login Amazon Cognito pengguna) 7. Dari menu Izin di sebelah bidang email, pilih Penampil. 8. Untuk mengirim undangan dan memberikan akses, pilih Bagikan.	Administrator Penglihatan Cepat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Setiap pengguna akan menerima email dengan tautan ke dasbor. Anda dapat mengubah izin kapan saja melalui menu Bagikan. Untuk informasi selengkap nya, lihat Memberikan akses kepada pengguna dan grup Amazon Quick Sight individua l ke dasbor di Amazon Quick Sight dalam dokumentasi Amazon Quick Suite.	

Membangun dan menerapkan frontend React

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal dependensi dan bangun proyek.	Di direktori aplikasi React, jalankan perintah berikut untuk menghasilkan file produksi yang dioptimalkan:
	
	`cd my-app`
	`npm install`
	`npm run build`
		Pengembang aplikasi

Unggah file build ke Amazon S3.	Unggah semua file dari `my-app/dist/` direktori ke bucket S3 yang disediakan oleh CloudFormation (jangan unggah folder itu sendiri).	Pengembang aplikasi
Buat CloudFront pembatalan.	Di CloudFront konsol, buat pembatalan jalur `/*` untuk	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menyegarkan konten yang di-cache setelah penerapan . Untuk petunjuk, lihat Membatalkan file dalam dokumentasi. CloudFront	

Konfigurasikan daftar yang diizinkan Quick Sight

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan CloudFron t domain ke daftar yang diizinkan Quick Sight.	Untuk mengaktifkan CloudFront domain Anda menyematkan visual Quick Sight dengan aman: 1. Pada konsol Quick Sight, pilih Manage Quick Sight, Domains dan Embedding. 2. Tambahkan CloudFront domain ke kotak Domain (misalnya,`https://d1234567890abc.cloudfront.net`). 3. Pilih Tambahkan domain.	Administrator Penglihatan Cepat

Akses aplikasi dan verifikasi fungsionalitas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka aplikasi React.	Gunakan CloudFront domain (dari CloudFormation output) untuk membuka aplikasi web	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	React yang digunakan di browser.	
Verifikasi otentikasi.	Masuk ke aplikasi menggunakan kredensi Amazon Cognito untuk memverifikasi alur otentikasi dan validasi JWT melalui API Gateway.	Pemilik aplikasi
Verifikasi visual yang disematkan.	Konfirmasikan bahwa visual Quick Sight dimuat dengan benar di dalam aplikasi berdasarkan izin akses khusus pengguna.	Pemilik aplikasi
Validasi konektivitas API dan Lambda.	Konfirmasikan bahwa aplikasi dapat berhasil memanggil `/get-embed-url` API dan mengambil sematan Quick Sight yang valid URLs tanpa kesalahan.	Pemilik aplikasi

Memantau dan memelihara aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Monitor dengan menggunakan CloudWatch.	Anda dapat menggunakan alat AWS observabilitas untuk memantau aplikasi dan untuk mempertahankan kinerja yang aman dan terukur dalam produksi. Tinjau log Lambda, metrik API Gateway, dan peristiwa	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	autentikasi Amazon Cognito CloudWatch untuk memastikan kesehatan aplikasi dan mendeteksi anomali.	
Lacak AWS WAF dan CloudFront log.	Periksa AWS WAF log untuk permintaan yang diblokir atau mencurigakan dan log CloudFront akses untuk metrik kinerja dan caching.	Administrator AWS

Pemecahan masalah

Isu	Solusi
Kesalahan “Domain tidak diizinkan”	1. Pada konsol Quick Sight, pilih Manage Quick Sight, Domains dan Embedding. 2. Tambahkan CloudFront domain Anda (misalnya,`https://d1234567890abc.cloudfront.net`). 3. Simpan konfigurasi dan muat ulang aplikasi.
Kesalahan autentikasi	Kemungkinan penyebab: • ID kumpulan pengguna Amazon Cognito atau ID klien aplikasi yang salah dalam file. `.env` • Pengguna tidak dikonfirmasi di kumpulan pengguna Amazon Cognito. • Token JWT tidak valid atau hilang dalam permintaan API. Solusi:

Isu	Solusi
	• Verifikasi nilai konfigurasi Amazon Cognito dalam file. `.env` • Di konsol Amazon Cognito, periksa tab Pengguna untuk memverifikasi bahwa pengguna aktif. • Konfirmasikan bahwa header Otorisasi berisi token JWT yang valid. • Pada konsol API Gateway, periksa Authorizers untuk memverifikasi bahwa API Gateway menggunakan otorisasi Amazon Cognito untuk otentikasi.

Isu	Solusi
Kesalahan API Gateway	Kemungkinan penyebab: • Otorisasi Amazon Cognito yang hilang atau salah konfigurasi di API Gateway. • Izin Lambda tidak memadai untuk memanggil Quick Sight API. • Ketidakcocokan URL API antara `.env` dan CloudFormation output. Solusi: • Konfirmasikan kembali bahwa URL pemanggilan API Gateway ada di file Anda`.env`. • Di konsol API Gateway, periksa Authorizers untuk memverifikasi penyiapan Amazon Cognito. • Lampirkan kebijakan IAM yang diperlukan dengan `quicksight:GenerateEmbedUrlForRegisteredUser` peran eksekusi Lambda. • Tinjau CloudWatch log untuk API mendetail atau kesalahan izin. Untuk informasi selengkapnya, lihat dokumentasi API Gateway.

Isu	Solusi
Visual Quick Sight tidak dimuat	Kemungkinan penyebab: • ID dasbor Quick Sight salah, ID lembar, atau ID visual dalam konfigurasi lingkungan. • Tidak ada akses pengguna Quick Sight ke visual atau dasbor. • Tidak ada konfigurasi berbagi sumber daya lintas asal (CORS) di API Gateway. Solusi: • Konfirmasikan Quick Sight IDs dari menu tiga titik di sudut kanan atas visual dasbor. • Konfirmasikan bahwa pengguna Quick Sight memiliki peran Viewer untuk dasbor bersama. • Aktifkan CORS di API Gateway dan terapkan ulang API. Untuk informasi selengkapnya, lihat dokumentasi API Gateway. • Periksa konsol browser untuk CORS untuk kesalahan terkait jaringan.

Isu	Solusi
Kesalahan “Pengguna tidak memiliki akses”	Kemungkinan penyebab: • Email pengguna tidak dibagikan dengan dasbor Quick Sight. • Peran atau izin pengguna salah dalam Quick Sight. Solusi: 1. Di konsol Quick Sight, buka dasbor Anda, lalu pilih Bagikan, Bagikan Dasbor. 2. Tambahkan alamat email pengguna (cocok dengan email login Amazon Cognito). 3. Tetapkan peran izin Viewer. 4. Minta pengguna untuk keluar dan masuk lagi untuk menyegarkan izin.

Sumber daya terkait

AWS dokumentasi

- Mendaftar untuk berlangganan Amazon Quick Suite
- Analisis tertanam untuk Amazon Quick Sight
- Referensi API Penglihatan Cepat - GenerateEmbedUrlForRegisteredUser
- Kumpulan pengguna Amazon Cognito
- AWS Lambda Panduan Pengembang
- Panduan Pengembang Amazon API Gateway
- Pemantauan dasar dan pemantauan terperinci di CloudWatch
- AWS CloudFormation Panduan Pengguna
- Panduan CloudFront Pengembang Amazon
- AWS WAF Panduan Pengembang

Tutorial dan video

- Analisis tertanam untuk Amazon Quick Sight
- YouTube Tutorial Penglihatan Cepat Amazon

Jelajahi pengembangan aplikasi web cloud-native full-stack dengan Green Boost

Ben Stickley dan Amiin Samatar, Amazon Web Services

Ringkasan

Menanggapi kebutuhan pengembang yang terus berkembang, Amazon Web Services (AWS) mengakui permintaan kritis untuk pendekatan yang efisien untuk mengembangkan aplikasi web cloud-native. Fokus AWS adalah membantu Anda mengatasi hambatan umum yang terkait dengan penerapan aplikasi web di AWS Cloud. Dengan memanfaatkan kemampuan teknologi modern seperti AWS Cloud Development Kit (TypeScriptAWS CDK), React, dan Node.js, pola ini bertujuan untuk merampingkan dan mempercepat proses pengembangan.

Didukung oleh toolkit Green Boost (GB), pola ini menawarkan panduan praktis untuk membangun aplikasi web yang sepenuhnya menggunakan kemampuan AWS yang luas. Ini bertindak sebagai peta jalan yang komprehensif, menuntun Anda melalui proses penerapan aplikasi web CRUD (Buat, Baca, Perbarui, Hapus) mendasar yang terintegrasi dengan Amazon Aurora PostgreSQL Edisi yang kompatibel. Ini dicapai dengan menggunakan antarmuka baris perintah Green Boost (Green Boost CLI) dan membangun lingkungan pengembangan lokal.

Setelah keberhasilan penerapan aplikasi, pola menggali komponen kunci dari aplikasi web, termasuk desain infrastruktur, pengembangan backend dan frontend, dan alat-alat penting seperti cdk-dia untuk visualisasi, memfasilitasi manajemen proyek yang efisien.

Prasyarat dan batasan

Prasyarat

- Git diinstal
- Visual Studio Code (VS Code) diinstal
- AWS Command Line Interface (AWS CLI) diinstal
- AWS CDK Toolkit diinstal
- Node.js 18 diinstal, atau Node.js 18 dengan pnpm diaktifkan
- pnpm diinstal, jika itu bukan bagian dari instalasi Node.js Anda
- Keakraban dasar dengan TypeScript, AWS CDK, Node.js, dan React
- Akun AWS yang aktif

- Akun AWS di-bootstrap dengan menggunakan AWS CDK di. `us-east-1` Wilayah `us-east-1` AWS diperlukan untuk mendukung fungsi Amazon CloudFront Lambda @Edge.
- Kredensyal keamanan AWS, termasuk`AWS_ACCESS_KEY_ID`, dikonfigurasi dengan benar di lingkungan terminal Anda
- Untuk pengguna Windows, terminal dalam mode administrator (untuk mengakomodasi cara pnpm menangani modul node)

Versi produk

- AWS SDK untuk JavaScript versi 3
- AWS CDK versi 2
- AWS CLI versi 2.2
- Node.js versi 18
- Bereaksi versi 18

Arsitektur

Tumpukan teknologi target

- Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL
- Amazon CloudFront
- Amazon CloudWatch
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Lambda
- AWS Secrets Manager
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Storage Service (Amazon S3)
- AWS WAF

Arsitektur Target

Diagram berikut menunjukkan bahwa permintaan pengguna melewati Amazon CloudFront, AWS WAF, dan AWS Lambda sebelum berinteraksi dengan bucket S3, database Aurora, instans EC2 ,

dan akhirnya menjangkau pengembang. Administrator, di sisi lain, menggunakan Amazon SNS dan CloudWatch Amazon untuk tujuan notifikasi dan pemantauan.

Untuk mendapatkan tampilan yang lebih mendalam pada aplikasi setelah penerapan, Anda dapat membuat diagram dengan menggunakan cdk-dia, seperti yang ditunjukkan pada contoh berikut.

Diagram ini menampilkan arsitektur aplikasi web dari dua sudut yang berbeda. Diagram cdk-dia menawarkan tampilan teknis terperinci tentang infrastruktur AWS CDK, menyoroti layanan AWS tertentu seperti Amazon Aurora PostgreSQL yang kompatibel dengan dan AWS Lambda. Sebaliknya, diagram lain mengambil perspektif yang lebih luas, menekankan aliran logis data dan interaksi pengguna. Perbedaan utama terletak pada tingkat detail: CDK-dia menggali seluk-beluk teknis, sedangkan diagram pertama memberikan tampilan yang lebih berpusat pada pengguna.

Pembuatan diagram cdk-dia tercakup dalam epik Memahami infrastruktur aplikasi dengan menggunakan AWS CDK.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon CloudFront mempercepat distribusi konten web Anda dengan mengirimkannya melalui jaringan pusat data di seluruh dunia, yang menurunkan latensi dan meningkatkan kinerja.
- Amazon CloudWatch membantu Anda memantau metrik sumber daya AWS Anda dan aplikasi yang Anda jalankan di AWS secara real time.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- AWS Systems Manager membantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan manajemen aplikasi dan sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola sumber daya AWS Anda dengan aman dalam skala besar. Pola ini menggunakan AWS Systems Manager Session Manager.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- AWS WAF adalah firewall aplikasi web yang membantu Anda memantau permintaan HTTP dan HTTPS yang diteruskan ke sumber daya aplikasi web Anda yang dilindungi

Alat-alat lainnya

- Git adalah sistem kontrol versi terdistribusi open-source.
- Green Boost adalah toolkit untuk membangun aplikasi web di AWS.
- Next.js adalah framework React untuk menambahkan fitur dan optimasi.
- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.
- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.
- pnpm adalah manajer paket untuk dependensi proyek Node.js.

Praktik terbaik

Lihat bagian Epik untuk informasi lebih lanjut tentang rekomendasi berikut:

- Pantau infrastruktur dengan menggunakan CloudWatch Dasbor Amazon dan alarm.

- Menerapkan praktik terbaik AWS dengan menggunakan cdk-nag untuk menjalankan analisis infrastruktur statis sebagai kode (IAc).
- Buat penerusan port DB melalui tunneling SSH (Secure Shell) dengan Systems Manager Session Manager, yang lebih aman daripada memiliki alamat IP yang terbuka secara publik.
- Kelola kerentanan dengan menjalankan`pnpm audit`.
- Terapkan praktik terbaik dengan menggunakan ESLintuntuk melakukan analisis TypeScript kode statis, dan Prettier untuk membakukan pemformatan kode.

Epik

Menyebarkan aplikasi web CRUD dengan Aurora PostgreSQL yang kompatibel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Green Boost CLI.	Untuk menginstal Green Boost CLI, jalankan perintah berikut. `pnpm add -g gboost`	Pengembang aplikasi
Buat aplikasi GB.	1. Untuk membuat aplikasi dengan menggunakan Green Boost, jalankan perintah`gboost create`. 2. Pilih `CRUD App with Aurora PostgreSQL` template.	Pengembang aplikasi
Instal dependensi dan terapkan aplikasi.	1. Arahkan ke direktori proyek:`cd <your directory> .` 2. Untuk menginstal dependensi, jalankan perintah. `pnpm i` 3. Arahkan ke direktori infra:`cd infra`.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk menerapkan aplikasi secara lokal, jalankan perintah. `pnpm deploy:local` Ini adalah alias untuk `cdk deploy ...` perintah yang didefinisikan dalam`infra/package.json` . Tunggu hingga penerapan selesai (sekitar 20 menit). Saat Anda menunggu, pantau CloudFormation tumpukan AWS di CloudFormation konsol. Perhatikan bagaimana konstruksi didefinisikan dalam peta kode ke sumber daya yang digunakan. Tinjau tampilan pohon Konstruksi CDK di konsol. CloudFormation	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses aplikasi.	Setelah menerapkan aplikasi GB secara lokal, Anda dapat mengaksesnya menggunakan URL. CloudFront URL dicetak dalam output terminal, tetapi bisa sedikit berlebihan untuk ditemukan. Untuk menemukannya lebih cepat, gunakan langkah-langkah berikut: 1. Buka terminal tempat Anda menjalankan `pnpm deploy:local` perintah. 2. Cari bagian dalam output terminal yang menyerupai teks berikut. `myapp5stickbui9C39A55A.CloudFrontDomainName = d1q16n5pof924c.cloudfront.net` URL akan unik untuk penyebaran Anda. Atau, Anda dapat menemukan CloudFront URL dengan mengakses CloudFront konsol Amazon: 1. Masuk ke AWS Management Console dan navigasikan ke CloudFront layanan.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Cari distribusi terbaru yang diterapkan dalam daftar. Salin Nama Domain yang terkait dengan distribus i. Ini akan terlihat mirip dengan`your-unique-id.cloudfront.net` .	

Monitor dengan menggunakan Amazon CloudWatch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat CloudWatch Dasbor.	1. Buka CloudWatch konsol dan pilih Dasbor. 2. Pilih dasbor yang memiliki nama - -dashboard <appId><stageName>. 3. Tinjau dasbor. Sumber daya apa yang sedang dipantau? Metrik apa yang sedang direkam? Dasbor ini dimungkinkan oleh konstruksi cdk-monitoring-constructssumber terbuka.	Pengembang aplikasi
Aktifkan peringatan.	CloudWatch Dasbor membantu Anda memantau aplikasi web secara aktif. Untuk memantau aplikasi web Anda secara pasif, Anda dapat mengaktifkan peringatan.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	1. Arahkan ke`/infra/src/app/stateless/monitor-stack.ts` , yang mendefinisikan tumpukan monitor.
	2. Hapus komentar pada baris berikut, dan ganti `admin@example.com` dengan alamat email Anda.
	`onAlarmTopic.addSubscription(new EmailSubscription("admin@example.com"));`
	3. Tambahkan informasi impor berikut ke bagian atas file.
	`import { EmailSubscription } from "aws-cdk-lib/aws-sns-subscriptions";`
	4. Di dalam`infra/`, jalankan perintah berikut.
	`cdk deploy "*/monitor" --exclusively.`
	5. Untuk mengonfirmasi langganan Anda ke topik SNS yang diintitasi saat alarm pemantauan dimulai, pilih tautan dalam pesan email.	

Memahami infrastruktur aplikasi dengan menggunakan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat diagram arsitektur.	Buat diagram arsitektur aplikasi web Anda dengan menggunakan cdk-dia. Memvisualisasikan arsitektur membantu meningkatkan pemahaman dan komunikasi di antara anggota tim. Ini memberikan gambaran yang jelas tentang komponen sistem dan hubungannya. 1. Instal Graphviz. 2. Di dalam `infra/`, jalankan perintah `pnpm cdk-dia`. 3. Lihat Anda `infra/diagram.png` .	Pengembang aplikasi
Gunakan cdk-nag untuk menerapkan praktik terbaik.	Gunakan cdk-nag untuk membantu Anda menjaga infrastruktur yang aman dan patuh dengan menerapkan praktik terbaik, mengurangi risiko kerentanan keamanan dan kesalahan konfigurasi. 1. Jelajahi penegakan praktik terbaik cdk-nag melalui bagian aturannya, termasuk pemeriksaan dari Paket Aturan AWS Solutions Library. 2. Untuk melihat bagaimana cdk-nag memberlakukan	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	aturan, buat perubahan pada kode. Misalnya, di`infra/src/app/stateful/data-stacks.ts` , ubah `storageEncrypted: true` kestorageEncrypted: false . 3. Di dalam`infra/`, jalankan perintah`cdk synth "*/data"`. Selama sintesis, Anda akan menemukan kesalahan build yang menunjukkan pelanggaran aturan. `AwsSolutions-RDS2: The RDS instance or Aurora DB cluster does not have storage encryption enabled.` Kesalahan ini menunjukkan bagaimana cdk-nag adalah mekanisme keamanan untuk menegakkan praktik terbaik infrastruktur dan mencegah kesalahan konfigurasi keamanan. 4. Jika perlu, Anda juga dapat menekan aturan pada cakupan yang berbeda. Misalnya, untuk menekan AwsSolutions -RDS2,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tambahkan kode berikut di bawah instantiasi. `DbIamCluster` `NagSuppressions.addResourceSuppressions( cluster.node.findChild("Resource"), [ { id: "AwsSolutions-RDS2", reason: "Customer requirement necessitates having unencrypted DB storage", }, ], );` 5. Setelah penindasan, lari `cdk synth "*/data"` lagi. Aplikasi AWS CDK Anda sekarang harus berhasil mensintesis. Anda dapat menemukan semua aturan yang ditekan di`infra/cdk.out/assembly-<appId>-<stageName>/AwsSolutions-<appId>-<stageName>-${stackId}-NagReport.csv` .	

Mengevaluasi konfigurasi database dan skema

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memperoleh variabel lingkungan.	Untuk mendapatkan variabel lingkungan yang diperluka n, gunakan langkah-langkah berikut: 1. Untuk menemukan `DB_BASTION_ID` , masuk ke konsol, dan navigasikan ke EC2 konsol. Pilih Contoh (berjalan), dan temukan baris yang berisi - ssm-db-bastion Nama<stageNam e>. ID instance dimulai dengan i-. 2. Untuk menemukan **`DB_ENDPOINT`** , di konsol Amazon Relational Database Service (Amazon RDS), pilih Instans DB, dan pilih cluster regional yang memiliki pengenal DB yang dimulai dengan - -data-. <appId><s tageName> Temukan titik akhir instance penulis, yang diakhiri dengan rds.amazo naws.com.	Pengembang aplikasi
Menetapkan port forwarding.	Untuk membuat port forwardin g, gunakan langkah-langkah berikut:	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Instal plugin AWS Systems Manager Session Manager. 2. Mulai penerusan port dengan menjalankan `pnpm db:connect` di dalam `core/` untuk membuat koneksi aman melalui host bastion. 3. Setelah Anda melihat teks `Waiting for connections...,` di terminal Anda, terowongan SSH telah berhasil dibuat antara mesin lokal Anda dan server Aurora melalui EC2 host bastion.	
Sesuaikan batas waktu Systems Manager Session Manager.	(Opsional) Jika batas waktu sesi 20 menit default terlalu singkat, Anda dapat meningkatkannya hingga 60 menit di konsol Systems Manager dengan memilih Session Manager, Preferences, Edit, Idle session timeout.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Visualisasikan database.	pgAdmin adalah alat sumber terbuka yang mudah digunakan untuk mengelola database PostgreSQL. Ini menyederhanakan tugas database, memungkinkan Anda untuk secara efisien membuat, mengelola, dan mengoptimalkan database. Bagian ini memandu Anda untuk menginstal pgAdmin dan menggunakan fitur-fit urnya untuk manajemen database PostgreSQL. 1. Di Object Explorer, buka menu konteks (klik kanan) untuk Server, lalu pilih Register, Server. 2. Pada tab Umum, masukkan - <appId><stageName> untuk bidang Nama. 3. Untuk mengambil kata sandi DB, buka konsol AWS Secrets Manager, pilih rahasia yang memiliki deskripsi Dihasilkan oleh CDK untuk tumpukan: - -data <appId><stageName> , dan pilih kartu Nilai Rahasia. Pilih Ambil Nilai Rahasia, dan salin nilai Rahasia dengan kunci kata sandi.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pada tab Connection, masukkan 0.0.0 untuk name/address bidang Host, dan masukkan _admin untuk bidang Username. <appId> Untuk bidang Kata Sandi, gunakan rahasia yang Anda ambil sebelumnya. Pilih ya untuk Simpan kata sandi? lapangan. 5. Pilih Simpan. 6. <appId>Untuk melihat tabel, navigasikan ke -, Database, _db, Skema,, Tabel. <appId><stageName> <appId> 7. Buka menu konteks (klik kanan) untuk tabel item, lalu pilih Lihat/Edit Data, Semua Baris. 8. Jelajahi meja.	

Debug dengan Node.js

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Debug kasus penggunaan item buat.	Untuk men-debug kasus penggunaan item buat, ikuti langkah-langkah berikut: 1. Buka `core/src/modules/item/create-item.use-case.ts`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	file, dan masukkan kode berikut.
	
	`import { fileURLTo Path } from "node:url ";`
	
	`// existing create-it em.use-case.ts code here`
	
	`if (process.argv[1] === fileURLTo Path(import.meta.u rl)) { createItemUseCase( { description: "Item 1's Descripti on", name: "Item 1", }); }`
	
	2. Kode yang ditambahkan pada langkah sebelumny a memastikan `createIte mUseCase` fungsi akan dipanggil ketika modul ini dijalankan secara langsung. Tetapkan breakpoint pada baris dalam blok kode ini di mana Anda ingin memulai debugging. line-by-line
	
	1. Buka VS Code JavaScrip t Debug Terminal, lalu jalankan `pnpm tsx core/`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`src/modules/item/create-item.use-case.ts` untuk menjalankan kode dengan line-by-line debugging. Atau, Anda dapat menggunakan `console.log` pernyataan, tetapi pernyataan cetak bisa tidak memadai ketika Anda bekerja dengan logika bisnis yang kompleks. Line-by-linedebugging memberi Anda lebih banyak konteks.	

Kembangkan frontend

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan server pengembangan.	1. Arahkan ke`ui/`, dan jalankan `pnpm dev` untuk memulai server pengembangan Next.js. 2. Akses aplikasi web Anda secara lokal di `http://localhost:3000`. Server pengembangan Next.js disiapkan dengan umpan balik instan Fast Refresh pada pengeditan yang dilakukan pada komponen React Anda. 3. Bereksperimenlah dengan menyesuaikan warna bilah aplikasi. Buka `ui/src/co`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`mponents/theme/theme.tsx` file dan temukan bagian yang menentukan tema untuk bilah aplikasi. Di `colorSchemes.light.palette.primary` bagian ini, perbarui nilai utama dari `colors.lagoon` `kecolors.carrot` . Setelah melakukan perubahan ini, simpan file dan amati pembaruan di browser Anda. 4. Eksperimen dengan memodifikasi teks, komponen, dan menambahkan halaman baru.	

Perkakas dengan Green Boost

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan monorepo dan manajer paket pnpm.	1. Tinjau `pnpm-workspace.yaml` di root repositori GB Anda, dan perhatikan bagaimana ruang kerja didefinisikan. Untuk informasi selengkapnya tentang ruang kerja, lihat dokumentasi pnpm. 2. Tinjau`ui/package.json` , dan perhatikan bagaimana referensi ruang	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kerja `core/` dengan nama `"<appId>/core": "workspace:^",` paket. 3. Amati bagaimana TypeScript dan ESLint konfigurasi terpusat dalam paket utilitas yang didefinisikan di dalamnya`packages/` . Konfigurasi ini kemudian digunakan oleh paket aplikasi seperti`core/`,`infra/`, dan`ui/`. Ini berguna saat aplikasi Anda menskalakan dan Anda menentukan lebih banyak paket aplikasi, yang dapat mereferensikan paket utilitas tanpa menduplikasi kode konfigurasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip pnpm.	Jalankan perintah berikut di root repositori Anda: 1. Jalankan `pnpm lint`. Perintah ini menjalankan analisis kode statis dengan ESLint. 2. Jalankan `pnpm typecheck .` Perintah ini menjalankan TypeScript compiler untuk memeriksa jenis kode Anda. 3. Jalankan `pnpm test`. Perintah ini menjalankan Vitest untuk menjalankan pengujian unit. Perhatikan bagaimana perintah ini dijalankan di semua ruang kerja. Perintah didefinisikan di setiap `package.json#scripts` bidang ruang kerja.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan ESLint untuk analisis kode statis.	Untuk menguji kemampuan analisis kode statis ESLint, lakukan hal berikut: 1. Pertama, pastikan ESLint ekstensi VS Code (ID:`dbaeumer.vscode-eslint`) diinstal. Kami merekomendasikan juga menginstal VS Code Error Lens (ID:`usernameh w.errorlens`) untuk melihat kesalahan sebaris. 2. Dalam kode Anda, sengaja menyertakan baris kode yang menggunakan `eval()` fungsi, seperti yang ditunjukkan pada contoh berikut. `const userInput = "import("fs").then((fs) => console.log(fs.readFileSync("/etc/passwd", { encoding: "utf8" })))"; eval(userInput);` **Important** Ini hanya untuk tujuan pengujian. Penggunaan `eval()` dianggap berpotensi	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	berbahaya dan harus dihindari karena risiko keamanan. 3. Setelah Anda memasukkan `eval()` baris, buka editor kode Anda untuk mengonfirmasi bahwa ESLint menunjukkan bau kode dengan menggunakan coretan merah. 4. Tinjau ESLint plugin dan konfigurasi di`packages/eslint-config-{node,next}/.eslintrc.cjs` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kelola dependensi dan kerentanan.	1. Untuk mengidentifikasi Kerentanan Umum dan Exposures (CVEs), jalankan `pnpm audit` di root repositori Anda. Anda akan melihat Tidak ada kerentanan yang diketahui ditemukan. 2. Instal paket yang sengaja rentan di dalam `core/` dengan menjalank anpnpm `add minimist@0.2.3` , dan kemudian jalankan`pnpm audit`. Perhatikan kerentanan yang dilaporkan. 3. Copot pemasangan paket rentan di dalamnya `core/` dengan menjalankan`pnpm remove minimist`.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kait pra-komit dengan Husky.	1. Buat beberapa perubahan kecil dalam TypeScript file di seluruh repositori. Perubahan dapat menjadi dasar seperti menambahkan komentar. 2. Tahap dan lakukan perubahan ini dengan menggunakan `git add -A` dan kemudian`git commit -m "test husky"`. Pemicu kait pra-komit Husky, yang didefinisikan dalam`.husky/pre-commit`, menjalankan perintah. `pnpm lint-staged` 3. Perhatikan bagaimana lint-staged menjalankan perintah yang ditentukan dalam `*/.lintstagedrc.js` file di seluruh repositori pada file yang telah dipentaskan oleh Git. Alat-alat ini adalah mekanisme untuk membantu mencegah kode buruk masuk ke aplikasi Anda.	Pengembang aplikasi

Meruntuhkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus penyebaran dari akun Anda.	1. Untuk meruntuhkan infrastruktur yang Anda sediakan di epik pertama, dengan menjalankan`pnpm destroy:local . infra/` 2. Tunggu 15 menit setelah `pnpm destroy:local` selesai, lalu hapus fungsi Lambda @Edge yang dipertahankan dengan mencari ID aplikasi Anda di konsol Lambda. Fungsi Lambda @Edge direplikasi., yang membuatnya sulit untuk dihapus. Untuk informasi selengkapnya tentang menghapus fungsi Lambda @Edge, lihat CloudFront dokumentasi.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Tidak dapat membuat penerusan port	Pastikan kredensyal AWS Anda dikonfigurasi dengan benar dan memiliki izin yang diperlukan. Periksa kembali apakah variabel lingkungan bastion host ID (`DB_BASTION_ID`) dan

Isu	Solusi
	endpoint database (`DB_ENDPOINT`) disetel dengan benar. Jika Anda masih mengalami masalah, lihat dokumentasi AWS untuk memecahkan masalah koneksi SSH dan Session Manager.
Situs web tidak dimuat di `localhost:3000`	Konfirmasikan bahwa output terminal menunjukkan penerusan port yang berhasil, termasuk alamat penerusan. Pastikan tidak ada proses yang bertentangan menggunakan port 3000 pada mesin lokal Anda. Verifikasi bahwa aplikasi Green Boost dikonfigurasi dengan benar dan berjalan pada port yang diharapkan (3000). Periksa browser web Anda untuk ekstensi atau pengaturan keamanan apa pun yang mungkin memblokir koneksi lokal.
Pesan galat selama penerapan lokal () `pnpm deploy:local`	Tinjau pesan kesalahan dengan hati-hati untuk mengidentifikasi penyebab masalah. Verifikasi bahwa variabel lingkungan dan file konfigurasi yang diperlukan diatur dengan benar.

Sumber daya terkait

- Dokumentasi AWS CDK
- Dokumentasi Green Boost
- Dokumentasi Next.js
- Dokumentasi Node.js

- Dokumentasi React
- TypeScript dokumentasi

Jalankan pengujian unit untuk aplikasi Node.js dari GitHub dengan menggunakan AWS CodeBuild

Thomas Scott dan Jean-Baptiste Guillois, Amazon Web Services

Ringkasan

Pola ini menyediakan contoh kode sumber dan komponen pengujian unit kunci untuk API game Node.js. Ini juga mencakup instruksi untuk menjalankan pengujian unit ini dari GitHub repositori dengan menggunakan AWS CodeBuild, sebagai bagian dari alur kerja integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) Anda.

Pengujian unit adalah proses pengembangan perangkat lunak di mana berbagai bagian aplikasi, yang disebut unit, diuji secara individual dan independen untuk operasi yang benar. Pengujian memvalidasi kualitas kode dan mengkonfirmasi bahwa itu berfungsi seperti yang diharapkan. Pengembang lain juga dapat dengan mudah mendapatkan keakraban dengan basis kode Anda dengan berkonsultasi dengan tes. Tes unit mengurangi waktu refactoring di masa depan, membantu teknisi mempercepat basis kode Anda lebih cepat, dan memberikan kepercayaan pada perilaku yang diharapkan.

Pengujian unit melibatkan pengujian fungsi individual, termasuk fungsi AWS Lambda. Untuk membuat pengujian unit, Anda memerlukan kerangka pengujian dan cara memvalidasi tes (pernyataan). Contoh kode dalam pola ini menggunakan kerangka pengujian Mocha dan pustaka pernyataan Chai.

Untuk informasi selengkapnya tentang pengujian unit dan contoh komponen pengujian, lihat bagian Informasi tambahan.

Prasyarat dan batasan

- Akun AWS aktif dengan CodeBuild izin yang benar
- GitHub Akun (lihat instruksi untuk mendaftar)
- Git (lihat petunjuk instalasi)
- Editor kode untuk membuat perubahan dan mendorong kode Anda ke GitHub

Arsitektur

Pola ini mengimplementasikan arsitektur yang ditunjukkan pada diagram berikut.

Alat

Alat

- Git adalah sistem kontrol versi yang dapat Anda gunakan untuk pengembangan kode.
- AWS CodeBuild adalah layanan integrasi berkelanjutan yang dikelola sepenuhnya yang mengkompilasi kode sumber, menjalankan pengujian, dan menghasilkan paket perangkat lunak yang siap digunakan. Dengan CodeBuild, Anda tidak perlu menyediakan, mengelola, dan menskalakan server build Anda sendiri. CodeBuild menskalakan terus menerus dan memproses beberapa build secara bersamaan, sehingga build Anda tidak dibiarkan menunggu dalam antrian. Anda dapat memulai dengan cepat menggunakan lingkungan build yang dikemas sebelumnya, atau Anda dapat membuat lingkungan build kustom yang menggunakan alat build Anda sendiri. Dengan CodeBuild, Anda dikenakan biaya per menit untuk sumber daya komputasi yang Anda gunakan.

Kode

Kode sumber untuk pola ini tersedia di GitHub, di repositori aplikasi pengujian unit game Sample. Anda dapat membuat GitHub repositori Anda sendiri dari sampel ini (opsi 1) atau menggunakan repositori sampel secara langsung (opsi 2) untuk pola ini. Ikuti instruksi untuk setiap opsi di bagian berikutnya. Opsi yang Anda ikuti akan tergantung pada kasus penggunaan Anda.

Epik

Opsi 1 - Jalankan pengujian unit pada GitHub repositori pribadi Anda dengan CodeBuild

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat GitHub repositori Anda sendiri berdasarkan proyek sampel.	1. Masuk ke GitHub. 2. Buat repositori baru. Untuk instruksi, lihat GitHub dokumentasi. 3. Kloning dan dorong repositori sampel ke repositori baru di akun Anda.	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodeBuild proyek baru.	1. Masuk ke AWS Management Console dan buka CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home. 2. Pilih Buat proyek build. 3. Di bagian konfigurasi Proyek, untuk nama Proyek, ketik aws-tests-sample-node-js. 4. Di bagian Sumber, untuk penyedia Sumber, pilih GitHub. 5. Untuk Repositori, pilih Repositori di GitHub akun saya, lalu tempel URL ke repositori yang baru Anda buat. GitHub 6. Di bagian Peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini. 7. Untuk jenis acara, pilih PUSH. 8. Di bagian Lingkungan, pilih Managed image, Amazon Linux, dan image terbaru. 9. Tinggalkan pengaturan default untuk semua opsi	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lain, lalu pilih Buat proyek build.	
Mulai membangun.	Pada halaman Review, pilih Start build untuk menjalankan build.	Pengembang aplikasi, administrator AWS, AWS DevOps

Opsi 2 - Jalankan pengujian unit pada repositori publik dengan CodeBuild

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat proyek CodeBuild build baru.	1. Masuk ke AWS Management Console dan buka CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home. 2. Pilih Buat proyek build. 3. Di bagian konfigurasi Proyek, untuk nama Proyek, ketik aws-tests-sample-node-js. 4. Di bagian Sumber, untuk penyedia Sumber, pilih GitHub. 5. Untuk Repositori, pilih Repositori publik, lalu tempel URL:. https://github.com/aws-samples/node-js-tests-sample 6. Di bagian Lingkungan, pilih Managed image, Amazon Linux, dan image terbaru.	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Tinggalkan pengaturan default untuk semua opsi lain, lalu pilih Buat proyek build.	
Mulai membangun.	Pada halaman Review, pilih Start build untuk menjalankan build.	Pengembang aplikasi, administrator AWS, AWS DevOps

Menganalisis tes unit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat hasil tes.	Di CodeBuild konsol, tinjau hasil pengujian unit dari CodeBuild pekerjaan. Mereka harus cocok dengan hasil yang ditunjukkan di bagian Informasi tambahan. Hasil ini memvalidasi integrasi GitHub repositori dengan. CodeBuild	Pengembang aplikasi, administrator AWS, AWS DevOps
Terapkan webhook.	Anda sekarang dapat menerapkan webhook, sehingga Anda dapat secara otomatis memulai build setiap kali Anda mendorong perubahan kode ke cabang utama repositori Anda. Untuk instruksi, lihat CodeBuild dokumentasi.	Pengembang aplikasi, administrator AWS, AWS DevOps

Sumber daya terkait

- Contoh aplikasi uji unit game (GitHub repositori dengan kode sampel)
- CodeBuild Dokumentasi AWS
- GitHub acara webhook (CodeBuild dokumentasi)
- Membuat repositori baru (dokumentasi) GitHub

Informasi tambahan

Hasil tes unit

Di CodeBuild konsol, Anda akan melihat hasil pengujian berikut setelah proyek berhasil dibangun.

Contoh komponen uji unit

Bagian ini menjelaskan empat jenis komponen uji yang digunakan dalam pengujian unit: pernyataan, mata-mata, rintisan, dan tiruan. Ini mencakup penjelasan singkat dan contoh kode dari setiap komponen.

Pernyataan

Pernyataan digunakan untuk memverifikasi hasil yang diharapkan. Ini adalah komponen pengujian yang penting karena memvalidasi respons yang diharapkan dari fungsi yang diberikan. Contoh pernyataan berikut memvalidasi bahwa ID yang dikembalikan adalah antara 0 dan 1000 saat menginisialisasi game baru.

```
const { expect }  = require('chai');
const { Game } = require('../src/index');

describe('Game Function Group', () => {
 it('Check that the Game ID is between 0 and 1000', function() {
      const game = new Game();
      expect(game.id).is.above(0).but.below(1000)
 });
});
```

Mata-mata

Mata-mata digunakan untuk mengamati apa yang terjadi ketika suatu fungsi sedang berjalan. Misalnya, Anda mungkin ingin memvalidasi bahwa fungsi telah dipanggil dengan benar. Contoh berikut menunjukkan bahwa metode start dan stop dipanggil pada objek kelas Game.

```
const { expect }  = require('chai');
const { spy } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () => {
   it('should verify that the correct function is called', () => {
      const spyStart = spy(Game.prototype, "start");
      const spyStop = spy(Game.prototype, "stop");

      const game = new Game();
      game.start();
      game.stop();

      expect(spyStart.called).to.be.true
      expect(spyStop.called).to.be.true
    });
});
```

Rintisan

Sebuah rintisan digunakan untuk mengganti respons default fungsi. Ini sangat berguna ketika fungsi membuat permintaan eksternal, karena Anda ingin menghindari membuat permintaan eksternal dari pengujian unit. (Permintaan eksternal lebih cocok untuk pengujian integrasi, yang secara fisik dapat menguji permintaan antara komponen yang berbeda.) Dalam contoh berikut, sebuah rintisan memaksa ID kembali dari fungsi getID.

```
const { expect }  = require('chai');
const {.stub } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () =>  {
   it('Check that the Game ID is between 0 and 1000', function() {
      let generateIdStub = stub(Game.prototype, 'getId').returns(999999);

      const game = new Game();
```

```
      expect(game.getId).is.equal(999999);

      generateIdStub.restore();
    });
});
```

Mengolok-olok

Mock adalah metode palsu yang memiliki perilaku yang telah diprogram sebelumnya untuk menguji skenario yang berbeda. Mock dapat dianggap sebagai bentuk rintisan yang diperluas dan dapat melakukan banyak tugas secara bersamaan. Dalam contoh berikut, tiruan digunakan untuk memvalidasi tiga skenario:

- Fungsi ini disebut
- Fungsi disebut dengan argumen
- Fungsi mengembalikan bilangan bulat 9

```
const { expect }  = require('chai');
const {.mock } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () =>  {
   it('Check that the Game ID is between 0 and 1000', function() {
      let mock = mock(Game.prototype).expects('getId').withArgs().returns(9);

      const game = new Game();
      const id = get.getId();

      mock.verify();
      expect(id).is.equal(9);
    });
});
```

Struktur proyek Python dalam arsitektur heksagonal menggunakan AWS Lambda

Furkan Oruc, Dominik Goby, Darius Kunce, dan Michal Ploski, Amazon Web Services

Ringkasan

Pola ini menunjukkan bagaimana menyusun proyek Python dalam arsitektur heksagonal dengan menggunakan AWS Lambda. Pola ini menggunakan AWS Cloud Development Kit (AWS CDK) sebagai alat infrastruktur sebagai kode (IAc), Amazon API Gateway sebagai REST API, dan Amazon DynamoDB sebagai lapisan persistensi. Arsitektur heksagonal mengikuti prinsip-prinsip desain berbasis domain. Dalam arsitektur heksagonal, perangkat lunak terdiri dari tiga komponen: domain, port, dan adaptor. Untuk informasi rinci tentang arsitektur heksagonal dan manfaatnya, lihat panduan Membangun arsitektur heksagonal di AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Pengalaman dalam Python
- Keakraban dengan AWS Lambda, AWS CDK, Amazon API Gateway, dan DynamoDB
- GitHub Akun (lihat instruksi untuk mendaftar)
- Git (lihat petunjuk instalasi)
- Editor kode untuk membuat perubahan dan mendorong kode Anda ke GitHub (misalnya, Visual Studio Code atau JetBrains PyCharm)
- Docker diinstal, dan daemon Docker aktif dan berjalan

Versi produk

- Git versi 2.24.3 atau yang lebih baru
- Python versi 3.7 atau yang lebih baru
- AWS CDK v2
- Puisi versi 1.1.13 atau yang lebih baru
- AWS Lambda Powertools untuk Python versi 1.25.6 atau yang lebih baru
- pytest versi 7.1.1 atau yang lebih baru

- Moto versi 3.1.9 atau yang lebih baru
- pydantic versi 1.9.0 atau yang lebih baru
- Boto3 versi 1.22.4 atau yang lebih baru
- mypy-boto3-dynamodb versi 1.24.0 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

Tumpukan teknologi target terdiri dari layanan Python yang menggunakan API Gateway, Lambda, dan DynamoDB. Layanan ini menggunakan adaptor DynamoDB untuk mempertahankan data. Ini menyediakan fungsi yang menggunakan Lambda sebagai titik masuk. Layanan ini menggunakan Amazon API Gateway untuk mengekspos REST API. API menggunakan AWS Identity and Access Management (IAM) untuk otentikasi klien.

Arsitektur target

Untuk mengilustrasikan implementasi, pola ini menerapkan arsitektur target tanpa server. Klien dapat mengirim permintaan ke titik akhir API Gateway. API Gateway meneruskan permintaan ke fungsi Lambda target yang mengimplementasikan pola arsitektur heksagonal. Fungsi Lambda melakukan operasi create, read, update, and delete (CRUD) pada tabel DynamoDB.

> **Important**
>
> Pola ini diuji di lingkungan PoC. Anda harus melakukan tinjauan keamanan untuk mengidentifikasi model ancaman dan membuat basis kode yang aman sebelum Anda menerapkan arsitektur apa pun ke lingkungan produksi.

API mendukung lima operasi pada entitas produk:

- `GET /products` mengembalikan semua produk.
- `POST /products` menciptakan produk baru.
- `GET /products/{id}` mengembalikan produk tertentu.

- `PUT /products/{id}`memperbarui produk tertentu.
- `DELETE /products/{id}`menghapus produk tertentu.

Anda dapat menggunakan struktur folder berikut untuk mengatur proyek Anda untuk mengikuti pola arsitektur heksagonal:

```
app/  # application code
|--- adapters/  # implementation of the ports defined in the domain
     |--- tests/  # adapter unit tests
|--- entrypoints/  # primary adapters, entry points
     |--- api/  # api entry point
          |--- model/  # api model
          |--- tests/  # end to end api tests
|--- domain/  # domain to implement business logic using hexagonal architecture
     |--- command_handlers/  # handlers used to execute commands on the domain
     |--- commands/  # commands on the domain
     |--- events/  # events triggered via the domain
     |--- exceptions/  # exceptions defined on the domain
     |--- model/  # domain model
     |--- ports/  # abstractions used for external communication
     |--- tests/  # domain tests
|--- libraries/  # List of 3rd party libraries used by the Lambda function
infra/  # infrastructure code
simple-crud-app.py  # AWS CDK v2 app
```

Alat

Layanan AWS

- Amazon API Gateway adalah layanan yang dikelola sepenuhnya yang memudahkan pengembang untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan APIs pada skala apa pun.
- Amazon DynamoDB adalah database NoSQL bernilai kunci yang dikelola sepenuhnya, tanpa server, yang dirancang untuk menjalankan aplikasi berkinerja tinggi pada skala apa pun.
- AWS Lambda adalah layanan komputasi tanpa server dan berbasis peristiwa yang memungkinkan Anda menjalankan kode untuk hampir semua jenis aplikasi atau layanan backend tanpa menyediakan atau mengelola server. Anda dapat meluncurkan fungsi Lambda dari lebih dari 200 layanan AWS dan aplikasi perangkat lunak sebagai layanan (SaaS), dan hanya membayar untuk apa yang Anda gunakan.

Alat

- Git digunakan sebagai sistem kontrol versi untuk pengembangan kode dalam pola ini.
- Python digunakan sebagai bahasa pemrograman untuk pola ini. Python menyediakan struktur data tingkat tinggi dan pendekatan untuk pemrograman berorientasi objek. AWS Lambda menyediakan runtime Python bawaan yang menyederhanakan pengoperasian layanan Python.
- Visual Studio Code digunakan sebagai IDE untuk pengembangan dan pengujian untuk pola ini. Anda dapat menggunakan IDE apa pun yang mendukung pengembangan Python (misalnya, PyCharm).
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak open-source yang memungkinkan Anda menentukan sumber daya aplikasi cloud Anda dengan menggunakan bahasa pemrograman yang sudah dikenal. Pola ini menggunakan CDK untuk menulis dan menyebarkan infrastruktur cloud sebagai kode.
- Puisi digunakan untuk mengelola dependensi dalam pola.
- Docker digunakan oleh AWS CDK untuk membangun paket dan lapisan Lambda.

Kode

Kode untuk pola ini tersedia di repositori sampel arsitektur heksagonal GitHub Lambda.

Praktik terbaik

Untuk menggunakan pola ini dalam lingkungan produksi, ikuti praktik terbaik berikut:

- Gunakan kunci terkelola pelanggan di AWS Key Management Service (AWS KMS) untuk mengenkripsi grup log Amazon dan tabel CloudWatch Amazon DynamoDB.
- Konfigurasikan AWS WAF untuk Amazon API Gateway agar hanya mengizinkan akses dari jaringan organisasi Anda.
- Pertimbangkan opsi lain untuk otorisasi API Gateway jika IAM tidak memenuhi kebutuhan Anda. Misalnya, Anda dapat menggunakan kumpulan pengguna Amazon Cognito atau otorisasi API Gateway Lambda.
- Gunakan backup DynamoDB.
- Konfigurasikan fungsi Lambda dengan penyebaran virtual private cloud (VPC) untuk menjaga lalu lintas jaringan di dalam cloud.
- Perbarui konfigurasi asal yang diizinkan untuk preflight cross-origin resource sharing (CORS) untuk membatasi akses ke domain asal yang meminta saja.

- Gunakan cdk-nag untuk memeriksa kode AWS CDK untuk praktik terbaik keamanan.
- Pertimbangkan untuk menggunakan alat pemindaian kode untuk menemukan masalah keamanan umum dalam kode. Misalnya, Bandit adalah alat yang dirancang untuk menemukan masalah keamanan umum dalam kode Python. Pip-Audit memindai lingkungan Python untuk paket yang memiliki kerentanan yang diketahui.

Pola ini menggunakan AWS X-Ray untuk melacak permintaan melalui titik masuk, domain, dan adaptor aplikasi. AWS X-Ray membantu pengembang mengidentifikasi kemacetan dan menentukan latensi tinggi untuk meningkatkan kinerja aplikasi.

Epik

Inisialisasi proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Anda sendiri.	1. Masuk ke GitHub. 2. Buat repositori baru. Untuk instruksi, lihat GitHub dokumentasi. 3. Kloning dan dorong repositori sampel untuk pola ini ke dalam repositori baru di akun Anda.	Pengembang aplikasi
Instal dependensi.	1. Instal Puisi. `pip install poetry` 2. Instal paket dari direktori root. Perintah berikut menginstal aplikasi dan paket AWS CDK. Ini juga menginstal paket pengembangan yang diperlukan untuk menjalankan pengujian unit. Semua	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	paket yang diinstal ditempatkan di lingkungan virtual baru.
	`poetry install`
	3. Untuk melihat represent asi grafis dari paket yang diinstal, jalankan perintah berikut.
	`poetry show --tree`
	4. Perbarui semua dependens i.
	`poetry update`
	5. Buka shell baru dalam lingkungan virtual yang baru dibuat. Ini berisi semua dependensi yang diinstal.
	`poetry shell`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan IDE Anda.	Kami merekomendasikan Visual Studio Code, tetapi Anda dapat menggunakan IDE pilihan Anda yang mendukung Python. Langkah-langkah berikut adalah untuk Visual Studio Code. 1. Perbarui `.vscode/settings` file. `{ "python.testing.pytestArgs": [ "app/adapters/tests", "app/entrypoints/api/tests", "app/domain/tests" ], "python.testing.unittestEnabled": false, "python.testing.pytestEnabled": true, "python.envFile": "${workspaceFolder}/.env", }` 2. Buat `.env` file di direktori root proyek. Ini memastikan bahwa direktori root proyek disertakan dalam `PYTHONPATH` sehingga `pytest` dapat menemukan	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nya dan menemukan semua paket dengan benar. `PYTHONPATH=.`	
Jalankan pengujian unit, opsi 1: Gunakan Kode Visual Studio.	1. Pilih penerjemah Python dari lingkungan virtual yang dikelola oleh Puisi. 2. Jalankan tes dari Test Explorer.	Pengembang aplikasi

Jalankan pengujian unit, opsi 2: Gunakan perintah shell.	1. Mulai shell baru dalam lingkungan virtual.
	`poetry shell`
	2. Jalankan `pytest` perintah dari direktori root.
	`python -m pytest`
	Atau Anda dapat menjalankan perintah langsung dari Puisi.
	`poetry run python -m pytest`
		Pengembang aplikasi

Menyebarkan dan menguji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Minta kredensi sementara.	Untuk memiliki kredensi AWS di shell saat Anda menjalankan`cdk deploy`,	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	buat kredensil sementara dengan menggunakan AWS IAM Identity Center (penerus AWS Single Sign-On). Untuk petunjuknya, lihat posting blog Cara mengambil kredensi jangka pendek untuk penggunaan CLI dengan AWS IAM Identity Center.	

Men-deploy aplikasi.	1. Instal AWS CDK v2.
	`npm install -g aws-cdk`
	Untuk informasi selengkap nya, lihat dokumentasi AWS CDK.
	2. Bootstrap AWS CDK ke akun dan Wilayah Anda.
	`cdk bootstrap aws://12345678900/ us-east-1 --profile aws-profile-name`
	3. Terapkan aplikasi sebagai CloudFormation tumpukan AWS dengan menggunakan profil AWS.
	`cdk deploy --profile aws-profile-name`
		Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji API, opsi 1: Gunakan konsol.	Gunakan konsol API Gateway untuk menguji API. Untuk informasi selengkapnya tentang operasi dan request/response pesan API, lihat bagian penggunaan API dari file readme di GitHub repositori.	Pengembang aplikasi, AWS DevOps
Uji API, opsi 2: Gunakan Tukang Pos.	Jika Anda ingin menggunakan alat seperti Tukang Pos: 1. Instal Postman sebagai aplikasi mandiri atau ekstensi browser. 2. Salin URL titik akhir untuk API Gateway. Ini akan dalam format berikut. `https://{api-id}.execute-api.{region}.amazonaws.com/{stage}/{path}` 3. Konfigurasikan tanda tangan AWS di tab otorisasi. Untuk petunjuknya, lihat artikel AWS re:Post tentang mengaktifkan autentikasi IAM untuk API Gateway REST. APIs 4. Gunakan Postman untuk mengirim permintaan ke titik akhir API Anda.	Pengembang aplikasi, AWS DevOps

Kembangkan layanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tulis tes unit untuk domain bisnis.	1. Buat file Python di `app/domain/tests` folder dengan menggunakan awalan nama `test_` file.
	2. Buat metode pengujian baru untuk menguji logika bisnis baru dengan menggunakan contoh berikut.
	`def test_create_product_should_store_in_repository():`
	`  # Arrange`
	`  command = create_product_command.CreateProductCommand(`
	`  name="Test Product",`
	`  description="Test Description",`
	`  )`
	`  # Act`
	`  create_product_command_handler.handle_create_product_command(`
	`  command=command, unit_of_work=mock_unit_of_work`
	`  )`
	`  # Assert`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Buat kelas perintah di `app/domain/commands` folder. 4. Jika fungsionalitasnya baru, buat rintisan untuk pengendali perintah di folder. `app/domain/command_handlers` 5. Jalankan unit test untuk melihatnya gagal, karena masih belum ada logika bisnis. `python -m pytest`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menerapkan perintah dan penangan perintah.	1. Menerapkan logika bisnis dalam file handler perintah yang baru dibuat.
	2. Untuk setiap dependensi yang berinteraksi dengan sistem eksternal, deklarasi kan kelas abstrak di folder. `app/domain/ports`
	`class ProductsR epository(ABC):`
	`@abstractmethod`
	`def add(self, product: product.P roduct) -> None:`
	`...`
	`class UnitOfWor k(ABC):`
	`products: ProductsRepository`
	`@abstractmethod`
	`def commit(self) -> None:`
	`...`
	`@abstractmethod`
	`def __enter__ (self) -> typing.An y:`
	`...`
	`@abstractmethod`
	`def __exit__( self, *args) -> None:`
	`...`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Perbarui tanda tangan pengendali perintah untuk menerima dependensi yang baru dideklarasikan dengan menggunakan kelas port abstrak sebagai anotasi tipe.
	
	`def handle_create_product_command(`
	`  command: create_product_command.CreateProductCommand,`
	`  unit_of_work: unit_of_work.UnitOfWork,`
	`) -> str:`
	`  ...`
	
	4. Perbarui pengujian unit untuk mensimulasikan perilaku semua dependensi yang dideklarasikan untuk pengendali perintah.
	
	`  # Arrange`
	`  mock_unit_of_work = unittest.mock.create_autospec(`
	`  spec=unit_of_work.UnitOfWork, instance=True`
	`  )`
	`  mock_unit_of_work.products = unittest.mock.create_autospec(`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>spec=unit
	_of_work.ProductsR
	epository, instance=
	True
	)</pre>
	5. Perbarui logika pernyataan dalam pengujian untuk memeriksa pemanggilan ketergantungan yang diharapkan.
	<pre># Assert
	mock_unit
	_of_work.commit.as
	sert_called_once()
	product =
	mock_unit_of_work.
	products.add.call_
	args.args[0]
	
	assertpy.
	assert_that(produc
	t.name).is_equal_t
	o("Test Product")
	assertpy.
	assert_that(produc
	t.description).is_
	equal_to("Test
	Description")</pre>
	6. Jalankan unit test untuk melihatnya berhasil.
	<pre>python -m pytest</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tulis tes integrasi untuk adaptor sekunder.	1. Buat file uji di `app/adapters/tests` folder dengan menggunakan `test_` sebagai awalan nama file. 2. Gunakan pustaka Moto untuk mengejek layanan AWS. 3. Buat metode pengujian baru untuk uji integrasi adaptor.	Pengembang aplikasi

```
@pytest.fixture
def mock_dynamodb():
   with moto.mock
_dynamodb():
    yield boto3.res
ource("dynamodb",
 region_name="eu-ce
ntral-1")
```

```
def test_add_
and_commit_should_
store_product(mock
_dynamodb):
    # Arrange
    unit_of_work =
 dynamodb_unit_of_w
ork.DynamoDBUnitOf
Work(
        table_nam
e=TEST_TABLE_NAME,
 dynamodb_client=mo
ck_dynamodb.meta.c
lient
    )
    current_time =
 datetime.datetime.
```

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>now(datetime.timezone.utc).isoformat()
	
	    new_product_id = str(uuid.uuid4())
	    new_product = product.Product(
	        id=new_product_id,
	        name="test-name",
	        description="test-description",
	        createDate=current_time,
	        lastUpdateDate=current_time,
	    )
	
	    # Act
	    with unit_of_work:
	        unit_of_work.products.add(new_product)
	        unit_of_work.commit()
	
	    # Assert</pre> 4. Buat kelas adaptor di `app/adapters` folder. Gunakan kelas abstrak dari folder port sebagai kelas dasar.
	5. Jalankan unit test untuk melihatnya gagal, karena masih belum ada logika.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`python -m pytest`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menerapkan adaptor sekunder.	1. Menerapkan logika dalam file adaptor yang baru dibuat.
	2. Perbarui pernyataan uji.
	
	``# Assert
	    with unit_of_work_readonly:
	        product_from_db = unit_of_work_readonly.products.get(new_product_id)
	
	    assertpy.assert_that(product_from_db).is_not_none()
	    assertpy.assert_that(product_from_db.dict()).is_equal_to(
	        {
	            "id": new_product_id,
	            "name": "test-name",
	            "description": "test-description",
	            "createDate": current_time,
	            "lastUpdateDate": current_time,
	        }
	    )```
	
	3. Jalankan unit test untuk melihatnya berhasil.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`python -m pytest`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tulis end-to-end tes.	1. Buat file uji di `app/entry points/api/tests` folder dengan menggunak an `test_` sebagai awalan nama file. 2. Buat perlengkapan konteks Lambda yang akan digunakan oleh tes untuk memanggil Lambda. 3. Buat metode pengujian untuk pemanggilan API.	Pengembang aplikasi

```
@pytest.fixture
def lambda_co
ntext():
    @dataclass
    class LambdaCon
text:
        function_
name: str = "test"
        memory_li
mit_in_mb: int = 128
        invoked_f
unction_arn: str =
 "arn:aws:lambda:eu
-west-1:809313241:
function:test"
        aws_reque
st_id: str =
 "52fdfc07-2182-154
f-163f-5f0f9a621d7
2"

 return LambdaCon
text()
```

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`def test_crea te_product(lambda_ context):`
	`    # Arrange`
	`    name = "TestName"`
	`    description = "Test description"`
	`    request = api_model.CreatePr oductRequest(name= name, descripti on=description)`
	
	`    minimal_event = api_gateway_proxy_ event.APIGatewayPr oxyEvent(`
	`        {`
	`            "path": "/ products",`
	`            "httpMeth od": "POST",`
	`            "requestC ontext": {  # correlation ID`
	
	`  "requestId": "c6af9ac6-7b61-11e 6-9a41-93e8deadbee f"`
	`            },`
	`            "body": json.dumps(request .dict()),`
	`        }`
	`    )`
	
	`    create_pr oduct_func_mock = unittest.mock.crea te_autospec(`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>spec=crea
	te_product_command
	_handler.handle_cr
	eate_product_comma
	nd
	)
	handler.c
	reate_product_comm
	and_handler.handle
	_create_product_co
	mmand = (
	create_pr
	oduct_func_mock
	)
	
	# Act
	handler.h
	andler(minimal_eve
	nt, lambda_context)</pre> 4. Jalankan unit test untuk melihatnya gagal, karena masih belum ada logika.
	<pre>python -m pytest</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menerapkan adaptor utama.	1. Buat fungsi untuk logika bisnis API dan deklarasikan sebagai sumber daya API.
	
	`@tracer.capture_method`
	`@app.post("/products")`
	`@utils.parse_event(model=api_model.CreateProductRequest, app_context=app)`
	`def create_product(`
	`request: api_model.CreateProductRequest,`
	`) -> api_model.CreateProductResponse:`
	`"""Creates a product."""`
	`...`
	
	Note
	Semua dekorator yang Anda lihat adalah fitur dari AWS Lambda Powertools untuk pustaka Python. Untuk detailnya, lihat situs web AWS Lambda Powertools for Python.
	
	2. Menerapkan logika API.
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>id=create_product_
	command_handler.ha
	ndle_create_produc
	t_command(
	command=c
	reate_product_comm
	and.CreateProductC
	ommand(
	name=requ
	est.name,
	descripti
	on=request.descrip
	tion,
	),
	unit_of_w
	ork=unit_of_work,
	)
	response =
	api_model.CreatePr
	oductResponse(id=i
	d)
	return response.
	dict()</pre> 3. Jalankan unit test untuk melihatnya berhasil. <pre>python -m pytest</pre>	

Sumber daya terkait

Panduan APG

- Membangun arsitektur heksagonal di AWS

Referensi AWS

- Dokumentasi AWS Lambda

- Dokumentasi AWS CDK
  - Aplikasi AWS CDK pertama Anda
- Dokumentasi API Gateway
  - Kontrol akses ke API dengan izin IAM
  - Menggunakan konsol API Gateway untuk menguji metode REST API
- Dokumentasi Amazon DynamoDB

Alat

- Situs git-scm.com
- Instalasi Git
- Membuat GitHub repositori baru
- Situs web Python
- AWS Lambda Powertools untuk Python
- Situs Postman
- Pustaka objek tiruan Python
- Situs web puisi

IDEs

- Situs Visual Studio Code
- PyCharm situs web

Lebih banyak pola

- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS Fargate, PrivateLink AWS, dan Network Load Balancer
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS PrivateLink dan Network Load Balancer
- Kaitkan AWS CodeCommit repositori di satu dengan Akun AWS Amazon SageMaker AI Studio Classic di akun lain
- Mengotomatiskan penghapusan AWS CloudFormation tumpukan dan sumber daya terkait
- Membatasi akses berdasarkan alamat IP atau geolokasi dengan menggunakan AWS WAF
- Buat aplikasi seluler React Native tanpa server dengan menggunakan AWS Amplify
- Buat dan uji aplikasi iOS dengan AWS CodeCommit, AWS CodePipeline, dan AWS Device Farm
- Konfigurasikan logging untuk aplikasi.NET di Amazon CloudWatch Logs dengan menggunakan NLog
- Konsolidasikan pembuatan URL presigned Amazon S3 dan unduhan objek dengan menggunakan titik akhir yang terkait dengan alamat IP statis
- Buat definisi tugas Amazon ECS dan pasang sistem file pada EC2 instans menggunakan Amazon EFS
- Menerapkan aplikasi berbasis GRPC di cluster Amazon EKS dan mengaksesnya dengan Application Load Balancer
- Terapkan ChatOps solusi untuk mengelola hasil pemindaian SAST dengan menggunakan Amazon Q Developer dalam tindakan kustom aplikasi obrolan dan CloudFormation
- Menyebarkan kenari CloudWatch Synthetics dengan menggunakan Terraform
- Menerapkan layanan mikro Java di Amazon ECS menggunakan AWS Fargate
- Terapkan kasus penggunaan RAG AWS dengan menggunakan Terraform dan Amazon Bedrock
- Menerapkan sumber daya di AWS Wavelength Zona dengan menggunakan Terraform
- Menerapkan versi API berbasis jalur dengan menggunakan domain khusus di Amazon API Gateway
- Memigrasikan antrian pesan dari Microsoft Azure Service Bus ke Amazon SQS
- Migrasi aplikasi.NET dari Microsoft Azure App Service ke AWS Elastic Beanstalk
- Memigrasikan aplikasi web Go lokal ke AWS Elastic Beanstalk menggunakan metode biner
- Memigrasi server SFTP lokal untuk menggunakan AWS AWS Transfer for SFTP

- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2
- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2 dengan Auto Scaling
- Migrasikan aplikasi Java lokal ke AWS menggunakan AWS App2Container
- Migrasikan sertifikat SSL Windows ke Application Load Balancer menggunakan ACM
- Memodernisasi aplikasi Formulir Web ASP.NET di AWS
- Memantau aktivitas aplikasi dengan menggunakan Wawasan CloudWatch Log
- Jalankan wadah API Docker web ASP.NET Core pada instance Amazon Linux EC2
- Kirim atribut khusus ke Amazon Cognito dan masukkan ke dalam token
- Sajikan konten statis dalam bucket Amazon S3 melalui VPC dengan menggunakan Amazon CloudFront
- Siapkan PeopleSoft arsitektur yang sangat tersedia di AWS
- Merampingkan pengembangan dan penerapan bot Amazon Lex dengan menggunakan alur kerja otomatis
- Memecahkan masalah status AWS Step Functions dengan menggunakan Amazon Bedrock
- Gunakan Network Firewall untuk menangkap nama domain DNS dari Indikasi Nama Server untuk lalu lintas keluar
- Visualisasikan hasil AI/ML model menggunakan Flask dan AWS Elastic Beanstalk

IoT

Topik

- Ekstrak dan kueri atribut AWS IoT SiteWise metadata di danau data
- Mengatur dan memecahkan masalah AWS IoT Greengrass dengan perangkat klien
- Lebih banyak pola

Ekstrak dan kueri atribut AWS IoT SiteWise metadata di danau data

Ambarish Dongaonkar, Amazon Web Services

Ringkasan

AWS IoT SiteWise menggunakan model aset dan hierarki untuk mewakili peralatan, proses, dan fasilitas industri Anda. Setiap model atau aset dapat memiliki beberapa atribut yang spesifik untuk lingkungan Anda. Contoh atribut metadata mencakup lokasi atau lokasi fisik aset, detail pabrik, dan pengidentifikasi peralatan. Nilai atribut ini melengkapi data pengukuran aset untuk memaksimalkan nilai bisnis. Pembelajaran mesin (ML) dapat memberikan wawasan tambahan tentang metadata ini dan merampingkan tugas-tugas rekayasa.

Namun, atribut metadata tidak dapat ditanyakan langsung dari layanan. AWS IoT SiteWise Untuk membuat atribut dapat dikueri, Anda harus mengekstrak dan menelannya ke dalam danau data. Pola ini menggunakan skrip Python untuk mengekstrak atribut untuk semua AWS IoT SiteWise aset dan memasukkannya ke dalam data lake di bucket Amazon Simple Storage Service (Amazon S3). Setelah menyelesaikan proses ini, Anda dapat menggunakan kueri SQL di Amazon Athena untuk mengakses AWS IoT SiteWise atribut metadata dan kumpulan data lainnya, seperti kumpulan data pengukuran. Informasi atribut metadata juga berguna saat bekerja dengan AWS IoT SiteWise monitor atau dasbor. Anda juga dapat membuat dasbor Amazon Quick Sight dengan menggunakan atribut yang diekstrak di bucket Amazon S3.

Pola memiliki kode referensi, dan Anda dapat menerapkan kode dengan menggunakan layanan komputasi terbaik untuk kasus penggunaan Anda, seperti AWS Lambda atau AWS Glue.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Izin untuk mengatur AWS Lambda fungsi atau AWS Glue pekerjaan.
- Bucket Amazon S3.
- Model aset dan hierarki diatur dalam AWS IoT SiteWise. Untuk informasi selengkapnya, lihat Membuat model aset dalam AWS IoT SiteWise dokumentasi.

Arsitektur

Anda dapat menggunakan fungsi Lambda atau AWS Glue pekerjaan untuk menyelesaikan proses ini. Kami merekomendasikan menggunakan Lambda jika Anda memiliki kurang dari 100 model dan setiap model memiliki rata-rata 15 atau lebih sedikit atribut. Untuk semua kasus penggunaan lainnya, kami sarankan untuk menggunakan AWS Glue.

Arsitektur solusi dan alur kerja ditunjukkan pada diagram berikut.

1. AWS Glue Pekerjaan terjadwal atau fungsi Lambda berjalan. Ini mengekstrak atribut metadata aset dari AWS IoT SiteWise dan mencernanya ke dalam bucket Amazon S3.
2. AWS Glue Crawler merayapi data yang diekstraksi di bucket Amazon S3 dan membuat tabel dalam file. AWS Glue Data Catalog
3. Menggunakan SQL standar, Amazon Athena menanyakan tabel di file. AWS Glue Data Catalog

Otomatisasi dan skala

Anda dapat menjadwalkan fungsi atau AWS Glue pekerjaan Lambda untuk dijalankan setiap hari atau mingguan, sesuai dengan frekuensi pembaruan konfigurasi AWS IoT SiteWise aset Anda.

Tidak ada batasan jumlah AWS IoT SiteWise aset yang dapat diproses oleh kode sampel, tetapi sejumlah besar aset dapat meningkatkan jumlah waktu yang dibutuhkan untuk menyelesaikan proses.

Alat

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar.
- AWS Glueadalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS IoT SiteWisemembantu Anda mengumpulkan, memodelkan, menganalisis, dan memvisualisasikan data dari peralatan industri dalam skala besar.

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS

Epik

Mengatur pekerjaan atau fungsi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan izin di IAM.	Di konsol IAM, berikan izin ke peran IAM yang diasumsikan oleh fungsi atau AWS Glue tugas Lambda untuk melakukan hal berikut: • Baca dari AWS IoT SiteWise layanan • Menulis ke ember Amazon S3 Untuk informasi selengkapnya, lihat Membuat peran Layanan AWS untuk dokumentasi IAM.	AWS Umum
Buat fungsi atau AWS Glue pekerjaan Lambda.	Jika Anda menggunakan Lambda, buat fungsi Lambda baru. Untuk Runtime, pilih Python. Untuk informasi selengkapnya, lihat Membangun fungsi Lambda	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan Python di dokumentasi Lambda. Jika Anda menggunakan AWS Glue, buat pekerjaan shell Python baru di konsol. AWS Glue Untuk informasi selengkapnya, lihat Menambahkan pekerjaan shell Python di dokumentasi. AWS Glue	
Perbarui fungsi atau AWS Glue pekerjaan Lambda.	Ubah fungsi atau AWS Glue pekerjaan Lambda baru, dan masukkan contoh kode di bagian Informasi tambahan. Ubah kode sesuai kebutuhan untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Mengedit kode menggunakan editor konsol di dokumentasi Lambda dan lihat Bekerja dengan skrip dalam dokumentasi.AWS Glue	AWS Umum

Jalankan pekerjaan atau fungsi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan fungsi atau AWS Glue pekerjaan Lambda.	Jalankan fungsi atau AWS Glue pekerjaan Lambda. Untuk informasi selengkapnya, lihat Memanggil fungsi Lambda di dokumentasi	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Lambda atau lihat Memulai pekerjaan menggunakan pemicu dalam dokumenta si. AWS Glue Ini mengekstr ak atribut metadata untuk aset dan model dalam AWS IoT SiteWise hierarki dan menyimpannya di bucket Amazon S3 yang ditentukan.	
Siapkan AWS Glue crawler.	Siapkan AWS Glue crawler dengan pengklasifikasi format yang diperlukan untuk file berformat CSV. Gunakan bucket Amazon S3 dan detail awalan yang digunakan dalam fungsi atau pekerjaan Lambda. AWS Glue Untuk informasi selengkapnya, lihat Mendefinisikan crawler dalam dokumentasi. AWS Glue	AWS Umum
Jalankan AWS Glue crawler.	Jalankan crawler untuk memproses file data yang dibuat oleh fungsi AWS Glue atau pekerjaan Lambda. Crawler membuat tabel dalam yang ditentukan AWS Glue Data Catalog. Untuk informasi selengkapnya, lihat atau Memulai crawler menggunak an pemicu dalam dokumenta si. AWS Glue	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kueri atribut metadata.	Menggunakan Amazon Athena, gunakan SQL standar untuk menanyakan sesuai AWS Glue Data Catalog kebutuhan untuk kasus penggunaan Anda. Anda dapat bergabung dengan tabel atribut metadata dengan database dan tabel lainnya. Untuk informasi selengkapnya, lihat Memulai di dokumentasi Amazon Athena.	AWS Umum

Sumber daya terkait

- Dokumentasi Amazon Athena
- AWS Glue dokumentasi
- AWS IoT SiteWise Referensi API
- AWS IoT SiteWise panduan pengguna
  - Mulai menggunakan
  - Pemodelan aset industri
  - Mendefinisikan hubungan antara model aset (hierarki)
  - Mengasosiasikan dan memisahkan aset
  - Membuat AWS IoT SiteWise demo
- IOTSiteWise (SDK untuk dokumentasi Python)
- Dokumentasi Lambda

Informasi tambahan

Kode

Kode sampel yang disediakan adalah untuk referensi, dan Anda dapat menyesuaikan kode ini sesuai kebutuhan untuk kasus penggunaan Anda.

```
# Following code can be used in an AWS Lambda function or in an AWS Glue Python shell
 job.
# IAM roles used for this job need read access to the AWS IoT SiteWise service and
 write access to the S3 bucket.
sw_client = boto3.client('iotsitewise')
s3_client = boto3.client('s3')
output = io.StringIO()

attribute_list=[]
bucket = '{3_bucket name}'
prefix = '{s3_bucket prefix}'
output.write("model_id,model_name,asset_id,asset_name,attribuet_id,attribute_name,attribute_val
\n")

m_resp = sw_client.list_asset_models()
for m_rec in m_resp['assetModelSummaries']:
     model_id = m_rec['id']
     model_name = m_rec['name']

     attribute_list.clear()
     dam_response = sw_client.describe_asset_model(assetModelId=model_id)
     for rec in dam_response['assetModelProperties']:
         if 'attribute' in rec['type']:
            attribute_list.append(rec['name'])

     response = sw_client.list_assets(assetModelId=model_id, filter='ALL')
     for asset in response['assetSummaries']:
         asset_id = asset['id']
         asset_name = asset['name']
         resp = sw_client.describe_asset(assetId=asset_id)
         for rec in resp['assetProperties']:
            if rec['name'] in attribute_list:
                p_resp = sw_client.get_asset_property_value(assetId=asset_id,
 propertyId=rec['id'])
                if 'propertyValue' in p_resp:
                    if p_resp['propertyValue']['value']:
                        if 'stringValue' in p_resp['propertyValue']['value']:
                             output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
```

```
 str(p_resp['propertyValue']['value']['stringValue']) + "\n")

                        if 'doubleValue' in p_resp['propertyValue']['value']:
                             output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
 str(p_resp['propertyValue']['value']['doubleValue']) + "\n")
                        if 'integerValue' in p_resp['propertyValue']['value']:
                             output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
 str(p_resp['propertyValue']['value']['integerValue']) + "\n")
                         if 'booleanValue' in p_resp['propertyValue']['value']:
                             output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
 str(p_resp['propertyValue']['value']['booleanValue']) + "\n")

output.seek(0)
s3_client.put_object(Bucket=bucket, Key= prefix + '/data.csv', Body=output.getvalue())
output.close()
```

Mengatur dan memecahkan masalah AWS IoT Greengrass dengan perangkat klien

Marouane Sefiani dan Akalanka De Silva, Amazon Web Services

Ringkasan

AWS IoT Greengrass adalah runtime edge open-source dan layanan cloud untuk membangun, menerapkan, dan mengelola perangkat lunak Internet of Things (IoT) pada perangkat edge. Kasus penggunaan untuk AWS IoT Greengrass meliputi:

- Rumah pintar tempat gateway AWS IoT Greengrass digunakan sebagai hub untuk otomatisasi rumah
- Pabrik pintar tempat AWS IoT Greengrass dapat memfasilitasi konsumsi dan pemrosesan data lokal dari lantai toko

AWS IoT Greengrass dapat bertindak sebagai titik akhir koneksi MQTT yang aman dan terautentikasi untuk perangkat edge lainnya (juga dikenal sebagai perangkat klien), yang biasanya akan terhubung langsung ke AWS IoT Core. Kemampuan ini berguna ketika perangkat klien tidak memiliki akses jaringan langsung ke titik akhir AWS IoT Core.

Anda dapat mengatur AWS IoT Greengrass untuk digunakan dengan perangkat klien untuk kasus penggunaan berikut:

- Untuk perangkat klien untuk mengirim data ke AWS IoT Greengrass
- Untuk AWS IoT Greengrass untuk meneruskan data ke AWS IoT Core
- Untuk memanfaatkan fitur engine aturan AWS IoT Core yang canggih

Kemampuan ini memerlukan penginstalan dan konfigurasi komponen berikut pada perangkat AWS IoT Greengrass:

- Pialang MQTT
- Jembatan MQTT
- Otentikasi perangkat klien
- Detektor IP

Selain itu, pesan yang dipublikasikan dari perangkat klien harus dalam format JSON atau format Protocol Buffers (protobuf).

Pola ini menjelaskan cara menginstal dan mengkonfigurasi komponen yang diperlukan ini, dan memberikan tips pemecahan masalah dan praktik terbaik.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS Command Line Interface (AWS CLI) versi 2
- Dua perangkat klien yang menjalankan Python 3.7 atau yang lebih baru
- Satu perangkat inti yang menjalankan Java Runtime Environment (JRE) versi 8 atau yang lebih baru, dan Amazon Corretto 11 atau OpenJDK 11

Batasan

- Anda harus memilih Wilayah AWS tempat AWS IoT Core tersedia. Untuk daftar Wilayah AWS IoT Core saat ini, lihat AWS Services by Region.
- Perangkat inti harus memiliki setidaknya 172 MB RAM dan 512 MB ruang disk.

Arsitektur

Diagram berikut menunjukkan arsitektur solusi untuk pola ini.

Arsitekturnya meliputi:

- Dua perangkat klien. Setiap perangkat berisi kunci pribadi, sertifikat perangkat, dan sertifikat otoritas sertifikat root (CA). AWS IoT Device SDK, yang berisi klien MQTT, juga diinstal pada setiap perangkat klien.
- Perangkat inti yang memiliki AWS IoT Greengrass yang digunakan dengan komponen berikut:
  - Pialang MQTT
  - Jembatan MQTT
  - Otentikasi perangkat klien
  - Detektor IP

Arsitektur ini mendukung skenario berikut:

- Perangkat klien dapat menggunakan klien MQTT mereka untuk berkomunikasi satu sama lain melalui broker MQTT perangkat inti.
- Perangkat klien juga dapat berkomunikasi dengan AWS IoT Core di cloud melalui broker MQTT perangkat inti dan jembatan MQTT.
- AWS IoT Core di cloud dapat mengirim pesan ke perangkat klien melalui klien pengujian MQTT dan jembatan MQTT perangkat inti dan broker MQTT.

Untuk informasi selengkapnya tentang komunikasi antara perangkat klien dan perangkat inti, lihat bagian Informasi tambahan.

Alat

Layanan AWS

- AWS IoT Greengrass adalah layanan runtime dan cloud edge Internet of Things (IoT) open source yang membantu Anda membangun, menerapkan, dan mengelola aplikasi IoT di perangkat Anda.
- AWS IoT Core menyediakan komunikasi dua arah yang aman untuk perangkat yang terhubung ke internet agar tersambung ke AWS Cloud.
- AWS IoT Device SDK adalah kit pengembangan perangkat lunak yang mencakup pustaka sumber terbuka, panduan pengembang dengan sampel, dan panduan porting sehingga Anda dapat membuat produk atau solusi IoT inovatif pada platform perangkat keras pilihan Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Praktik terbaik

- Muatan pesan dari perangkat klien harus dalam format JSON atau Protobuf untuk memanfaatkan fitur-fitur canggih dari mesin aturan AWS IoT Core, seperti transformasi dan tindakan bersyarat.
- Konfigurasikan jembatan MQTT untuk memungkinkan komunikasi dua arah.
- Konfigurasikan dan terapkan komponen detektor IP di AWS IoT Greengrass untuk memastikan bahwa alamat IP perangkat inti disertakan dalam bidang nama alternatif subjek (SAN) sertifikat broker MQTT.

Epik

Siapkan perangkat inti

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS IoT Greengrass di perangkat inti Anda.	Instal perangkat lunak AWS IoT Greengrass Core dengan mengikuti petunjuk di panduan pengembang.	AWS IoT Greengrass

Periksa status instalasi Anda.	Gunakan perintah berikut untuk memeriksa status layanan AWS IoT Greengrass di perangkat inti Anda:
	`sudo systemctl status greengrass.service`
	Output yang diharapkan dari perintah ini adalah:
	`Launched Nucleus successfully`
		AWS Umum

Siapkan kebijakan IAM dan lampirkan ke peran layanan Greengrass.	1. Buat kebijakan IAM untuk memungkinkan komunikasi ke dan dari jembatan MQTT. Berikut ini contoh kebijakan: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow",`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Action":
	[
	
	"iot:*"
	],
	"Resource
	": "*"
	},
	{
	"Sid":
	"GreengrassActions
	",
	"Effect":
	"Allow",
	"Action":
	[
	
	"greengrass:*"
	],
	"Resource
	": "*"
	}
	]
	}</pre>	

	2. Lampirkan kebijakan ke peran layanan Greengrass. Untuk mendapatkan peran layanan, gunakan perintah:
	
	<pre>aws greengrassv2
	get-service-role-f
	or-account --region
	<region></pre>
	dimana `<region>` mengacu pada Wilayah AWS Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan terapkan komponen yang diperluka n di perangkat inti AWS IoT Greengrass.	Konfigurasikan dan terapkan komponen-komponen berikut: • `greengrass.clientd evices.mqtt.Moquet te` (lihat detail konfigurasi) • `greengrass.clientd evices.mqtt.Bridge` (lihat detail konfigurasi dan tugas berikutnya) • `greengrass.clientd evices.Auth` (lihat detail konfigurasi dan tugas setelah yang berikutnya) • `aws.greengrass.cli entdevices.IPDetec tor` (lihat detail konfigurasi)	AWS IoT Greengrass

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan bahwa jembatan MQTT memungkinkan komunikasi dua arah.	Untuk menyampaikan pesan MQTT antara perangkat klien dan AWS IoT Core, konfigurasikan dan terapkan komponen jembatan MQTT dan tentukan topik yang akan disampaikan. Berikut ini contohnya: `{ "mqttTopicMapping": { "ClientDevicesToCloud": { "topic": "dt/#", "source": "LocalMqtt", "target": "IotCore" }, "CloudToClientDevices": { "topic": "cmd/#", "source": "IotCore", "target": "LocalMqtt" } } }`	AWS IoT Greengrass

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan bahwa komponen autentikasi memungkinkan perangkat klien untuk terhubung dan menerbitkan atau berlangga nan topik.	`aws.greengrass.cli entdevice s.Auth` Konfigurasi berikut memungkinkan semua perangkat klien untuk terhubung, mempublikasikan pesan, dan berlangganan semua topik. `{ "deviceGroups": { "formatVersion": "2021-03-05", "definitions": { "MyPermis siveDeviceGroup": { "selectio nRule": "thingName: *", "policyName": "MyPermissivePolicy" } }, "policies": { "MyPermis sivePolicy": { "AllowAll": { "statemen tDescription": "Allow client devices to perform all actions.", "operations": ["*"], "resources": ["*"] }`	AWS IoT Greengrass

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> }
	    }
	  }
	}</pre>	

Siapkan perangkat klien

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal AWS IoT Device SDK.	Instal AWS IoT Device SDK pada perangkat klien. Untuk daftar lengkap bahasa yang didukung dan yang terkait SDKs, lihat dokumentasi AWS IoT Core.
	
	Misalnya, AWS IoT Device SDK untuk Python SDK terletak di. GitHub Untuk menginstal SDK ini:
	
	1. Konfirmasikan bahwa Python 3.7 atau yang lebih baru diinstal, seperti yang diinstruksikan pada halaman Prasyarat repositori. GitHub
	2. Gunakan perintah pip untuk menginstal SDK.
	
	Untuk macOS dan Linux:
	
	<pre>python3 -m pip install
	awsiotsdk</pre>
		AWS IoT Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Untuk Windows:
	`python -m pip install awsiotsdk`
	Atau, Anda dapat menginstal SDK dari repositori sumber:
	`# Create a workspace directory to hold all the SDK files`
	`mkdir sdk-workspace`
	`cd sdk-workspace`
	`# Clone the repository`
	`git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git`
	`# Install using Pip (use 'python' instead of 'python3' on Windows)`
	`python3 -m pip install ./aws-iot-device-sdk-python-v2`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ciptakan sesuatu.	1. Di konsol AWS IoT, jika tombol Mulai muncul, pilih tombol tersebut. Jika tidak, di panel navigasi, pilih Keamanan, Kebijakan. 2. Jika kotak dialog Anda belum memiliki kebijakan apa pun muncul, pilih Buat kebijakan. Jika tidak, pilih Buat. 3. Masukkan nama untuk kebijakan AWS IoT (misalnya,`ClientDevicePolicy`). 4. Di bagian Tambah pernyataan, ganti kebijakan yang ada dengan kode JSON berikut. Ganti `<region>` dan `<account>` dengan AWS Region dan nomor akun AWS Anda. `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "iot:Connect", "Resource": "arn:aws:iot:region:account:client/*" }, {`	AWS IoT Core

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`"Effect": "Allow", "Action": "iot:Publish", "Resource": "*" }, { "Effect": "Allow", "Action": "iot:Receive", "Resource": "*" }, { "Effect": "Allow", "Action": "iot:Subscribe", "Resource": "*" }, { "Effect": "Allow", "Action": ["iot:GetThingShadow", "iot:UpdateThingShadow", "iot:DeleteThingShadow"], "Resource": "arn:aws:iot:region:account:thing/*" }] }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Pilih Buat. 6. Di konsol AWS IoT, di panel navigasi, pilih Kelola, Hal. 7. Jika kotak dialog Anda belum memiliki hal apa pun ditampilkan, pilih Daftarkan sesuatu. Jika tidak, pilih Buat. 8. Pada halaman Creating AWS IoT things, pilih Create a single. 9. Pada halaman Tambahkan perangkat Anda ke registri perangkat, masukkan nama untuk hal IoT Anda (misalnya, **`ClientDevice1`**), lalu pilih Berikutnya. **Note** Anda tidak dapat mengubah nama sesuatu setelah Anda membuatnya. Untuk mengubah nama, Anda harus membuat hal baru, memberinya nama baru, dan kemudian menghapus yang lama.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	10. Pada halaman Tambahkan sertifikat untuk hal Anda, pilih Buat sertifikat. 11. Pilih tautan Unduh untuk mengunduh sertifikat, kunci pribadi, dan sertifikat root CA. **Important** Ini adalah satu-satunya kesempatan Anda untuk mengunduh sertifikat dan kunci pribadi Anda. 12. Untuk mengaktifkan sertifikat, pilih Aktifkan. Sertifikat harus aktif agar perangkat dapat terhubung ke AWS IoT. 13. Pilih Lampirkan kebijakan. 14. Untuk Tambahkan kebijakan untuk barang Anda, pilih ClientDevicePolicy, Register Thing.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh sertifikat CA dari perangkat inti Greengrass.	Jika Anda mengharapkan perangkat inti Greengras s berfungsi di lingkungan offline, Anda harus membuat sertifikat CA inti Greengrass tersedia untuk perangkat klien sehingga dapat memverifi kasi sertifikat broker MQTT (yang dikeluarkan oleh CA inti Greengrass). Karena itu, penting untuk mendapatkan salinan sertifikat ini. Gunakan salah satu pendekatan berikut untuk mengunduh sertifikat CA: • Jika Anda memiliki akses jaringan ke perangkat AWS IoT Greengrass dari `https://<device IP>:8883` PC Anda, masukkan di browser web Anda dan lihat sertifikat broker MQTT dan sertifika t CA. Anda juga dapat menyimpan sertifikat CA ke perangkat klien. • Atau, Anda dapat menggunakan baris perintah OpenSSL: `openssl s_client -showcerts -connect <device IP>:8883`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin kredensi di perangkat klien.	Salin sertifikat CA inti Greengrass, sertifikat perangkat, dan kunci pribadi di perangkat klien.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kaitkan perangkat klien dengan perangkat inti.	Mengaitkan perangkat klien dengan perangkat inti sehingga mereka dapat menemukan perangkat inti. Perangkat klien kemudian dapat menggunakan Greengrass discovery API untuk mengambil informasi konektivitas dan sertifikat untuk perangkat inti terkait mereka. Untuk informasi selengkapnya, lihat Mengaitkan perangkat klien dalam dokumentasi AWS IoT Greengrass. 1. Di konsol AWS IoT Greengrass, pilih perangkat Core. 2. Pilih perangkat inti untuk dikelola. 3. Pada halaman detail perangkat inti, pilih tab Perangkat klien. 4. Di bagian Perangkat klien terkait, pilih Kaitkan perangkat klien. 5. Di modal Kaitkan perangkat klien dengan perangkat inti, lakukan hal berikut untuk setiap perangkat klien yang akan dikaitkan: a. Masukkan nama hal AWS IoT untuk diasosias	AWS IoT Greengrass

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ikan sebagai perangkat klien. b. Pilih Tambahkan. 6. Pilih Kaitkan. Perangkat klien yang telah Anda kaitkan sekarang dapat menggunakan Greengras s penemuan API untuk menemukan perangkat inti ini.	

Mengirim dan menerima data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim data dari satu perangkat klien ke perangkat klien lain.	Gunakan klien MQTT di perangkat Anda untuk mempublikasikan pesan tentang topik tersebut. `dt/client1/sensor`	AWS Umum
Kirim data dari perangkat klien ke AWS IoT Core.	Gunakan klien MQTT di perangkat Anda untuk mempublikasikan pesan tentang topik tersebut. `dt/client1/sensor` Di klien pengujian MQTT, berlangganan topik tempat perangkat mengirim pesan, atau berlangganan # untuk semua topik (lihat detail).	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim pesan dari AWS IoT Core ke perangkat klien.	Pada halaman klien pengujian MQTT, di tab Publikasikan ke topik, di bidang Nama topik, masukkan nama topik pesan Anda. Dalam contoh ini, gunakan `cmd/client1` untuk topik.	AWS Umum

Pemecahan masalah

Isu	Solusi
Tidak dapat memverifikasi kesalahan sertifikat server	Kesalahan ini terjadi ketika klien MQTT tidak dapat memverifikasi sertifikat yang disajikan oleh broker MQTT selama jabat tangan TLS. Alasan paling umum adalah bahwa klien MQTT tidak memiliki sertifikat CA. Ikuti langkah-l angkah ini untuk memastikan bahwa sertifikat CA diberikan kepada klien MQTT. 1. Jika Anda memiliki akses jaringan ke perangkat AWS IoT Greengrass dari `https://<device IP>:8883` PC Anda, masukkan di jendela browser untuk melihat sertifikat broker MQTT dan sertifikat CA. Anda juga dapat menyimpan sertifikat CA ke perangkat klien. Atau gunakan baris perintah OpenSSL: `openssl s_client -showcerts -connect <device IP>:8883` 2. Simpan konten sertifikat Moquette CA dan Greengrass Core CA ke dalam file, lalu

Isu	Solusi

	lihat konten yang diterjemahkan dengan menggunakan perintah:
	
	`openssl x509 -in <Name of CA>.pem -text`
	
	Sertifikat Moquette CA harus menunjukkan bidang SAN seperti dalam contoh ini:
	
	`X509v3 Subject Alternative Name:`
	`IP Address:XXX.XXX.XXX.XXX, IP Address:127.0.0.1, DNS:localhost`

Tidak dapat memverifikasi kesalahan nama server	Kesalahan ini terjadi ketika klien MQTT tidak dapat memverifikasi bahwa itu terhubung ke server yang benar. Alasan paling umum adalah bahwa alamat IP perangkat Greengrass tidak terdaftar di bidang SAN sertifikat. Ikuti petunjuk di solusi sebelumnya untuk mendapatkan sertifikat broker MQTT dan verifikasi bahwa bidang SAN berisi alamat IP perangkat AWS IoT Greengrass, seperti yang dijelaskan di bagian Informasi tambahan. Jika tidak, konfirmasikan bahwa komponen detektor IP diinstal dengan benar dan restart perangkat inti.

Isu	Solusi
Tidak dapat memverifikasi nama server hanya saat menghubungkan dari perangkat klien yang disematkan	Mbed TLS, yang merupakan pustaka TLS populer yang digunakan dalam perangkat tertanam, saat ini mendukung verifikasi nama DNS hanya di bidang SAN sertifikat, seperti yang ditunjukkan dalam kode perpustakaan Mbed TLS. Karena perangkat inti tidak memiliki nama domain sendiri dan bergantung pada alamat IP, klien TLS yang menggunakan Mbed TLS akan gagal verifikasi nama server selama jabat tangan TLS, menyebabkan kegagalan koneksi. Kami menyarankan Anda menambahk an verifikasi alamat IP SAN ke pustaka Mbed TLS Anda di fungsi x509_crt_check_san.

Sumber daya terkait

- Dokumentasi AWS IoT Greengrass
- Dokumentasi AWS IoT Core
- Komponen broker MQTT
- Komponen jembatan MQTT
- Komponen autentikasi perangkat klien
- Komponen detektor IP
- SDK Perangkat AWS IoT s
- Menerapkan Perangkat Klien Lokal dengan AWS IoT Greengrass (postingan blog AWS)
- RFC 5280 - Profil Sertifikat Infrastruktur Kunci Publik Internet X.509 dan Daftar Pencabutan Sertifikat (CRL)

Informasi tambahan

Bagian ini memberikan informasi tambahan tentang komunikasi antara perangkat klien dan perangkat inti.

Broker MQTT mendengarkan port 8883 di perangkat inti untuk upaya koneksi klien TLS. Ilustrasi berikut menunjukkan contoh sertifikat server broker MQTT.

Sertifikat contoh menampilkan rincian berikut:

- Sertifikat ini dikeluarkan oleh AWS IoT Greengrass Core CA, yang bersifat lokal dan khusus untuk perangkat inti; artinya, ia bertindak sebagai CA lokal.
- Sertifikat ini secara otomatis diputar setiap minggu oleh komponen autentikasi klien seperti yang ditunjukkan pada ilustrasi berikut. Anda dapat mengatur interval ini dalam konfigurasi komponen autentikasi klien.

- Nama alternatif subjek (SAN) memainkan peran penting dalam verifikasi nama server di ujung klien TLS. Ini membantu klien TLS memastikan bahwa itu terhubung ke server yang benar dan membantu menghindari man-in-the-middle serangan selama pengaturan sesi TLS. Dalam contoh sertifikat, bidang SAN menunjukkan bahwa server ini mendengarkan di localhost (soket domain Unix lokal), dan antarmuka jaringan memiliki alamat IP 192.168.1.12.

Klien TLS menggunakan bidang SAN dalam sertifikat untuk memverifikasi bahwa itu terhubung ke server yang sah selama verifikasi server. Sebaliknya, selama jabat tangan TLS khas antara server HTTP dan browser, nama domain di bidang common name (CN) atau bidang SAN digunakan untuk memeriksa silang domain yang sebenarnya terhubung ke browser selama proses verifikasi server. Jika perangkat inti tidak memiliki nama domain, alamat IP yang disertakan dalam bidang SAN memiliki tujuan yang sama. Untuk informasi selengkapnya, lihat bagian Nama Alternatif Subjek dari Profil RFC 5280 - Internet X.509 Sertifikat Infrastruktur Kunci Publik dan Daftar Pencabutan Sertifikat (CRL).

Komponen detektor IP di AWS IoT Greengrass memastikan bahwa alamat IP yang benar disertakan dalam bidang SAN sertifikat.

Sertifikat dalam contoh ditandatangani oleh perangkat AWS IoT Greengrass yang bertindak sebagai CA lokal. Klien TLS (klien MQTT) tidak mengetahui CA ini, jadi kami harus memberikan sertifikat CA yang terlihat seperti berikut ini.

Lebih banyak pola

- Mengkonsumsi data IoT secara hemat biaya langsung ke Amazon S3 menggunakan AWS IoT Greengrass
- Terapkan sistem agen di Amazon Bedrock dengan framework CreWai dengan menggunakan Terraform
- Menerapkan aplikasi kontainer saat AWS IoT Greengrass V2 berjalan sebagai wadah Docker

Migrasi & modernisasi

Topik

- Migrasi
- Modernisasi
- Mainframe

Migrasi

Topik

- Buat CloudFormation templat AWS untuk tugas AWS DMS menggunakan Microsoft Excel dan Python
- Memulai dengan penemuan portofolio otomatis
- Memigrasikan beban kerja Cloudera lokal ke Platform Data Cloudera di AWS
- Mengatasi kesalahan koneksi setelah memigrasikan Microsoft SQL Server ke AWS Cloud
- Mulai ulang AWS Replication Agent secara otomatis tanpa menonaktifkan SELinux setelah me-reboot server sumber RHEL
- Arsitek ulang
- Rehost
- Pindah
- Platform Ulang
- Pola migrasi berdasarkan beban kerja
- Lebih banyak pola

Buat CloudFormation templat AWS untuk tugas AWS DMS menggunakan Microsoft Excel dan Python

Venkata Naveen Koppula, Amazon Web Services

Ringkasan

Pola ini menguraikan langkah-langkah untuk membuat CloudFormation template AWS secara otomatis untuk AWS Database Migration Service (AWS DMS) menggunakan Microsoft Excel dan Python.

Migrasi database menggunakan AWS DMS sering kali melibatkan pembuatan CloudFormation templat AWS untuk menyediakan tugas AWS DMS. Sebelumnya, membuat CloudFormation template AWS membutuhkan pengetahuan tentang bahasa pemrograman JSON atau YAMG. Dengan alat ini, Anda hanya perlu pengetahuan dasar tentang Excel dan cara menjalankan skrip Python menggunakan terminal atau jendela perintah.

Sebagai input, alat ini mengambil buku kerja Excel yang menyertakan nama tabel yang akan dimigrasikan, Amazon Resource Names (ARNs) dari titik akhir AWS DMS, dan instans replikasi AWS DMS. Alat ini kemudian menghasilkan CloudFormation templat AWS untuk tugas AWS DMS yang diperlukan.

Untuk langkah mendetail dan informasi latar belakang, lihat posting blog Membuat CloudFormation templat AWS untuk tugas AWS DMS menggunakan Microsoft Excel di blog AWS Database.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Microsoft Excel versi 2016 atau yang lebih baru
- Python versi 2.7 atau yang lebih baru
- Modul Python xlrd (diinstal pada prompt perintah dengan perintah: pip install xlrd)
- Sumber AWS DMS dan titik akhir target serta instans replikasi AWS DMS

Batasan

- Nama-nama skema, tabel, dan kolom terkait diubah menjadi karakter huruf kecil di titik akhir tujuan.

- Alat ini tidak membahas pembuatan titik akhir AWS DMS dan instance replikasi.
- Saat ini, alat ini hanya mendukung satu skema untuk setiap tugas AWS DMS.

Arsitektur

Tumpukan teknologi sumber

- Database lokal
- Microsoft Excel

Tumpukan teknologi target

- CloudFormation Templat AWS
- Database di AWS Cloud

Arsitektur

Alat

- Pycharm IDE, atau lingkungan pengembangan terintegrasi (IDE) apa pun yang mendukung Python versi 3.6
- Microsoft Office 2016 (untuk Microsoft Excel)

Epik

Konfigurasikan jaringan, instans replikasi AWS DMS, dan titik akhir

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jika perlu, mintalah peningkatan kuota layanan.	Minta peningkatan kuota layanan untuk tugas AWS DMS jika diperlukan.	AWS Umum
Konfigurasikan Wilayah AWS, virtual private cloud (VPCs),		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
rentang CIDR, Availability Zone, dan subnet.		
Konfigurasikan instans replikasi AWS DMS.	Instans replikasi AWS DMS dapat terhubung ke database lokal dan AWS.	AWS Umum
Konfigurasikan titik akhir AWS DMS.	Konfigurasikan titik akhir untuk basis data sumber dan target.	AWS Umum

Siapkan lembar kerja untuk tugas dan tag AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan daftar tabel.	Daftar semua tabel yang terlibat dalam migrasi.	Basis Data
Siapkan lembar kerja tugas.	Siapkan lembar kerja Excel menggunakan daftar tabel yang Anda konfigurasikan.	AWS Umum, Microsoft Excel
Siapkan lembar kerja tag.	Detail tag sumber daya AWS untuk dilampirkan ke tugas AWS DMS.	AWS Umum, Microsoft Excel

Unduh dan jalankan alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh dan ekstrak alat pembuatan template dari GitHub repositori.	GitHub repositori:/ https://github.com/aws-samples/dms-cloudformation-templates-generator	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan alat.	Ikuti petunjuk terperinci dalam posting blog yang tercantum di bawah “Referensi dan bantuan.”	

Sumber daya terkait

- Buat CloudFormation templat AWS untuk tugas AWS DMS menggunakan Microsoft Excel (posting blog)
- Generator CloudFormation Template DMS (GitHub repositori)
- Dokumentasi Python
- deskripsi dan unduhan xlrd
- Dokumentasi AWS DMS
- CloudFormation Dokumentasi AWS

Memulai dengan penemuan portofolio otomatis

Pratik Chunawala dan Rodolfo Jr. Cerrada, Amazon Web Services

Ringkasan

Menilai portofolio dan mengumpulkan metadata merupakan tantangan penting saat memigrasikan aplikasi dan server ke Amazon Web Services (AWS) Cloud, terutama untuk migrasi besar yang memiliki lebih dari 300 server. Menggunakan alat penemuan portofolio otomatis dapat membantu Anda mengumpulkan informasi tentang aplikasi Anda, seperti jumlah pengguna, frekuensi penggunaan, dependensi, dan informasi tentang infrastruktur aplikasi. Informasi ini sangat penting saat merencanakan gelombang migrasi sehingga Anda dapat memprioritaskan dan mengelompokkan aplikasi dengan sifat serupa dengan benar. Menggunakan alat penemuan merampingkan komunikasi antara tim portofolio dan pemilik aplikasi karena tim portofolio dapat memvalidasi hasil alat penemuan daripada mengumpulkan metadata secara manual. Pola ini membahas pertimbangan utama untuk memilih alat penemuan otomatis dan informasi tentang cara menerapkan dan mengujinya di lingkungan Anda.

Pola ini mencakup templat, yang merupakan titik awal untuk membangun daftar periksa aktivitas tingkat tinggi Anda sendiri. Di sebelah daftar periksa adalah templat untuk matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI). Anda dapat menggunakan matriks RACI ini untuk menentukan siapa yang bertanggung jawab atas setiap tugas dalam daftar periksa Anda.

Epik

Pilih alat penemuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan apakah alat penemuan sesuai untuk kasus penggunaan Anda.	Alat penemuan mungkin bukan solusi terbaik untuk kasus penggunaan Anda. Pertimbangkan jumlah waktu yang diperlukan untuk memilih, mendapatkan, menyiapkan, dan menyebarkan alat penemuan. Diperlukan waktu 4-8 minggu untuk	Pimpinan migrasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menyiapkan alat pemindaian untuk alat penemuan tanpa agen di lingkungan Anda atau untuk memasang agen ke semua beban kerja dalam ruang lingkup. Setelah digunakan, Anda harus mengizinkan 4—12 minggu agar alat penemuan mengumpulkan metadata dengan memindai beban kerja aplikasi dan melakukan analisis tumpukan aplikasi. Jika Anda bermigrasi kurang dari 100 server, Anda mungkin dapat mengumpulkan metadata secara manual dan menganalisis dependensi lebih cepat daripada waktu yang diperlukan untuk menerapkan dan mengumpulkan metadata dengan alat penemuan otomatis.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih alat penemuan.	Tinjau Pertimbangan untuk memilih alat penemuan otomatis di bagian Informasi tambahan. Tentukan kriteria yang sesuai untuk memilih alat penemuan untuk kasus penggunaan Anda, dan kemudian evaluasi setiap alat berdasarkan kriteria tersebut. Untuk daftar lengkap alat penemuan otomatis, lihat Alat migrasi Penemuan, Perencanaan, dan Rekomendasi.	Pimpinan migrasi, insinyur Migrasi

Bersiaplah untuk instalasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan daftar periksa pra-penerapan.	Buat daftar periksa tugas yang harus Anda selesaikan sebelum menggunakan alat. Sebagai contoh, lihat Daftar Periksa Pra-Deployment di situs web dokumentasi Flexera.	Membangun prospek, Insinyur migrasi, Pimpinan migrasi, Administrator jaringan
Siapkan persyaratan jaringan.	Menyediakan port, protokol, alamat IP, dan routing yang diperlukan untuk alat untuk menjalankan dan mengakses server target. Untuk informasi selengkapnya, lihat panduan instalasi untuk alat penemuan	Insinyur migrasi, Administrator jaringan, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda. Sebagai contoh, lihat Persyaratan Penerapan di situs web dokumentasi Flexera.	
Siapkan akun dan persyaratan kredensi.	Identifikasi kredensil yang Anda butuhkan untuk mengakses server target dan menginstal semua komponen alat.	Administrator cloud, AWS Umum, Insinyur migrasi, Pimpinan migrasi, Administrator jaringan, Administrator AWS
Siapkan peralatan tempat Anda akan memasang alat.	Pastikan bahwa peralatan tempat Anda akan memasang komponen alat memenuhi spesifikasi dan persyaratan platform untuk alat tersebut.	Insinyur migrasi, Pimpinan migrasi, Administrator jaringan
Siapkan perintah perubahan.	Menurut proses manajemen perubahan di organisasi Anda, siapkan pesanan perubahan apa pun yang diperlukan, dan pastikan pesanan perubahan ini disetujui.	Bangun prospek, Pimpinan migrasi
Kirim persyaratan ke pemangku kepentingan.	Kirim daftar periksa pra-penerapan dan persyaratan jaringan ke pemangku kepentingan. Pemangku kepentingan harus meninjau, mengevaluasi, dan menyiapkan persyaratan yang diperlukan sebelum melanjutkan penyebaran.	Bangun prospek, Pimpinan migrasi

Menyebarkan alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh penginstal.	Unduh installer atau gambar mesin virtual. Gambar mesin virtual biasanya datang dalam Open Virtualization Format (OVF).	Bangun prospek, Pimpinan migrasi
Ekstrak file.	Jika Anda menggunakan penginstal, Anda harus mengunduh dan menjalankan penginstal di server lokal.	Bangun prospek, Pimpinan migrasi
Menyebarkan alat di server.	Menerapkan alat penemuan di server lokal target sebagai berikut: • Jika file sumber Anda adalah gambar mesin virtual, gunakan ke lingkungan mesin virtual Anda, seperti VMware. • Jika file sumber Anda adalah penginstal, jalankan penginstal untuk menginstal dan mengatur alat.	Membangun prospek, Pimpinan migrasi, Administrator jaringan
Masuk ke alat penemuan.	Ikuti petunjuk di layar, dan masuk untuk memulai dengan alat ini.	Pimpin migrasi, Bangun prospek
Aktifkan produk.	Masukkan kunci lisensi Anda.	Bangun prospek, Pimpinan migrasi
Konfigurasikan alat.	Masukkan kredensil apa pun yang diperlukan untuk	Bangun prospek, Pimpinan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengakses server target, seperti kredensil untuk Windows, Simple Network Management Protocol (SNMP) VMware, dan Secure Shell Protocol (SSH), atau database.	

Uji alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih server uji.	Identifikasi sekumpulan kecil subnet non-produksi atau alamat IP yang dapat Anda gunakan untuk menguji alat penemuan. Ini membantu Anda memvalidasi pemindaian dengan cepat, mengidentifikasi dan memecahkan masalah kesalahan apa pun dengan cepat, dan mengisolasi pengujian Anda dari lingkungan produksi.	Membangun prospek, Pimpinan migrasi, Administrator jaringan
Mulai memindai server pengujian yang dipilih.	Untuk alat penemuan tanpa agen, masukkan subnet atau alamat IP untuk server pengujian yang dipilih di konsol alat penemuan, dan mulai pemindaian. Untuk alat penemuan berbasis agen, instal agen di server pengujian yang dipilih.	Membangun prospek, Pimpinan migrasi, Administrator jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau hasil pemindaian.	Tinjau hasil pemindaian untuk server pengujian. Jika ada kesalahan yang ditemukan, pecahkan masalah dan perbaiki kesalahan. Dokumentasikan kesalahan dan solusinya. Anda mereferensikan informasi ini di masa mendatang, dan Anda dapat menambahk an informasi ini ke runbook portofolio Anda.	Membangun prospek, Pimpinan migrasi, Administr ator jaringan
Memindai ulang server uji.	Setelah pemindaian ulang selesai, ulangi pemindaian sampai tidak ada kesalahan.	Membangun prospek, Pimpinan migrasi, Administr ator jaringan

Sumber daya terkait

AWS sumber daya

- Panduan penilaian portofolio aplikasi untuk AWS Cloud migrasi
- Alat migrasi Penemuan, Perencanaan, dan Rekomendasi

Panduan penerapan untuk alat penemuan yang umum dipilih

- Terapkan alat virtual RN15 0 (dokumentasi Flexera)
- Instalasi Gatherer (dokumentasi ModelizeIt)
- Instalasi Server Analisis On-Prem (dokumentasi ModelizeIt)

Informasi tambahan

Pertimbangan untuk memilih alat penemuan otomatis

Setiap alat penemuan memiliki manfaat dan keterbatasan. Saat memilih alat yang sesuai untuk kasus penggunaan Anda, pertimbangkan hal berikut:

- Pilih alat penemuan yang dapat mengumpulkan sebagian besar, jika tidak semua, metadata yang Anda butuhkan untuk mencapai tujuan penilaian portofolio Anda.
- Identifikasi metadata apa pun yang perlu Anda kumpulkan secara manual karena alat tidak mendukungnya.
- Menyediakan persyaratan alat penemuan kepada para pemangku kepentingan sehingga mereka dapat meninjau dan menilai alat berdasarkan persyaratan keamanan dan kepatuhan internal mereka, seperti server, jaringan, dan persyaratan kredensi.
  - Apakah alat mengharuskan Anda menginstal agen dalam beban kerja dalam ruang lingkup?
  - Apakah alat ini mengharuskan Anda menyiapkan alat virtual di lingkungan Anda?
- Tentukan persyaratan residensi data Anda. Beberapa organisasi tidak ingin menyimpan data mereka di luar lingkungan mereka. Untuk mengatasinya, Anda mungkin perlu menginstal beberapa komponen alat di lingkungan lokal.
- Pastikan alat mendukung sistem operasi (OS) dan versi OS dari beban kerja dalam lingkup.
- Tentukan apakah portofolio Anda mencakup server mainframe, mid-range, dan lawas. Sebagian besar alat penemuan dapat mendeteksi beban kerja ini sebagai dependensi, tetapi beberapa alat mungkin tidak bisa mendapatkan detail perangkat, seperti pemanfaatan dan dependensi server. Device42 dan alat penemuan ModernizeIT mendukung server mainframe dan mid-range.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Memigrasikan beban kerja Cloudera lokal ke Platform Data Cloudera di AWS

Battulga Purevragchaa dan Nidhi Gupta, Amazon Web Services

Nijjwol Lamsal, Cloudera, Inc.

Ringkasan

Pola ini menjelaskan langkah-langkah tingkat tinggi untuk memigrasikan beban kerja Cloudera Distributed Hadoop (CDH), Hortonworks Data Platform (HDP), dan Cloudera Data Platform (CDP) lokal Anda ke CDP Public Cloud di AWS. Kami menyarankan Anda bermitra dengan Cloudera Professional Services dan integrator sistem (SI) untuk menerapkan langkah-langkah ini.

Ada banyak alasan mengapa pelanggan Cloudera ingin memindahkan beban kerja CDH, HDP, dan CDP lokal mereka ke cloud. Beberapa alasan khas meliputi:

- Merampingkan adopsi paradigma platform data baru seperti data lakehouse atau data mesh
- Meningkatkan kelincahan bisnis, mendemokratisasikan akses dan inferensi aset data yang ada
- Menurunkan total biaya kepemilikan (TCO)
- Meningkatkan elastisitas beban kerja
- Mengaktifkan skalabilitas yang lebih besar; secara drastis mengurangi waktu untuk menyediakan layanan data dibandingkan dengan basis instalasi lokal yang lama
- Pensiun perangkat keras lama; secara signifikan mengurangi siklus penyegaran perangkat keras
- Manfaatkan pay-as-you-go harga, yang diperluas ke beban kerja Cloudera di AWS dengan model lisensi Cloudera (CCU)
- Manfaatkan penyebaran yang lebih cepat dan integrasi yang lebih baik dengan platform integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)
- Gunakan platform terpadu tunggal (CDP) untuk beberapa beban kerja

Cloudera mendukung semua beban kerja utama, termasuk Machine Learning, Data Engineering, Data Warehouse, Operational Database, Stream Processing (CSP), dan keamanan dan tata kelola data. Cloudera telah menawarkan beban kerja ini selama bertahun-tahun di lokasi, dan Anda dapat memigrasikan beban kerja ini ke AWS Cloud dengan menggunakan CDP Public Cloud dengan Workload Manager dan Replication Manager.

Cloudera Shared Data Experience (SDX) menyediakan katalog metadata bersama di seluruh beban kerja ini untuk memfasilitasi pengelolaan dan operasi data yang konsisten. SDX juga mencakup keamanan terperinci yang komprehensif untuk melindungi dari ancaman, dan tata kelola terpadu untuk kemampuan audit dan pencarian untuk kepatuhan terhadap standar seperti Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS) dan GDPR.

Sekilas tentang migrasi CDP

Beban kerja	Beban kerja sumber	CDH, HDP, dan CDP Private Cloud
	Lingkungan sumber	• Windows, Linux • Lokal, kolokasi, atau lingkungan non-AWS
	Beban kerja tujuan	Cloud Publik CDP di AWS
	Lingkungan tujuan	• Model penyebaran: akun pelanggan • Model operasi: bidang customer/Cloudera kontrol
Migrasi	Strategi migrasi (7Rs)	Rehost, replatform, atau refactor
	Apakah ini peningkatan dalam versi beban kerja?	Ya
	Durasi migrasi	• Deployment: Sekitar 1 minggu untuk membuat akun pelanggan, virtual private cloud (VPC), dan lingkungan yang dikelola pelanggan CDP Public Cloud. • Durasi migrasi: 1-4 bulan, tergantung pada kompleksi tas dan ukuran beban kerja.

Biaya	Biaya menjalankan beban kerja di AWS	• Pada tingkat tinggi, biaya migrasi beban kerja CDH ke AWS mengasumsikan bahwa Anda akan membangun lingkungan baru di AWS. Ini termasuk akuntansi untuk waktu dan upaya personel serta penyediaan sumber daya komputasi dan perangkat lunak lisensi untuk lingkungan baru. • Model penetapan harga berbasis konsumsi cloud Cloudera memberi Anda fleksibilitas untuk memanfaatkan kemampuan penskalaan yang meledak dan otomatis. Untuk informasi selengkapnya, lihat tarif layanan CDP Public Cloud di situs web Cloudera. • Cloudera Enterprise Data Hub didasarkan pada Amazon Elastic Compute Cloud EC2 (Amazon) dan memodelkan cluster tradisional dengan cermat. Data Hub dapat disesuaikan, tetapi ini akan mempengaruhi biaya. • CDP Public Cloud Data Warehouse, Cloudera Machine Learning, dan Cloudera Data Engineering

		(CDE) berbasis container dan dapat dikonfigurasi untuk skala secara otomatis.
Perjanjian dan kerangka infrastruktur	Persyaratan sistem	Lihat bagian Prasyarat.
	SLA	Lihat Perjanjian Tingkat Layanan Cloudera untuk CDP Public Cloud.
	DR	Lihat Pemulihan Bencana di dokumentasi Cloudera.
	Model lisensi dan pengoperasian (untuk akun AWS target)	Bawa model Lisensi Anda Sendiri (BYOL)
Kepatuhan	Persyaratan keamanan	Lihat Ikhtisar Keamanan Cloudera di dokumentasi Cloudera.
	Sertifikasi kepatuhan lainnya	Lihat informasi di situs web Cloudera tentang kepatuhan Peraturan Perlindungan Data Umum (GDPR) dan Pusat Kepercayaan CDP.

Prasyarat dan batasan

Prasyarat

- Persyaratan akun AWS, termasuk akun, sumber daya, layanan, dan izin, seperti pengaturan peran dan kebijakan AWS Identity and Access Management (IAM)
- Prasyarat untuk menyebarkan CDP dari situs web Cloudera

Migrasi membutuhkan peran dan keahlian berikut:

Peran	Keterampilan dan tanggung jawab

Pimpin migrasi	Memastikan dukungan eksekutif, kolaborasi tim, perencanaan, implementasi, dan penilaian
UKM Cloudera	Keterampilan ahli dalam administrasi CDH, HDP, dan CDP, administrasi sistem, dan arsitektur
Arsitek AWS	Keterampilan dalam layanan AWS, jaringan, keamanan, dan arsitektur

Arsitektur

Membangun arsitektur yang sesuai adalah langkah penting untuk memastikan bahwa migrasi dan kinerja memenuhi harapan Anda. Agar upaya migrasi Anda memenuhi asumsi buku pedoman ini, lingkungan data target Anda di AWS Cloud, baik pada instance yang dihosting virtual private cloud (VPC) atau CDP, harus setara dengan lingkungan sumber Anda dalam hal versi sistem operasi dan perangkat lunak serta spesifikasi mesin utama.

Diagram berikut (direproduksi dengan izin dari lembar data Pengalaman Data Bersama Cloudera) menunjukkan komponen infrastruktur untuk lingkungan CDP dan bagaimana tingkatan atau komponen infrastruktur berinteraksi.

Arsitektur mencakup komponen CDP berikut:

- Data Hub adalah layanan untuk meluncurkan dan mengelola klaster beban kerja yang didukung oleh Cloudera Runtime. Anda dapat menggunakan definisi klaster di Data Hub untuk menyediakan dan mengakses klaster beban kerja untuk kasus penggunaan kustom dan menentukan konfigurasi klaster kustom. Untuk informasi lebih lanjut, lihat situs web Cloudera.
- Aliran Data dan Streaming mengatasi tantangan utama yang dihadapi perusahaan dengan data yang bergerak. Ini mengelola yang berikut:
  - Memproses streaming data real-time pada volume tinggi dan skala tinggi
  - Melacak asal data dan garis keturunan data streaming
  - Mengelola dan memantau aplikasi edge dan sumber streaming

  Untuk informasi lebih lanjut, lihat Cloudera DataFlow dan CSP di situs web Cloudera.

- Rekayasa Data mencakup integrasi data, kualitas data, dan tata kelola data, yang membantu organisasi membangun dan memelihara jalur data dan alur kerja. Untuk informasi lebih lanjut, lihat situs web Cloudera. Pelajari tentang dukungan untuk instans spot guna memfasilitasi penghematan biaya AWS untuk beban kerja Rekayasa Data Cloudera.
- Data Warehouse memungkinkan Anda membuat gudang data dan data mart independen yang secara otomatis menskalakan untuk memenuhi tuntutan beban kerja. Layanan ini menyediakan instans komputasi terisolasi dan pengoptimalan otomatis untuk setiap gudang data dan data mart, dan membantu Anda menghemat biaya saat rapat. SLAs Untuk informasi lebih lanjut, lihat situs web Cloudera. Pelajari cara mengelola biaya dan auto-scaling untuk Cloudera Data Warehouse di AWS.
- Database Operasional dalam CDP menyediakan fondasi yang andal dan fleksibel untuk aplikasi berkinerja tinggi yang dapat diskalakan. Ini memberikan database real-time, selalu tersedia, terukur yang melayani data terstruktur tradisional bersama data baru yang tidak terstruktur dalam platform operasional dan pergudangan terpadu. Untuk informasi lebih lanjut, lihat situs web Cloudera.
- Machine Learning adalah platform pembelajaran mesin cloud-native yang menggabungkan kemampuan ilmu data swalayan dan rekayasa data menjadi satu layanan portabel dalam cloud data perusahaan. Ini memungkinkan penyebaran pembelajaran mesin dan kecerdasan buatan (AI) yang dapat diskalakan pada data di mana saja. Untuk informasi lebih lanjut, lihat situs web Cloudera.

CDP di AWS

Diagram berikut (diadaptasi dengan izin dari situs web Cloudera) menunjukkan arsitektur CDP tingkat tinggi di AWS. CDP mengimplementasikan model keamanannya sendiri untuk mengelola akun dan aliran data. Ini terintegrasi dengan IAM melalui penggunaan peran lintas akun.

Pesawat kontrol CDP berada di akun master Cloudera di VPC-nya sendiri. Setiap akun pelanggan memiliki sub-akun dan VPC uniknya sendiri. Peran IAM lintas akun dan teknologi SSL mengarahkan lalu lintas manajemen ke dan dari bidang kontrol ke layanan pelanggan yang berada di subnet publik yang dapat dirutekan internet dalam setiap VPC pelanggan. Pada VPC pelanggan, Cloudera Shared Data Experience (SDX) menyediakan keamanan kekuatan perusahaan dengan tata kelola dan kepatuhan terpadu sehingga Anda bisa mendapatkan wawasan dari data Anda dengan lebih cepat. SDX adalah filosofi desain yang dimasukkan ke dalam semua produk Cloudera. Untuk informasi selengkapnya tentang SDX dan arsitektur jaringan Cloud Publik CDP untuk AWS, lihat dokumentasi Cloudera.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan Kubernetes di AWS tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Otomatisasi dan perkakas

- Untuk perkakas tambahan, Anda dapat menggunakan Cloudera Backup Data Recovery (BDR), AWS Snowball, dan AWS Snowmobile untuk membantu memigrasikan data dari CDH, HDP, dan CDP lokal ke CDP yang dihosting AWS.
- Untuk penerapan baru, sebaiknya gunakan AWS Partner Solution untuk CDP.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Libatkan tim Cloudera.	Cloudera mengejar model keterlibatan standar dengan pelanggannya dan dapat bekerja dengan integrato r sistem (SI) Anda untuk mempromosikan pendekata	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n yang sama. Hubungi tim pelanggan Cloudera sehingga mereka dapat memberikan panduan dan sumber daya teknis yang diperlukan untuk memulai proyek. Menghubun gi tim Cloudera memastika n bahwa semua tim yang diperlukan dapat mempersia pkan migrasi saat tanggalnya semakin dekat. Anda dapat menghubun gi Cloudera Professional Services untuk memindahkan penyebaran Cloudera Anda dari pilot ke produksi dengan cepat, dengan biaya lebih rendah, dan dengan kinerja puncak. Untuk daftar lengkap penawaran, lihat situs web Cloudera.	
Buat lingkungan Cloud Publik CDP di AWS untuk VPC Anda.	Bekerja dengan Cloudera Professional Services atau SI Anda untuk merencanakan dan menerapkan CDP Public Cloud ke dalam VPC di AWS.	Arsitek awan, UKM Cloudera

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memprioritaskan dan menilai beban kerja untuk migrasi.	Evaluasi semua beban kerja lokal Anda untuk menentukan beban kerja yang paling mudah untuk dimigrasi. Aplikasi yang tidak kritis adalah yang terbaik untuk bergerak terlebih dahulu, karena mereka akan memiliki dampak minimal pada pelanggan Anda. Simpan beban kerja penting misi untuk yang terakhir, setelah Anda berhasil memigrasikan beban kerja lainnya. **Note** Beban kerja transient (CDP Data Engineering) lebih mudah untuk dimigrasikan daripada beban kerja persisten (CDP Data Warehouse). Penting juga untuk mempertimbangkan volume data dan lokasi saat bermigrasi. Tantangan dapat mencakup mereplikasi data secara terus menerus dari lingkungan lokal ke cloud, dan mengubah pipeline konsumsi data untuk	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengimpor data langsung ke cloud.	
Diskusikan CDH, HDP, CDP, dan aktivitas migrasi aplikasi lama.	Pertimbangkan dan mulailah merencanakan kegiatan berikut dengan Cloudera Workload Manager: • Data dan beban kerja untuk disalin ke lingkungan AWS Anda • Data siap cloud • Tetangga yang berisik, yang menggunakan sumber daya dan menciptakan masalah bagi penyewa lain • Beban kerja elastis • Cluster kecil dengan overhead operasional tinggi	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi persyaratan dan rekomendasi Cloudera Replication Manager.	Bekerja sama dengan Cloudera Professional Services dan SI Anda untuk mempersiapkan migrasi beban kerja ke lingkungan CDP Public Cloud Anda di AWS. Memahami persyaratan dan rekomendasi berikut dapat membantu Anda menghinda ri masalah umum selama dan setelah Anda menginstal layanan Manajer Replikasi. • Tinjau dokumen pendukung Manajer Replikasi untuk mengonfirmasi bahwa Anda memenuhi persyaratan lingkungan dan sistem. Untuk informasi selengkap nya, lihat Matriks Support untuk CDP Public Cloud Replication Manager di situs web Cloudera. • Anda tidak memerlukan akses root ke node tempat mesin Replication Manager App dan Data Lifecycle Manager (DLM) akan diinstal. • Instal Apache Hive selama instalasi awal Replicati on Manager, kecuali Anda yakin bahwa Anda tidak akan menggunak	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an replikasi Hive di masa depan. Jika Anda memutuskan untuk menginstal Hive setelah membuat kebijakan replikasi HDFS di Manajer Replikasi, Anda harus menghapus dan kemudian membuat ulang semua kebijakan replikasi HDFS setelah menambahk an Hive. • Cluster yang digunakan dalam Manajer Replikasi harus memiliki konfigura si simetris. Setiap cluster dalam hubungan replikasi harus dikonfigurasi persis sama untuk keamanan (Kerberos), manajemen pengguna (LDAP/AD), dan Knox Proxy. Layanan cluster seperti Hadoop Distributed File System (HDFS), Apache Hive, Apache Knox, Apache Ranger, dan Apache Atlas dapat memiliki konfigura si yang berbeda untuk ketersediaan tinggi (HA). Misalnya, cluster sumber dan target mungkin memiliki konfigurasi HA dan non-HA yang terpisah.	

Migrasi CDP ke AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan beban kerja pertama untuk dev/test lingkungan menggunakan Cloudera Workload Manager.	SI Anda dapat membantu Anda memigrasikan beban kerja pertama Anda ke AWS Cloud. Ini harus menjadi aplikasi yang tidak menghadapi pelanggan atau kritis misi. Kandidat ideal untuk dev/test migrasi adalah aplikasi yang memiliki data yang dapat dengan mudah dikonsumsi oleh cloud, seperti beban kerja Rekayasa Data CDP. Ini adalah beban kerja sementara yang biasanya memiliki lebih sedikit pengguna yang mengaksesnya, dibandingkan dengan beban kerja persisten seperti beban kerja CDP Data Warehouse yang dapat memiliki banyak pengguna yang membutuhkan akses tanpa gangguan. Beban kerja Rekayasa Data tidak persisten, yang meminimal kan dampak bisnis jika terjadi kesalahan. Namun, pekerjaan ini bisa menjadi penting untuk pelaporan produksi, jadi prioritaskan beban kerja Rekayasa Data berdampak rendah terlebih dahulu.	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ulangi langkah migrasi seperlunya.	Cloudera Workload Manager membantu mengidentifikasi beban kerja yang paling cocok untuk cloud. Ini menyediakan metrik seperti peringkat kinerja cloud, sizing/capacity rencana untuk lingkungan target, dan rencana replikasi. Kandidat terbaik untuk migrasi adalah beban kerja musiman, pelaporan ad hoc, dan pekerjaan intermiten yang tidak menghabiskan banyak sumber daya. Cloudera Replication Manager memindahkan data dari lokasi ke cloud, dan dari cloud ke tempat. Secara proaktif mengoptimalkan beban kerja, aplikasi, kinerja, dan kapasitas infrastruktur untuk pergudangan data, rekayasa data, dan pembelajaran mesin dengan menggunakan Workload Manager. Untuk panduan lengkap tentang cara memodernisasi gudang data, lihat situs web Cloudera.	UKM Cloudera

Sumber daya terkait

Dokumentasi Cloudera:

- Mendaftarkan cluster klasik dengan CDP, Cloudera Manager, dan Replication Manager:
  - Konsol Manajemen
  - Replikasi Replikasi Manajer sarang
- Replikasi penjaga
- Izin penjaga
- Daftar periksa perencanaan klaster Data Hub
- Arsitektur Manajer Beban Kerja
- Persyaratan Manajer Replikasi
- Observabilitas Platform Data Cloudera
- Persyaratan AWS

Dokumentasi AWS:

- Migrasi Data Cloud

Mengatasi kesalahan koneksi setelah memigrasikan Microsoft SQL Server ke AWS Cloud

Premkumar Chelladurai, Amazon Web Services

Ringkasan

Setelah Anda memigrasikan Microsoft SQL Server yang berjalan di Windows Server 2008 R2, 2012, atau 2012 R2 ke instans Amazon Elastic Compute Cloud (EC2Amazon) di Amazon Web Services (AWS) Cloud, koneksi ke SQL Server gagal dan kesalahan berikut muncul:

- `[Microsoft][ODBC SQL Server Driver][DBNETLIB] General Network error`
- `ERROR [08S01] [Microsoft][SQL Native Client]Communication link failure. System.Data.SqlClient.SqlException: A transport-level error has occurred when sending the request to the server. (provider: TCP Provider, error: 0 - An existing connection was forcibly closed by the remote host.)`
- `TCP Provider: The semaphore timeout period has expired`

Pola ini menjelaskan bagaimana Anda dapat mengatasi kesalahan ini dengan mematikan fitur Windows Scalable Networking Pack (SNP) di sistem operasi (OS) dan tingkat antarmuka jaringan untuk SQL Server yang berjalan pada Windows Server 2008 R2, 2012, atau 2012 R2.

Prasyarat dan batasan

Prasyarat

- Hak istimewa administrator untuk Windows Server.
- Jika Anda menggunakan AWS Application Migration Service sebagai alat migrasi, Anda memerlukan salah satu versi Windows Server berikut:
  - Windows Server 2008 R2 Paket Layanan 1, 2012, atau 2012 R2
- Jika Anda menggunakan CloudEndure Migrasi sebagai alat migrasi, Anda memerlukan salah satu versi Windows Server berikut:
  - Windows Server 2003 R2 Paket Layanan 3, 2008, 2008 R2 Paket Layanan 1, 2012, atau 2012 R2

Alat

- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, dan Anda dapat meningkatkan skala atau menskalakan.
- Windows Server — Windows Server adalah platform untuk membangun infrastruktur aplikasi, jaringan, dan layanan web yang terhubung.

Epik

Matikan fitur SNP di tingkat OS dan elastic network interface

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan fitur SNP di tingkat OS.	1. Masuk ke Windows Server dan buka prompt perintah sebagai administrator. 2. Jalankan perintah `netsh int tcp show global`. 3. Dalam output, periksa apakah salah satu `Receive-Side Scaling` atau `Chimney Offload` dalam `enabled` mode. Jika salah satu dari merekaenabled, jalankan perintah berikut: • `netsh int tcp set global chimney=disabled` • `netsh int tcp set global rss=disabled`	Administrator AWS, administrator sistem AWS, Insinyur migrasi, Administrator cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan fitur SNP di level elastic network interface.	1. Pilih Mulai, masukkan`ncpa.cpl`, lalu tekan Enter. 2. Klik kanan Adaptor Jaringan Elastis. 3. Di menu popup, pilih Properties. 4. Di jendela Ethernet Adapter Properties, pilih Configure. 5. Di jendela popup Amazon Elastic Network Adapter Properties, pilih tab Advanced. 6. Di bagian Properti, matikan semua offload dan RSS.	Administrator AWS, administrator Cloud, administrator sistem AWS

Sumber daya terkait

- Cara memecahkan masalah fitur kinerja jaringan tingkat lanjut seperti RSS dan NetDMA

Mulai ulang AWS Replication Agent secara otomatis tanpa menonaktifkan SELinux setelah me-reboot server sumber RHEL

Anil Kunapareddy, Venkatramana Chintha, dan Shanmugam Shanker, Amazon Web Services

Ringkasan

AWS Application Migration Service membantu menyederhanakan, mempercepat, dan mengotomatiskan migrasi beban kerja Red Hat Enterprise Linux (RHEL) Anda ke Amazon Web Services (AWS) Cloud. Untuk menambahkan server sumber ke Layanan Migrasi Aplikasi, Anda menginstal AWS Replication Agent di server.

Layanan Migrasi Aplikasi menyediakan replikasi real-time, asinkron, tingkat blok. Ini berarti Anda dapat melanjutkan operasi TI normal selama seluruh proses replikasi. Operasi TI ini mungkin mengharuskan Anda me-reboot atau memulai ulang server sumber RHEL Anda selama migrasi. Jika ini terjadi, AWS Replication Agent tidak akan dimulai ulang secara otomatis, dan replikasi data Anda akan berhenti. Biasanya, Anda dapat menyetel Security-Enhanced Linux (SELinux) ke mode nonaktif atau permisif untuk memulai ulang AWS Replication Agent secara otomatis. Namun, kebijakan keamanan organisasi Anda mungkin melarang penonaktifan SELinux, dan Anda mungkin juga harus memberi label ulang file Anda.

Pola ini menjelaskan cara memulai ulang AWS Replication Agent secara otomatis tanpa mematikan SELinux saat server sumber RHEL Anda melakukan reboot atau memulai ulang selama migrasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Beban kerja RHEL lokal yang ingin Anda migrasikan ke AWS Cloud.
- Layanan Migrasi Aplikasi diinisialisasi dari konsol Layanan Migrasi Aplikasi. Inisialisasi hanya diperlukan saat pertama kali Anda menggunakan layanan ini. Untuk petunjuk, lihat dokumentasi Layanan Migrasi Aplikasi.
- Kebijakan AWS Identity and Access Management (IAM) yang ada untuk Layanan Migrasi Aplikasi. Untuk informasi selengkapnya, lihat dokumentasi Layanan Migrasi Aplikasi.

Versi

- RHEL versi 7 atau yang lebih baru

Alat

Layanan AWS

- AWS Application Migration Service adalah solusi lift-and-shift (rehost) yang sangat otomatis yang menyederhanakan, mempercepat, dan mengurangi biaya migrasi aplikasi ke AWS.

Perintah Linux

Tabel berikut menyediakan daftar perintah Linux yang akan Anda jalankan di server sumber RHEL Anda. Ini juga dijelaskan dalam epos dan cerita untuk pola ini.

Perintah	Deskripsi
`#systemctl –version`	Mengidentifikasi versi sistem.
`#systemctl list-units --type=service`	Daftar semua layanan aktif yang tersedia di server RHEL.
`#systemctl list-units --type=service | grep running`	Daftar semua layanan yang saat ini berjalan di server RHEL.
`#systemctl list-units --type=service | grep failed`	Daftar semua layanan yang gagal dimuat setelah server RHEL reboot atau restart.
`restorecon -Rv /etc/rc.d/init.d/aws-replication-service`	Mengubah konteksnya menjadi `aws-replication-service` .
`yum install policycoreutils*`	Menginstal utilitas inti kebijakan yang diperlukan untuk pengoperasian SELinux sistem.
`ausearch -c "insmod" --raw | audit2allow -M my-modprobe`	Mencari log audit dan membuat modul untuk kebijakan.
`semodule -i my-modprobe.pp`	Mengaktifkan kebijakan.
`cat my-modprobe.te`	Menampilkan isi `my-modprobe.te` file.

`semodule -l | grep my-modprobe`	Memeriksa apakah kebijakan telah dimuat ke SELinux modul.

Epik

Instal AWS Replication Agent dan reboot server sumber RHEL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna Layanan Migrasi Aplikasi dengan kunci akses dan kunci akses rahasia.	Untuk menginstal AWS Replication Agent, Anda harus membuat pengguna Layanan Migrasi Aplikasi dengan kredensi AWS yang diperlukan. Untuk petunjuk, lihat dokumentasi Layanan Migrasi Aplikasi.	Insinyur migrasi
Instal AWS Replication Agent.	1. Masuk ke AWS Management Console dan buka konsol AWS Migration Service di https://console.aws.amazon.com/mgn/rumah. 2. Konfigurasikan pengaturan replikasi dengan mengikuti petunjuk dalam dokumentasi Layanan Migrasi Aplikasi. 3. Instal AWS Replication Agent dengan mengikuti petunjuk dalam dokumentasi Layanan Migrasi Aplikasi. 4. Pada halaman Server Sumber, pilih server sumber RHEL, lalu pilih Replikasi untuk memulai	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	replikasi awal. Untuk informasi selengkapnya, lihat dokumentasi Layanan Migrasi Aplikasi.	
Mulai ulang atau reboot server sumber RHEL.	Mulai ulang atau reboot server sumber RHEL Anda saat status replikasi datanya ditampilkan Terhenti di dasbor Migrasi.	Insinyur migrasi
Periksa status replikasi data.	Tunggu selama satu jam dan kemudian periksa status replikasi Data lagi di dasbor Migrasi. Itu harus dalam keadaan sehat.	Insinyur migrasi

Periksa status AWS Replication Agent di server sumber RHEL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi versi sistem.	Buka antarmuka baris perintah untuk server sumber RHEL Anda dan jalankan perintah berikut untuk mengidentifikasi versi sistem: `#systemctl –version`	Insinyur migrasi
Daftar semua layanan aktif.	Untuk membuat daftar semua layanan aktif yang tersedia di server RHEL, jalankan perintah:	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`#systemctl list-units --type=service`	
Daftar semua layanan yang berjalan.	Untuk membuat daftar semua layanan yang sedang berjalan di server RHEL, gunakan perintah: `#systemctl list-units --type=service | grep running`	Insinyur migrasi
Buat daftar semua layanan yang gagal dimuat.	Untuk membuat daftar semua layanan yang gagal dimuat setelah server RHEL reboot atau restart, jalankan perintah: `#systemctl list-units --type=service | grep failed`	Insinyur migrasi

Buat dan jalankan SELinux modul

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah konteks keamanan.	Di antarmuka baris perintah untuk server sumber RHEL Anda, jalankan perintah berikut untuk mengubah konteks keamanan ke layanan replikasi AWS: `restorecon -Rv /etc/rc.d/init.d/aws-replication-service`	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal utilitas inti.	Untuk menginstal utilitas inti yang diperlukan untuk pengoperasian SELinux sistem dan kebijakannya, jalankan perintah: `yum install policycoreutils*`	Insinyur migrasi
Cari log audit dan buat modul untuk kebijakan.	Jalankan perintah : `ausearch -c "insmod" --raw | audit2allow -M my-modprobe`	Insinyur migrasi
Menampilkan isi my-modprobe-te file.	`my-modprobe.te` File dihasilkan oleh perintah audit2allow. Ini mencakup SELinux domain, direktori sumber kebijakan, dan subdirektori, dan menentukan aturan vektor akses dan transisi yang terkait dengan domain. Untuk menampilkan isi file, jalankan perintah: `cat my modprobe.te`	Insinyur migrasi
Aktifkan kebijakan.	Untuk menyisipkan modul dan membuat paket kebijakan aktif, jalankan perintah: `semodule -i my-modprobe.pp`	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa apakah modul telah dimuat.	Jalankan perintah : `semodule -l | grep my-modprobe` Setelah SELinux modul dimuat, Anda tidak perlu lagi mengatur SELinux ke mode nonaktif atau permisif selama migrasi Anda.	Insinyur migrasi
Reboot atau restart server sumber RHEL dan verifikasi status replikasi data.	Buka konsol AWS Migration Service, navigasikan ke Progres replikasi data, lalu reboot atau restart server sumber RHEL Anda. Replikasi data sekarang harus dilanjutk an secara otomatis setelah server sumber RHEL reboot.	Insinyur migrasi

Sumber daya terkait

- Dokumentasi layanan Migrasi Aplikasi
- Materi pelatihan teknis
- Memecahkan masalah Agen Replikasi AWS
- Kebijakan Layanan Migrasi Aplikasi

Arsitek ulang

Topik

- Mengkonversi VARCHAR2 (1) tipe data untuk Oracle ke tipe data Boolean untuk Amazon Aurora PostgreSQL
- Buat pengguna dan peran aplikasi di Aurora PostgreSQL yang kompatibel
- Meniru Oracle DR dengan menggunakan database global Aurora yang kompatibel dengan PostgreSQL
- Menerapkan SHA1 hashing untuk data PII saat bermigrasi dari SQL Server ke PostgreSQL
- Bermigrasi secara bertahap dari Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL menggunakan Oracle SQL Developer dan AWS SCT
- Muat file BLOB ke TEXT dengan menggunakan pengkodean file di Aurora PostgreSQL yang kompatibel
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dengan dan menggunakan dan AWS SCT AWS DMS AWS CLI CloudFormation
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dalam mode SSL dengan menggunakan AWS DMS
- Migrasikan paket pragma Oracle SERIALLY_REUSABLE ke PostgreSQL
- Migrasikan tabel eksternal Oracle ke Amazon Aurora PostgreSQL yang kompatibel
- Migrasikan indeks berbasis fungsi dari Oracle ke PostgreSQL
- Migrasikan fungsi asli Oracle ke PostgreSQL menggunakan ekstensi
- Migrasikan database Db2 dari Amazon ke EC2 Aurora MySQL yang kompatibel dengan menggunakan AWS DMS
- Migrasi database Microsoft SQL Server dari Amazon EC2 ke Amazon DocumentDB dengan menggunakan AWS DMS
- Memigrasi database ThoughtSpot Falcon lokal ke Amazon Redshift
- Migrasi dari Oracle Database ke Amazon RDS for PostgreSQL dengan menggunakan Oracle GoldenGate
- Migrasikan tabel partisi Oracle ke PostgreSQL menggunakan AWS DMS
- Bermigrasi dari Amazon RDS for Oracle ke Amazon RDS untuk MySQL
- Bermigrasi dari IBM Db2 di Amazon EC2 ke Aurora PostgreSQL yang kompatibel dengan menggunakan AWS DMS dan AWS SCT

- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Amazon RDS for PostGresQL dan AWS DMS SharePlex
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk PostgreSQL menggunakan tampilan terwujud dan AWS DMS
- Bermigrasi dari Oracle di Amazon EC2 ke Amazon RDS untuk MySQL menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS untuk MariaDB menggunakan AWS DMS dan AWS SCT
- Memigrasikan database Oracle lokal ke Amazon RDS for MySQL menggunakan AWS DMS dan AWS SCT
- Memigrasikan database Oracle lokal ke Amazon RDS for PostgreSQL menggunakan Oracle bystander dan AWS DMS
- Migrasikan Database Oracle ke Amazon Redshift menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle ke Aurora PostgreSQL menggunakan AWS DMS dan AWS SCT
- Memigrasi data dari database Oracle lokal ke Aurora PostgreSQL
- Bermigrasi dari SAP ASE ke Amazon RDS for SQL Server menggunakan AWS DMS
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan AWS DMS
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT
- Migrasi aplikasi lama dari Oracle Pro*C ke ECPG
- Migrasikan kolom virtual yang dihasilkan dari Oracle ke PostgreSQL
- Siapkan fungsionalitas Oracle UTL_FILE pada Aurora PostgreSQL yang kompatibel
- Validasi objek database setelah bermigrasi dari Oracle ke Amazon Aurora PostgreSQL

Mengkonversi VARCHAR2 (1) tipe data untuk Oracle ke tipe data Boolean untuk Amazon Aurora PostgreSQL

Naresh Damera, Amazon Web Services

Ringkasan

Selama migrasi dari Amazon Relational Database Service (Amazon RDS) untuk Oracle ke Amazon Aurora PostgreSQL Compatible Edition, Anda mungkin mengalami ketidakcocokan data saat memvalidasi migrasi di (). AWS Database Migration Service AWS DMS Untuk mencegah ketidakcocokan ini, Anda dapat mengonversi VARCHAR2 (1) tipe data ke tipe data Boolean.

VARCHAR2 tipe data menyimpan string teks panjang variabel, dan VARCHAR2 (1) menunjukkan bahwa string adalah 1 karakter panjangnya atau 1 byte. Untuk informasi selengkapnya VARCHAR2, lihat tipe data bawaan Oracle (dokumentasi Oracle).

Dalam pola ini, di kolom tabel data sumber sampel, VARCHAR2 (1) data adalah Y, untuk Ya, atau N, untuk No. Pola ini mencakup instruksi untuk menggunakan AWS DMS dan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi tipe data ini dari nilai Y dan N di VARCHAR2 (1) ke nilai benar atau salah di Boolean.

Audiens yang dituju

Pola ini direkomendasikan bagi mereka yang memiliki pengalaman migrasi database Oracle ke Aurora PostgreSQL kompatibel dengan menggunakan. AWS DMS Saat Anda menyelesaikan migrasi, patuhi rekomendasi dalam Mengonversi Oracle ke Amazon RDS for PostgreSQL atau Amazon Aurora PostgreSQL (dokumentasi).AWS SCT

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Konfirmasikan bahwa lingkungan Anda disiapkan untuk Aurora, termasuk menyiapkan kredensil, izin, dan grup keamanan. Untuk informasi selengkapnya, lihat Menyiapkan lingkungan Anda untuk Amazon Aurora (dokumentasi Aurora).
- Sumber Amazon RDS for Oracle database yang berisi VARCHAR2 kolom tabel dengan (1) data.
- Instance database yang kompatibel dengan Amazon Aurora PostgreSQL target. Untuk informasi selengkapnya, lihat Membuat cluster database dan menghubungkan ke database pada cluster database PostgreSQL Aurora (dokumentasi Aurora).

Versi produk

- Amazon RDS for Oracle versi 12.1.0.2 atau yang lebih baru.
- AWS DMS versi 3.1.4 atau yang lebih baru. Untuk informasi selengkapnya, lihat Menggunakan database Oracle sebagai sumber untuk AWS DMS dan Menggunakan database PostgreSQL sebagai target untuk (dokumentasi). AWS DMSAWS DMS Kami menyarankan Anda menggunakan versi terbaru AWS DMS untuk versi dan dukungan fitur yang paling komprehensif.
- AWS Schema Conversion Tool (AWS SCT) versi 1.0.632 atau yang lebih baru. Kami menyarankan Anda menggunakan versi terbaru AWS SCT untuk versi dan dukungan fitur yang paling komprehensif.
- Aurora mendukung versi PostgreSQL yang tercantum dalam Versi Mesin Database untuk Aurora PostgreSQL kompatibel (dokumentasi Aurora).

Arsitektur

Tumpukan teknologi sumber

Amazon RDS for Oracle database instance

Tumpukan teknologi target

Contoh database yang kompatibel dengan Amazon Aurora PostgreSQL

Arsitektur sumber dan target

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke dalam AWS Cloud atau di antara kombinasi pengaturan cloud dan lokal.
- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di. AWS Cloud

- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Layanan lainnya

- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi yang menyederhanakan pengembangan dan pengelolaan database Oracle baik dalam penerapan tradisional maupun berbasis cloud. Dalam pola ini, Anda menggunakan alat ini untuk terhubung ke instance database Amazon RDS for Oracle dan menanyakan data.
- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database. Dalam pola ini, Anda menggunakan alat ini untuk terhubung ke instance database Aurora dan menanyakan data.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat laporan migrasi database.	1. Di AWS SCT, buat laporan penilaian migrasi database. Untuk informasi selengkapnya, lihat Membuat laporan penilaian migrasi. 2. Tinjau dan lakukan item tindakan dalam laporan penilaian migrasi. Untuk informasi selengkapnya, lihat Item tindakan laporan penilaian.	DBA, Pengembang
Nonaktifkan kendala kunci asing pada database target.	Di PostgreSQL, kunci asing diimplementasikan dengan menggunakan pemicu. Selama fase beban penuh,	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS DMS muat setiap tabel satu per satu. Kami sangat menyarankan agar Anda menonaktifkan batasan kunci asing selama pemuatan penuh dengan menggunakan salah satu metode berikut: • Nonaktifkan sementara semua pemicu dari instans, dan selesaikan beban penuh. • Gunakan parameter `session_replicatio n_role` di PostgreSQL. Jika menonaktifkan batasan kunci asing tidak layak, buat tugas AWS DMS migrasi untuk data primer yang khusus untuk tabel induk dan tabel anak.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Nonaktifkan kunci utama dan kunci unik pada database target.	Menggunakan perintah berikut, nonaktifkan kunci utama dan kendala pada database target. Ini membantu meningkatkan kinerja tugas pemuatan awal.
	`ALTER TABLE <table> DISABLE PRIMARY KEY;`
	`ALTER TABLE <table> DISABLE CONSTRAINT <constraint_name>;`
		DBA, Pengembang

Buat tugas pemuatan awal.	Di AWS DMS, buat tugas migrasi untuk pemuatan awal. Untuk petunjuk, lihat Membuat tugas. Untuk metode migrasi, pilih Migrasi data yang ada. Metode migrasi ini disebut `Full Load` dalam API. Jangan memulai tugas ini dulu.	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Edit pengaturan tugas untuk tugas pemuatan awal.	Edit pengaturan tugas untuk menambahkan validasi data. Pengaturan validasi ini harus dibuat dalam file JSON. Untuk petunjuk dan contoh, lihat Menentukan setelan tugas. Tambahkan validasi berikut: • Untuk memvalidasi bahwa VARCHAR2 (1) data secara akurat dikonversi ke Boolean dalam database target, tambahkan kode dalam skrip validasi Data di bagian Informasi tambahan dari pola ini. Skrip validasi mengubah nilai Boolean dari 1 ke Y dan 0 ke N dalam tabel target, dan kemudian membandingkan nilai-nilai dalam tabel target ke tabel sumber. Untuk memvalidasi sisa migrasi data, aktifkan validasi data dalam tugas. Untuk informasi selengkapnya, lihat Pengaturan tugas validasi data.	Administrator AWS, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas replikasi yang sedang berlangsung.	Di AWS DMS, buat tugas migrasi yang membuat database target tetap sinkron dengan database sumber. Untuk petunjuk, lihat Membuat tugas. Untuk metode migrasi, pilih Replikasi perubahan data saja. Jangan memulai tugas ini dulu.	DBA

Uji tugas migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat data sampel untuk pengujian.	Dalam database sumber, buat tabel sampel dengan data untuk tujuan pengujian.	Developer
Konfirmasikan tidak ada kegiatan yang saling bertentangan.	Gunakan tombol `pg_stat_activity` untuk memeriksa aktivitas apa pun di server yang mungkin mengganggu migrasi. Untuk informasi lebih lanjut, lihat The Statistics Collector (dokumentasi PostgreSQL).	Administrator AWS
Mulai tugas AWS DMS migrasi.	Di AWS DMS konsol, di halaman Dasbor, mulai pemuatan awal dan tugas replikasi berkelanjutan yang Anda buat di epik sebelumnya.	Administrator AWS
Pantau tugas dan status beban tabel.	Selama migrasi, pantau status tugas dan status tabel. Saat	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tugas pemuatan awal selesai, pada tab Statistik tabel: • Status Beban harus Tabel selesai. • Status validasi harus divalidasi.	
Verifikasi hasil migrasi.	Menggunakan pgAdmin, kueri tabel pada database target. Kueri yang berhasil menunjukkan bahwa data berhasil dimigrasi.	Developer
Tambahkan kunci utama dan kunci asing pada database target.	Buat kunci utama dan kunci asing pada basis data target. Untuk informasi lebih lanjut, lihat ALTER TABLE (situs web PostgreSQL).	DBA
Bersihkan data pengujian.	Pada basis data sumber dan target, bersihkan data yang dibuat untuk pengujian unit.	Developer

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Selesaikan migrasi.	Ulangi epik sebelumnya, Uji tugas migrasi, menggunak an data sumber nyata. Ini memigrasikan data dari sumber ke database target.	Developer
Validasi bahwa basis data sumber dan target sinkron.	Validasi bahwa basis data sumber dan target sinkron.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi dan instruksi selengkapnya, lihat validasi AWS DMS data.	
Hentikan database sumber.	Hentikan database Amazon RDS for Oracle. Untuk petunjuknya, lihat Menghentikan instans Amazon RDS DB sementara. Saat Anda menghentikan basis data sumber, beban awal dan tugas replikasi yang sedang berlangsung akan dihentikan secara otomatis. AWS DMS Tidak ada tindakan tambahan yang diperlukan untuk menghentikan tugas-tugas ini.	Developer

Sumber daya terkait

AWS referensi

- Migrasikan database Oracle ke Aurora PostgreSQL menggunakan dan (Panduan Preskriptif) AWS DMS AWS SCTAWS
- Mengonversi Oracle ke Amazon RDS untuk PostgreSQL atau Amazon Aurora PostgreSQL (dokumentasi)AWS SCT
- Cara AWS DMS Kerja (AWS DMS dokumentasi)

Referensi lainnya

- Tipe data Boolean (dokumentasi PostgreSQL)
- Tipe data bawaan Oracle (dokumentasi Oracle)
- pgAdmin (situs web pgAdmin)

- Pengembang SQL (situs web Oracle)

Tutorial dan video

- Memulai dengan AWS DMS
- Memulai Dengan Amazon RDS
- Pengantar AWS DMS (Video)
- Memahami Amazon RDS (Video)

Informasi tambahan

Skrip validasi data

Skrip validasi data berikut mengkonversi 1 ke Y dan 0 ke N. Ini membantu AWS DMS tugas berhasil menyelesaikan dan lulus validasi tabel.

```
{
"rule-type": "validation",
"rule-id": "5",
"rule-name": "5",
"rule-target": "column",
"object-locator": {
"schema-name": "ADMIN",
"table-name": "TEMP_CHRA_BOOL",
"column-name": "GRADE"
},
"rule-action": "override-validation-function",
"target-function": "case grade when '1' then 'Y' else 'N' end"
        }
```

`case`Pernyataan dalam skrip melakukan validasi. Jika validasi gagal, AWS DMS menyisipkan catatan dalam tabel public.awsdms_validation_failures_v1 pada instance database target. Catatan ini mencakup nama tabel, waktu kesalahan, dan detail tentang nilai yang tidak cocok dalam tabel sumber dan target.

Jika Anda tidak menambahkan skrip validasi data ini ke AWS DMS tugas dan data dimasukkan dalam tabel target, AWS DMS tugas menunjukkan status validasi sebagai Catatan Tidak Cocokkan.

Selama AWS SCT konversi, tugas AWS DMS migrasi mengubah tipe data VARCHAR2 (1) tipe data ke Boolean dan menambahkan kendala kunci utama pada kolom. `''N0''`

Buat pengguna dan peran aplikasi di Aurora PostgreSQL yang kompatibel

Abhishek Verma, Amazon Web Services

Ringkasan

Saat Anda bermigrasi ke Amazon Aurora PostgreSQL Compatible Edition, pengguna database dan peran yang ada di database sumber harus dibuat dalam database yang kompatibel dengan Aurora PostgreSQL. Anda dapat membuat pengguna dan peran di Aurora PostgreSQL kompatibel dengan menggunakan dua pendekatan yang berbeda:

- Gunakan pengguna dan peran serupa dalam target seperti dalam database sumber. Dalam pendekatan ini, bahasa definisi data (DDLs) diekstraksi untuk pengguna dan peran dari database sumber. Kemudian mereka ditransformasikan dan diterapkan ke database target yang kompatibel dengan Aurora PostgreSQL. Misalnya, posting blog Gunakan SQL untuk memetakan pengguna, peran, dan hibah dari Oracle ke PostgreSQL mencakup menggunakan ekstraksi dari mesin database sumber Oracle.
- Gunakan pengguna standar dan peran yang umum digunakan selama pengembangan, administrasi, dan untuk melakukan operasi terkait lainnya dalam database. Ini termasuk operasi read-only, read/write, development, administration, dan deployment yang dilakukan oleh masing-masing pengguna.

Pola ini berisi hibah yang diperlukan untuk pengguna dan pembuatan peran di Aurora PostgreSQL kompatibel yang diperlukan untuk pengguna standar dan pendekatan peran. Langkah-langkah pembuatan pengguna dan peran disejajarkan dengan kebijakan keamanan untuk memberikan hak istimewa paling sedikit kepada pengguna database. Tabel berikut mencantumkan pengguna, peran yang sesuai, dan detailnya di database.

Pengguna	Peran	Tujuan
`APP_read`	`APP_RO`	Digunakan untuk akses hanya-baca pada skema `APP`
`APP_WRITE`	`APP_RW`	Digunakan untuk operasi tulis dan baca pada skema `APP`
`APP_dev_user`	`APP_DEV`	Digunakan untuk tujuan pengembangan pada

		skemaAPP_DEV, dengan akses hanya-baca pada skema APP
Admin_User	rds_superuser	Digunakan untuk melakukan operasi administrator pada database
APP	APP_DEP	Digunakan untuk membuat objek di bawah APP skema dan untuk penyebaran objek dalam skema APP

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif
- Database PostgreSQL, database Edisi Amazon Aurora PostgreSQL yang kompatibel, atau Amazon Relational Database Service (Amazon RDS) untuk database PostgreSQL

Versi produk

- Semua versi PostgreSQL

Arsitektur

Tumpukan teknologi sumber

- Database apa pun

Tumpukan teknologi target

- Kompatibel dengan Amazon Aurora PostgreSQL

Arsitektur target

Diagram berikut menunjukkan peran pengguna dan arsitektur skema dalam database Aurora PostgreSQL kompatibel.

Otomatisasi dan skala

Pola ini berisi pengguna, peran, dan skrip pembuatan skema, yang dapat Anda jalankan beberapa kali tanpa berdampak pada pengguna database sumber atau target yang ada.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.

Layanan lainnya

- psql adalah alat front-end berbasis terminal yang diinstal dengan setiap instalasi PostgreSQL Database. Ini memiliki antarmuka baris perintah untuk menjalankan SQL, PL-PGSQL, dan perintah sistem operasi.
- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.

Epik

Buat pengguna dan peran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna penerapan.	Pengguna penyebaran APP akan digunakan untuk membuat dan memodifik asi objek database selama penerapan. Gunakan skrip berikut untuk membuat	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	peran pengguna penerapan APP_DEP dalam skema. APP Validasi hak akses untuk memastikan pengguna ini hanya memiliki hak istimewa untuk membuat objek dalam skema yang diperlukan. APP
	1. Connect ke pengguna admin, dan buat skema.
	`CREATE SCHEMA APP;`
	2. Buat pengguna.
	`CREATE USER APP WITH PASSWORD <password> ;`
	3. Buat peran.
	`CREATE ROLE APP_DEP ; GRANT all on schema APP to APP_DEP ; GRANT USAGE ON SCHEMA APP to APP_DEP ; GRANT connect on database <db_name> to APP_DEP ; GRANT APP_DEP to APP;`
	4. Untuk menguji hak istimewa, sambungkan ke APP dan buat tabel.
	`set search_path to APP; SET`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>CREATE TABLE test(id
	 integer) ;
	CREATE TABLE</pre>
	5. Periksa hak istimewa.
	<pre>select schemaname ,
	 tablename , tableowne
	r from pg_tables
	 where tablename like
	 'test' ;
	schemaname
	 | tablename |
	 tableowner
	APP |
	 test | APP</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat pengguna hanya-baca.	Pengguna hanya-baca `APP_read` akan digunakan untuk melakukan operasi hanya-baca dalam skema. APP Gunakan skrip berikut untuk membuat pengguna hanya-baca. Validasi hak akses untuk memastikan bahwa pengguna ini memiliki hak istimewa hanya untuk membaca objek dalam skema APP dan untuk secara otomatis memberikan akses baca untuk objek baru yang dibuat dalam skema. APP
	
	1. Buat penggunaAPP_read.
	
	`create user APP_read ;`
	`alter user APP_read`
	`with password`
	`'your_password' ;`
	
	2. Buat peran.
	
	`CREATE ROLE APP_ro ;`
	`GRANT SELECT ON ALL`
	`TABLES IN SCHEMA APP`
	`TO APP_RO ;`
	`GRANT USAGE ON SCHEMA`
	`APP TO APP_RO`
	`GRANT CONNECT ON`
	`DATABASE testdb TO`
	`APP_RO ;`
	`GRANT APP_RO TO`
	`APP_read;`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Untuk menguji hak istimewa, masuk menggunakan APP_read pengguna.
	
	`set search_path to APP ;`
	`create table test1( id integer) ;`
	`ERROR: permission denied for schema APP`
	`LINE 1: create table test1( id integer) ;`
	`insert into test values (34) ;`
	`ERROR: permission denied for table test`
	`SQL state: 42501`
	`select from test`
	`no rows selected`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat read/write pengguna.	read/write Pengguna `APP_WRITE` akan digunakan untuk melakukan operasi baca dan tulis pada skemaAPP. Gunakan skrip berikut untuk membuat read/writ e pengguna dan berikan `APP_RW` perannya. Validasi hak akses untuk memastikan bahwa pengguna ini memiliki hak baca dan tulis hanya pada objek dalam skema APP dan untuk secara otomatis memberikan akses baca dan tulis untuk objek baru yang dibuat dalam skema. APP
	
	1. Buat pengguna.
	
	`CREATE USER APP_WRITE ; alter user APP_WRITE with password 'your_password' ;`
	
	2. Buat peran.
	
	`CREATE ROLE APP_RW; GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA APP TO APP_RW ; GRANT CONNECT ON DATABASE postgres to APP_RW ; GRANT USAGE ON SCHEMA APP to APP_RW ;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>ALTER DEFAULT
	 PRIVILEGES IN SCHEMA
	 APP
	GRANT SELECT, INSERT,
	 UPDATE, DELETE ON
	 TABLES TO APP_RW ;
	GRANT APP_RW to
	 APP_WRITE</pre>
	3. Untuk menguji hak istimewa, masuk menggunakan APP_WRITE pengguna.
	<pre>SET SEARCH_PATH to
	 APP;
	CREATE TABLE
	 test1(id integer) ;
	ERROR: permission
	 denied for schema APP
	LINE 1: create table
	 test1(id integer) ;
	SELECT * FROM test ;
	id
	---- 	
	12
	INSERT INTO test
	 values (31) ;
	INSERT 0 1</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat pengguna admin.	Pengguna admin `Admin_User` akan digunakan untuk melakukan operasi admin pada database. Contoh operasi ini adalah `CREATE ROLE` dan`CREATE DATABASE`. `Admin_User` menggunakan peran bawaan `rds_superuser` untuk melakukan operasi admin pada database. Gunakan skrip berikut untuk membuat dan menguji hak istimewa bagi pengguna admin `Admin_User` dalam database.
	1. Buat pengguna dan berikan peran.
	`create user Admin_User WITH PASSWORD ‘Your password’ ALTER user Admin_user CREATEDB; ALTER user Admin_user CREATEROLE;`
	2. Untuk menguji hak istimewa, masuk dari `Admin_User` pengguna.
	`SELECT * FROM APP.test ; id ---- 31 CREATE ROLE TEST ;`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`CREATE DATABASE test123 ;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat pengguna pengemban gan.	Pengguna pengembangan `APP_dev_user` akan memiliki hak untuk membuat objek dalam skema lokalnya APP_DEV dan membaca akses dalam skemaAPP. Gunakan skrip berikut untuk membuat dan menguji hak istimewa pengguna `APP_dev_user` dalam database.
	1. Buat pengguna.
	`CREATE USER APP1_dev_ user with password ‘your password’;`
	2. Buat APP_DEV skema untuk. `App_dev_user`
	`CREATE SCHEMA APP1_DEV ;`
	3. Buat APP_DEV peran.
	`CREATE ROLE APP1_DEV ; GRANT APP1_RO to APP1_DEV ; GRANT SELECT ON ALL TABLES IN SCHEMA APP1_DEV to APP1_dev_ user GRANT USAGE, CREATE ON SCHEMA APP1_DEV to APP1_DEV_USER GRANT APP1_DEV to APP1_DEV_USER ;`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Untuk menguji hak istimewa, masuk dariAPP_dev_user .
	
	`CREATE TABLE APP1_dev.test1( id integer ) ;`
	`CREATE TABLE`
	`INSERT into APP1_dev.test1 ( select * from APP1.test );`
	`INSERT 0 1`
	`CREATE TABLE APP1.test4 ( id int) ;`
	`ERROR: permission denied for schema APP1`
	`LINE 1: create table APP1.test4 ( id int) ;`	

Sumber daya terkait

Dokumentasi PostgreSQL

- BUAT PERAN
- BUAT PENGGUNA
- Peran yang telah ditentukan

Informasi tambahan

PostgreSQL 14 peningkatan

PostgreSQL 14 menyediakan seperangkat peran yang telah ditentukan sebelumnya yang memberikan akses ke kemampuan dan informasi tertentu yang umum dibutuhkan dan istimewa. Administrator (termasuk peran yang memiliki `CREATE ROLE` hak istimewa) dapat memberikan peran

ini atau peran lain di lingkungan mereka kepada pengguna, memberi mereka akses ke kemampuan dan informasi yang ditentukan.

Administrator dapat memberikan pengguna akses ke peran ini menggunakan `GRANT` perintah. Misalnya, untuk memberikan `pg_signal_backend` peran`Admin_User`, Anda dapat menjalankan perintah berikut.

```
GRANT pg_signal_backend TO Admin_User;
```

`pg_signal_backend`Peran ini dimaksudkan untuk memungkinkan administrator mengaktifkan peran non-superuser tepercaya untuk mengirim sinyal ke backend lain. Untuk informasi lebih lanjut, lihat peningkatan PostgreSQL 14.

Akses fine-tuning

Dalam beberapa kasus, mungkin perlu menyediakan akses yang lebih terperinci ke pengguna (misalnya, akses berbasis tabel atau akses berbasis kolom). Dalam kasus seperti itu, peran tambahan dapat dibuat untuk memberikan hak istimewa tersebut kepada pengguna. Untuk informasi lebih lanjut, lihat Hibah PostgreSQL.

Meniru Oracle DR dengan menggunakan database global Aurora yang kompatibel dengan PostgreSQL

HariKrishna Boorgadda, Amazon Web Services

Ringkasan

Praktik terbaik untuk pemulihan bencana Enterprise (DR) pada dasarnya terdiri dari merancang dan menerapkan sistem perangkat keras dan perangkat lunak yang toleran terhadap kesalahan yang dapat bertahan dari bencana (kelanjutan bisnis) dan melanjutkan operasi normal (dimulainya kembali bisnis), dengan intervensi minimal dan, idealnya, tanpa kehilangan data. Membangun lingkungan yang toleran terhadap kesalahan untuk memenuhi tujuan Enterprise DR bisa mahal dan memakan waktu dan membutuhkan komitmen yang kuat dari bisnis.

Oracle Database menyediakan tiga pendekatan berbeda untuk DR yang memberikan tingkat perlindungan dan ketersediaan data tertinggi dibandingkan dengan pendekatan lain untuk melindungi data Oracle.

- Alat Pemulihan Kehilangan Data Oracle Zero
- Penjaga Data Aktif Oracle
- Oracle GoldenGate

Pola ini menyediakan cara untuk meniru Oracle GoldenGate DR dengan menggunakan database global Amazon Aurora. Arsitektur referensi menggunakan Oracle GoldenGate untuk DR di tiga Wilayah AWS. Pola ini berjalan melalui replatforming arsitektur sumber ke database global Aurora cloud-native berdasarkan Amazon Aurora PostgreSQL—Edisi yang kompatibel dengan Amazon Aurora.

Database global Aurora dirancang untuk aplikasi dengan jejak global. Basis data Aurora tunggal mencakup beberapa Wilayah AWS dengan sebanyak lima Wilayah sekunder. Database global Aurora menyediakan fitur-fitur berikut:

- Replikasi tingkat penyimpanan fisik
- Bacaan global latensi rendah
- Pemulihan bencana yang cepat dari pemadaman di seluruh wilayah
- Migrasi lintas wilayah yang cepat
- Kelambatan replikasi rendah di seluruh Wilayah

- Little-to-no dampak kinerja pada database Anda

Untuk informasi selengkapnya tentang fitur dan keunggulan database global Aurora, lihat Menggunakan database global Amazon Aurora. Untuk informasi selengkapnya tentang failover yang tidak direncanakan dan terkelola, lihat Menggunakan failover di database global Amazon Aurora.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Driver PostgreSQL Konektivitas Database Java (JDBC) untuk konektivitas aplikasi
- Database global Aurora berdasarkan Amazon Aurora PostgreSQL Edisi yang kompatibel
- Database Oracle Real Application Clusters (RAC) bermigrasi ke database global Aurora berdasarkan Aurora PostgreSQL yang kompatibel

Keterbatasan database global Aurora

- Database global Aurora tidak tersedia di semua Wilayah AWS. Untuk daftar Wilayah yang didukung, lihat database global Aurora dengan Aurora PostgreSQL.
- Untuk informasi tentang fitur yang tidak didukung dan batasan lain dari database global Aurora, lihat Batasan database global Amazon Aurora.

Versi produk

- Amazon Aurora PostgreSQL — Edisi yang kompatibel dengan versi 10.14 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database empat simpul Oracle RAC
- Oracle GoldenGate

Arsitektur sumber

Diagram berikut menunjukkan tiga cluster dengan empat node Oracle RAC di Wilayah AWS yang berbeda direplikasi menggunakan Oracle. GoldenGate

Tumpukan teknologi target

- Database global Amazon Aurora tiga cluster berdasarkan Aurora PostgreSQL — kompatibel, dengan satu cluster di Wilayah primer, dua cluster di Wilayah sekunder yang berbeda

Arsitektur target

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- Basis data global Amazon Aurora menjangkau beberapa Wilayah AWS, memberikan pembacaan global latensi rendah dan pemulihan cepat dari pemadaman langka yang mungkin memengaruhi seluruh Wilayah AWS.

Epik

Tambahkan Wilayah dengan instans DB pembaca

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pasang satu atau lebih cluster Aurora sekunder.	Di AWS Management Console, pilih Amazon Aurora. Pilih cluster utama, pilih Tindakan, dan pilih Tambah wilayah dari daftar dropdown.	DBA
Pilih kelas instance.	Anda dapat mengubah kelas instance dari cluster sekunder. Namun, kami sarankan untuk	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tetap sama dengan kelas instance cluster utama.	
Tambahkan wilayah ketiga.	Ulangi langkah-langkah dalam epik ini untuk menambahkan cluster di Wilayah ketiga.	DBA

Gagal atas database global Aurora

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus cluster utama dari database global Aurora.	1. Pada halaman Database, pilih cluster utama. 2. Pilih Hapus dari Global untuk gagal ke cluster sekunder.	DBA
Konfigurasi ulang aplikasi Anda untuk mengalihkan lalu lintas ke cluster yang baru dipromosikan.	Ubah titik akhir dalam aplikasi dengan cluster yang baru dipromosikan.	DBA
Berhenti mengeluarkan operasi penulisan apa pun ke cluster yang tidak tersedia.	Hentikan aplikasi dan aktivitas bahasa manipulasi data (DHTML) apa pun ke cluster yang Anda hapus.	DBA
Buat database global Aurora baru.	Sekarang Anda dapat membuat database global Aurora dengan cluster yang baru dipromosikan sebagai cluster utama.	DBA

Mulai cluster utama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih cluster utama yang akan dimulai dari database global.	Di konsol Amazon Aurora, dalam pengaturan Database Global, pilih klaster utama.	DBA
Mulai cluster.	Pada daftar dropdown Tindakan, pilih Mulai. Proses ini mungkin memakan waktu. Segarkan layar untuk melihat status, atau periksa kolom Status untuk status cluster saat ini setelah operasi selesai.	DBA

Bersihkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus cluster sekunder yang tersisa.	Setelah pilot failover selesai, hapus cluster sekunder dari database global.	DBA
Hapus cluster utama.	Hapus cluster.	DBA

Sumber daya terkait

- Menggunakan basis data global Amazon Aurora
- Solusi Pemulihan Bencana Aurora PostgreSQL menggunakan Amazon Aurora Global Database (posting blog)

Menerapkan SHA1 hashing untuk data PII saat bermigrasi dari SQL Server ke PostgreSQL

Rajkumar Raghuwanshi dan Jagadish Kantubugata, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerapkan hashing Secure Hash Algorithm 1 (SHA1) untuk alamat email saat bermigrasi dari SQL Server ke Amazon RDS for PostgreSQL atau yang kompatibel dengan Amazon Aurora PostgreSQL. Alamat email adalah contoh informasi identitas pribadi (PII). PII adalah informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar.

Pola ini mencakup tantangan mempertahankan nilai hash yang konsisten di berbagai kumpulan database dan pengkodean karakter, dan memberikan solusi menggunakan fungsi dan pemicu PostgreSQL. Meskipun pola ini berfokus pada SHA1 hashing, itu dapat disesuaikan untuk algoritma hashing lain yang didukung oleh modul PostgreSQL. `pgcrypto` Selalu pertimbangkan implikasi keamanan dari strategi hashing Anda dan konsultasikan dengan pakar keamanan jika menangani data sensitif.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Sumber database SQL Server
- Target database PostgreSQL (Amazon RDS untuk PostgreSQL atau Aurora PostgreSQL kompatibel)
- Keahlian pengkodean PL/PGSQL

Batasan

- Pola ini memerlukan perubahan pemeriksaan tingkat basis data berdasarkan kasus penggunaan.
- Dampak kinerja pada kumpulan data besar belum dievaluasi.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- Microsoft SQL Server 2012 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- SQL Server
- .NET Framework

Tumpukan teknologi target

- PostgreSQL
- `pgcrypto`perpanjangan

Otomatisasi dan skala

- Pertimbangkan untuk menerapkan fungsi hashing sebagai prosedur tersimpan untuk perawatan yang lebih mudah.
- Untuk kumpulan data besar, evaluasi kinerja dan pertimbangkan pemrosesan batch atau strategi pengindeksan.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL kompatibel adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke dalam AWS Cloud atau di antara kombinasi pengaturan cloud dan lokal.
- Amazon Relational Database Service Amazon RDS for PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di. AWS Cloud
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Alat lainnya

- pgAdmin adalah alat manajemen open source untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.
- SQL Server Management Studio (SSMS) adalah lingkungan terintegrasi untuk mengelola infrastruktur SQL apa pun.

Praktik terbaik

- Gunakan pengaturan pemeriksaan yang sesuai untuk menangani karakter khusus di sisi database target.
- Uji secara menyeluruh dengan berbagai alamat email, termasuk alamat dengan karakter non-ASCII.
- Pertahankan konsistensi dalam penanganan huruf besar dan kecil antara lapisan aplikasi dan database.
- Benchmark kinerja kueri menggunakan nilai hash.

Epik

Menganalisis implementasi hashing sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau kode SQL Server.	Untuk meninjau kode SQL Server yang menghasilkan SHA1 hash, lakukan hal berikut: • Menganalisis implementasi SQL Server yang ada dari SHA1 hashing. • Identifikasi metode yang tepat yang digunakan untuk pembuatan hash. • Dokumentasikan parameter input dan format output.	Insinyur data, DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Tinjau konversi atau transformasi tipe data apa pun. • Periksa pengatura n pemeriksaan dan dampaknya.	
Dokumentasikan algoritma hashing dan transformasi data.	Untuk mendokumentasikan algoritma hashing dan transformasi data yang tepat, lakukan hal berikut: • Buat dokumentasi teknis terperinci dari proses hashing. • Dokumentasikan logika step-by-step transformasi. • Tentukan format input dan output serta tipe data. • Sertakan kasus tepi dan penanganan karakter khusus.	Pengembang aplikasi, Insinyur data, DBA

Buat fungsi hashing PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat `pgcrypto` ekstensi.	Untuk membuat `pgcrypto` ekstensi, gunakan `pgAdmin/psql` untuk menjalankan perintah berikut: `CREATE EXTENSION pgcrypto;`	DBA, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan fungsi PostgreSQL.	Menerapkan fungsi PostgreSQL berikut untuk mereplikasi logika hashing SQL Server. Pada tingkat tinggi, fungsi ini menggunakan langkah-langkah berikut: 1. Secara opsional mengubah input menjadi huruf besar. 2. Menciptakan SHA1 hash dari input. 3. Mengambil 10 byte terakhir (80 bit) dari hash ini. 4. Mengkonversi byte ini ke integer 64-bit. `CREATE OR REPLACE FUNCTION utility.hex_to_bigint ( par_val character varying, par_upper character varying DEFAULT 'lower'::character varying) RETURNS bigint LANGUAGE 'plpgsql' AS $BODY$ DECLARE retnumber bigint; digest_bytes bytea; BEGIN if lower(par_upper) = 'upper' then`	Insinyur data, DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`digest_bytes := digest(upper(par_val), 'sha1'); else digest_bytes := digest((par_val), 'sha1'); end if; retnumber := ('x' || encode(substring(digest_bytes, length(digest_bytes)-10+1), 'hex'))::bit(64)::bigint; RETURN retnumber; END; $BODY$;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji fungsinya.	Untuk menguji fungsi, gunakan data sampel dari SQL Server untuk memverifi kasi nilai hash yang cocok. Jalankan perintah berikut:
	`select 'alejandr o_rosalez@example. com' as Email, utility.hex_to_big int('alejandro_ros alez@example.com', 'upper') as HashValue;`
	`--OUTPUT /* email hashvalue "alejandro_rosale z@example.com" 451 397011176045063 */`
		Pengembang aplikasi, DBA, Insinyur data

Menerapkan pemicu untuk hashing otomatis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pemicu pada tabel yang relevan.	Untuk membuat pemicu pada tabel yang relevan untuk secara otomatis menghasil kan nilai hash pada insert atau update, jalankan perintah berikut: `CREATE OR REPLACE FUNCTION update_em ail_hash()`	Pengembang aplikasi, Insinyur data, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`RETURNS TRIGGER AS $$ BEGIN NEW.email_hash = utility.hex_to_bigint(NEW.email, 'upper'); RETURN NEW; END; $$ LANGUAGE plpgsql;`
	
	`CREATE TRIGGER email_hash_trigger BEFORE INSERT OR UPDATE ON users FOR EACH ROW EXECUTE FUNCTION update_email_hash();`	

Migrasi data yang ada

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembangkan skrip migrasi atau gunakan AWS DMS.	Kembangkan skrip migrasi atau gunakan AWS DMS untuk mengisi nilai hash untuk data yang ada (termasuk nilai hash yang disimpan seperti `BIGINT` dalam sistem sumber.) Lakukan hal-hal berikut: • Buat skrip migrasi untuk transfer data dengan nilai hash.	Insinyur data, Pengembang aplikasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Konfigurasikan AWS DMS tugas dengan aturan transformasi yang sesuai. • Siapkan titik akhir sumber dan target di AWS DMS. • Menerapkan penanganan kesalahan dan mekanisme logging. • Rancang strategi pemrosesan batch untuk kumpulan data besar. • Buat kueri validasi untuk verifikasi data.	
Gunakan fungsi hashing PostgreSQL yang baru.	Untuk menggunakan fungsi hashing PostgreSQL baru untuk memastikan konsistensi, lakukan hal berikut: • Menerapkan prosedur validasi untuk memverifikasi konsistensi hash. • Buat skrip perbandingan antara sumber dan sistem target. • Siapkan pengujian otomatis untuk verifikasi nilai hash. • Dokumentasikan setiap perbedaan dan langkah resolusi.	Pengembang aplikasi, DBA, DevOps insinyur

Perbarui kueri aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi kueri aplikasi.	Untuk mengidentifikasi kueri aplikasi yang menggunakan nilai hash, lakukan hal berikut: • Analisis basis kode aplikasi untuk kueri menggunakan nilai hash. • Tinjau prosedur dan fungsi tersimpan yang mereferen sikan operasi hash. • Metrik kinerja kueri dokumen dan rencana eksekusi. • Identifikasi dependensi pada pencarian berbasis hash. • Memetakan komponen aplikasi yang terpengaruh.	Pengembang aplikasi, DBA, Insinyur data
Ubah kueri.	Jika perlu, ubah kueri untuk menggunakan fungsi hashing PostgreSQL yang baru. Lakukan hal-hal berikut: • Memfaktorkan ulang kueri yang ada untuk menggunakan fungsi hashing PostgreSQL. • Perbarui prosedur dan fungsi yang tersimpan. • Menerapkan dan menguji pola kueri baru.	Pengembang aplikasi, DBA, Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Optimalkan kueri yang dimodifikasi untuk kinerja.	

Uji dan validasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan pengujian.	Untuk melakukan pengujian menyeluruh dengan subset data produksi, lakukan hal berikut: • Buat rencana pengujian untuk validasi subset data. • Ekstrak sampel representatif dari data produksi. • Siapkan lingkungan pengujian dengan konfigura si yang tepat. • Jalankan pemuatan data dan tes transformasi. • Lakukan pengujian volume dan stress.	Pengembang aplikasi, Insinyur data, DBA
Validasi bahwa nilai hash cocok.	Untuk memvalidasi bahwa nilai hash cocok antara SQL Server dan PostgreSQL, lakukan hal berikut: • Kembangkan skrip perbandingan untuk nilai hash. • Buat laporan validasi untuk pencocokan hash.	Pengembang aplikasi, Insinyur data, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Menerapkan prosedur verifikasi otomatis. • Dokumentasikan setiap perbedaan yang ditemukan. • Menganalisis dan menyelesaikan ketidakco cokan hash.	
Verifikasi fungsionalitas aplikasi.	Untuk memverifikasi fungsiona litas aplikasi dengan menggunakan data yang dimigrasi dan implementasi hashing baru, lakukan hal berikut: • Jalankan pengujian end-to-end aplikasi. • Validasi semua fitur aplikasi menggunakan data hash. • Uji kinerja aplikasi dengan implementasi baru. • Verifikasi integrasi dan dependensi API.	Pengembang aplikasi, DBA, Insinyur data

Pemecahan masalah

Isu	Solusi
Nilai hash tidak cocok.	Verifikasi pengkodean karakter dan susunan antara sumber dan target. Untuk informasi selengkapnya, lihat Mengelola perubahan pemeriksaan di PostgreSQL di Amazon Aurora dan Amazon RDS (Blog).AWS

Sumber daya terkait

AWS Blog

- Kelola perubahan pemeriksaan di PostgreSQL di Amazon Aurora dan Amazon RDS
- Migrasikan SQL Server ke Amazon Aurora PostgreSQL menggunakan praktik terbaik dan pelajaran yang dipetik dari lapangan

Sumber daya lainnya

- PostgreSQL modul pgcrypto (dokumentasi PostgreSQL)
- Fungsi pemicu PostgreSQL (dokumentasi PostgreSQL)
- Fungsi SQL Server HASHBYTES (dokumentasi Microsoft)

Bermigrasi secara bertahap dari Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL menggunakan Oracle SQL Developer dan AWS SCT

Pinesh Singal, Amazon Web Services

Ringkasan

Banyak strategi dan pendekatan migrasi berjalan dalam beberapa fase yang dapat berlangsung dari beberapa minggu hingga beberapa bulan. Selama waktu ini, Anda dapat mengalami penundaan karena penambalan atau peningkatan dalam instans DB Oracle sumber yang ingin Anda migrasi ke instans PostgreSQL DB. Untuk menghindari situasi ini, kami menyarankan Anda secara bertahap memigrasikan kode database Oracle yang tersisa ke kode database PostgreSQL.

Pola ini menyediakan strategi migrasi inkremental tanpa downtime untuk instans Oracle DB multi-terabyte yang memiliki jumlah transaksi tinggi yang dilakukan setelah migrasi awal Anda dan yang harus dimigrasikan ke database PostgreSQL. Anda dapat menggunakan step-by-step pendekatan pola ini untuk memigrasikan Amazon Relational Database Service (Amazon RDS) untuk instans Oracle DB secara bertahap ke instans Amazon RDS for PostgreSQL DB tanpa masuk ke Konsol Manajemen Amazon Web Services (AWS).

Pola ini menggunakan Oracle SQL Developer untuk menemukan perbedaan antara dua skema dalam database sumber Oracle. Anda kemudian menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi objek skema database Amazon RDS for Oracle ke Amazon RDS untuk objek skema database Amazon RDS for PostgreSQL. Anda kemudian dapat menjalankan skrip Python di Windows Command Prompt untuk membuat objek AWS SCT untuk perubahan tambahan pada objek database sumber.

> **Note**
>
> Sebelum memigrasikan beban kerja produksi, sebaiknya jalankan proof of concept (PoC) untuk pendekatan pola ini di lingkungan pengujian atau non-produksi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Instans Amazon RDS for Oracle DB yang ada.

- Amazon RDS yang ada untuk instans DB Amazon RDS for PostGresQL.
- AWS SCT, diinstal dan dikonfigurasi dengan driver JDBC untuk mesin database Oracle dan PostgreSQL. Untuk informasi selengkapnya tentang ini, lihat Menginstal AWS SCT dan Menginstal driver database yang diperlukan dalam dokumentasi AWS SCT.
- Oracle SQL Developer, diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang ini, lihat dokumentasi Oracle SQL Developer.
- `incremental-migration-sct-sql.zip`File (terlampir), diunduh ke komputer lokal Anda.

Batasan

- Persyaratan minimum untuk instans Amazon RDS for Oracle DB sumber Anda adalah:
  - Oracle versi 10.2 dan yang lebih baru (untuk versi 10.x), 11g (versi 11.2.0.3.v1 dan yang lebih baru) dan hingga 12.2, dan 18c untuk edisi Enterprise, Standard, Standard One, dan Standard Two
- Persyaratan minimum untuk instans Amazon RDS for PostgreSQL DB target Anda adalah:
  - PostgreSQL versi 9.4 dan yang lebih baru (untuk versi 9.x), 10.x, dan 11.x
- Pola ini menggunakan Oracle SQL Developer. Hasil Anda mungkin berbeda jika Anda menggunakan alat lain untuk menemukan dan mengekspor perbedaan skema.
- Skrip SQL yang dihasilkan oleh Oracle SQL Developer dapat meningkatkan kesalahan transformasi, yang berarti Anda perlu melakukan migrasi manual.
- Jika koneksi pengujian sumber dan target AWS SCT gagal, pastikan Anda mengonfigurasi versi driver JDBC dan aturan masuk untuk grup keamanan virtual private cloud (VPC) untuk menerima lalu lintas masuk.

Versi produk

- Amazon RDS for Oracle DB instans versi 12.1.0.2 (versi 10.2 dan yang lebih baru)
- Amazon RDS for PostgreSQL DB instans versi 11.5 (versi 9.4 dan yang lebih baru)
- Oracle SQL Developer versi 19.1 dan yang lebih baru
- AWS SCT versi 1.0.632 dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Amazon RDS for Oracle DB instans

Tumpukan teknologi target

- Amazon RDS untuk instans DB Amazon RDS for PostgreSQL

Arsitektur sumber dan target

Diagram berikut menunjukkan migrasi instans Amazon RDS for Oracle DB ke instans Amazon RDS for PostgreSQL DB.

Diagram menunjukkan alur kerja migrasi berikut:

1. Buka Oracle SQL Developer dan sambungkan ke database sumber dan target.
2. Hasilkan laporan diff dan kemudian buat file skrip SQL untuk objek perbedaan skema. Untuk informasi selengkapnya tentang laporan diff, lihat Laporan diff terperinci dalam dokumentasi Oracle.
3. Konfigurasikan AWS SCT dan jalankan kode Python.
4. File skrip SQL mengkonversi dari Oracle ke PostgreSQL.
5. Jalankan file skrip SQL pada instance PostgreSQL DB target.

Otomatisasi dan skala

Anda dapat mengotomatiskan migrasi ini dengan menambahkan parameter tambahan dan perubahan terkait keamanan untuk beberapa fungsi dalam satu program ke skrip Python Anda.

Alat

- AWS SCT — AWS Schema Conversion Tool (AWS SCT) mengonversi skema database Anda yang ada dari satu mesin database ke mesin database lainnya.
- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi (IDE) yang menyederhanakan pengembangan dan pengelolaan database Oracle baik dalam penerapan tradisional maupun berbasis cloud.

Kode

`incremental-migration-sct-sql.zip`File (terlampir) berisi kode sumber lengkap untuk pola ini.

Epik

Buat file skrip SQL untuk perbedaan skema basis data sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan Database Diff di Oracle SQL Developer.	1. Masuk ke instans Oracle DB sumber Anda, pilih Tools, lalu pilih Database Diff. 2. Pilih database sumber Anda di Source Connection. 3. Pilih database sumber yang diperbarui atau ditambal di Koneksi Tujuan. 4. Konfigurasikan opsi yang tersisa sesuai dengan kebutuhan Anda, pilih Berikutnya, lalu pilih Selesai untuk menghasilkan laporan diff.	DBA
Hasilkan file skrip SQL.	Pilih Generate Script untuk menghasilkan perbedaan dalam file SQL. Ini menghasilkan file skrip SQL yang digunakan AWS SCT untuk mengonversi database Anda dari Oracle ke PostgreSQL.	DBA

Gunakan skrip Python untuk membuat objek DB target di AWS SCT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan AWS SCT dengan Windows Command Prompt.	1. Salin `AWSSchema ConversionToolBatch.jar` file dari folder AWS SCT yang sudah diinstal sebelumnya dan tempel ke direktori kerja Anda. 2. Menyebarkan kode Python dari file dari `run_aws_sct_sql.py` folder (`incremental-migration-sct-sql.zip` terlampir). Ini membuat file.xl dan file.sct di `projects` direktori dengan detail konfigurasi lingkungan basis data sumber dan target Anda. Ini juga membaca file skrip SQL yang Anda hasilkan di Oracle SQL Developer . Akhirnya, ia menciptakan objek file.sql dalam direktori . `output` 3. Konfigurasikan detail konfigurasi lingkungan sumber dan target dalam `database_migration.txt` file dengan menggunakan format berikut:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`#source_vendor,sou rce_hostname,sourc e_dbname,source_us er,source_pwd,sour ce_schema,source_p ort,source_sid,tar get_vendor,target_ hostname,target_us er,target_pwd,targ et_dbname,target_port`
	
	`ORACLE,myoracledb.cokm vis0v46q.us-east-1 .rds.amazonaws.com ,ORCL,orcl,orcl123 4,orcl,1521,ORCL,P OSTGRESQL,mypgdbin stance.cokmvis0v46 q.us-east-1.rds.am azonaws.com,pguser ,pgpassword,pgdb,5432`
	
	4. Ubah parameter konfigura si AWS SCT sesuai dengan kebutuhan Anda dan kemudian salin file skrip SQL ke direktori kerja Anda di subdirektori. `input`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip Python.	1. Jalankan skrip Python dengan menggunakan perintah berikut: `$ python run_aws_sct_sql.py database_migration.txt` 2. Ini menciptakan file SQL objek DB. Kode yang tidak dikonversi dengan kesalahan transformasi dapat dikonversi secara manual.	DBA
Buat objek di Amazon RDS untuk PostgreSQL	Jalankan file SQL dan buat objek di Amazon RDS for PostgreSQL DB instance.	DBA

Sumber daya terkait

- Oracle di Amazon RDS
- PostgreSQL di Amazon RDS
- Menggunakan antarmuka pengguna AWS SCT
- Menggunakan Oracle sebagai sumber AWS SCT

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Muat file BLOB ke TEXT dengan menggunakan pengkodean file di Aurora PostgreSQL yang kompatibel

Bhanu Ganesh Gudivada dan Jeevan Shetty, Amazon Web Services

Ringkasan

Seringkali selama migrasi, ada kasus di mana Anda harus memproses data tidak terstruktur dan terstruktur yang dimuat dari file pada sistem file lokal. Data mungkin juga berada dalam kumpulan karakter yang berbeda dari kumpulan karakter database.

File-file ini menyimpan jenis data berikut:

- Metadata — Data ini menjelaskan struktur file.
- Data semi-terstruktur — Ini adalah string tekstual dalam format tertentu, seperti JSON atau XHTML. Anda mungkin dapat membuat pernyataan tentang data tersebut, seperti “akan selalu dimulai dengan '<'” atau “tidak mengandung karakter baris baru.”
- Teks lengkap — Data ini biasanya berisi semua jenis karakter, termasuk karakter baris baru dan kutipan. Mungkin juga terdiri dari karakter multibyte di UTF-8.
- Data biner — Data ini mungkin berisi byte atau kombinasi byte termasuk, nol dan penanda. end-of-file

Memuat campuran jenis data ini bisa menjadi tantangan.

Pola ini dapat digunakan dengan database Oracle lokal, database Oracle yang ada di instans Amazon Elastic Compute Cloud (Amazon) di Amazon Web Services (AWS EC2) Cloud, dan Amazon Relational Database Service (Amazon RDS) untuk database Oracle. Sebagai contoh, pola ini menggunakan Amazon Aurora PostgreSQL Compatible Edition.

Di Oracle Database, dengan bantuan pointer `BFILE` (file biner), `DBMS_LOB` paket, dan fungsi sistem Oracle, Anda dapat memuat dari file dan mengkonversi ke CLOB dengan pengkodean karakter. Karena PostgreSQL tidak mendukung tipe data BLOB saat bermigrasi ke database Amazon Aurora PostgreSQL Edisi yang kompatibel dengan PostgreSQL, fungsi ini harus dikonversi ke skrip yang kompatibel dengan PostgreSQL.

Pola ini menyediakan dua pendekatan untuk memuat file ke dalam kolom database tunggal dalam database yang kompatibel dengan Amazon Aurora PostgreSQL:

- Pendekatan 1 - Anda mengimpor data dari bucket Amazon Simple Storage Service (Amazon S3) Simple Storage S3) dengan menggunakan `table_import_from_s3` fungsi ekstensi dengan `aws_s3` opsi encode.
- Pendekatan 2 — Anda menyandikan heksadesimal di luar database, dan kemudian Anda memecahkan kode untuk melihat di dalam database. `TEXT`

Sebaiknya gunakan Pendekatan 1 karena Aurora PostgreSQL kompatibel memiliki integrasi langsung dengan ekstensi. `aws_s3`

Pola ini menggunakan contoh memuat file datar yang berisi template email, yang memiliki karakter multibyte dan format berbeda, ke dalam database Amazon Aurora PostgreSQL yang kompatibel.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Instans Amazon RDS atau instans yang kompatibel dengan Aurora PostgreSQL
- Pemahaman dasar SQL dan Relational Database Management System (RDBMS)
- Bucket Amazon Simple Storage Service (Amazon S3).
- Pengetahuan tentang fungsi sistem di Oracle dan PostgreSQL
- RPM Package HexDump -XXD-0.1.1 (disertakan dengan Amazon Linux 2)

> **Note**
>
> Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs.

Batasan

- Untuk tipe `TEXT` data, string karakter terpanjang yang dapat disimpan adalah sekitar 1 GB.

Versi produk

- Aurora mendukung versi PostgreSQL yang tercantum dalam pembaruan Amazon Aurora PostgreSQL.

Arsitektur

Tumpukan teknologi target

- Kompatibel dengan Aurora PostgreSQL

Arsitektur target

Pendekatan 1 - Menggunakan aws_s3.table_import_from_s3

Dari server lokal, file yang berisi templat email dengan karakter multibyte dan pemformatan kustom ditransfer ke Amazon S3. Fungsi database kustom yang disediakan oleh pola ini menggunakan `aws_s3.table_import_from_s3` fungsi dengan `file_encoding` untuk memuat file ke dalam database dan mengembalikan hasil query sebagai tipe `TEXT` data.

1. File ditransfer ke bucket S3 pementasan.
2. File diunggah ke database yang kompatibel dengan Amazon Aurora PostgreSQL.
3. Menggunakan klien pgAdmin, `load_file_into_clob` fungsi kustom diterapkan ke database Aurora.
4. Fungsi kustom digunakan secara internal `table_import_from_s3` dengan file_encoding. Output dari fungsi diperoleh dengan menggunakan `array_to_string` dan `array_agg` sebagai `TEXT` output.

Pendekatan 2 - Pengkodean ke heksadesimal di luar database dan decoding untuk melihat TEXT di dalam database

File dari server lokal atau sistem file lokal diubah menjadi hex dump. Kemudian file tersebut diimpor ke PostgreSQL sebagai bidang. `TEXT`

1. Konversi file ke hex dump di baris perintah dengan menggunakan opsi. `xxd -p`
2. Unggah file hex dump ke Aurora PostgreSQL yang kompatibel dengan menggunakan `\copy` opsi, dan kemudian memecahkan kode file hex dump ke biner.
3. Mengkodekan data biner untuk kembali sebagai`TEXT`.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.

Alat lainnya

- pGADmin4 adalah platform administrasi dan pengembangan open source untuk PostgreSQL. pGADmin4 dapat digunakan di Linux, Unix, mac OS, dan Windows untuk mengelola PostgreSQL.

Epik

Pendekatan 1: Impor data dari Amazon S3 ke Aurora PostgreSQL yang kompatibel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan sebuah EC2 instance.	Untuk petunjuk tentang meluncurkan instance, lihat Meluncurkan instance Anda.	DBA
Instal alat pgAdmin klien PostgreSQL.	Unduh dan instal pgAdmin.	DBA
Buat kebijakan IAM.	Buat kebijakan AWS Identity and Access Managemen t (IAM) bernama `aurora-s3-access-pol` yang memberikan akses ke bucket S3 tempat file akan disimpan. Gunakan kode berikut, ganti `<bucket-name>` dengan nama bucket S3 Anda.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	{
	"Version":
	"2012-10-17",
	"Statement": [
	{
	"Effect":
	"Allow",
	"Action": [
	"s3:GetObject",
	"s3:AbortMultipart
	Upload",
	"s3:DeleteObject",
	"s3:ListMultipartU
	ploadParts",
	"s3:PutObject",
	"s3:ListBucket"
	],
	"Resource":
	[
	"arn:aws:s3:::<buc
	ket-name>/*",
	"arn:aws:s3:::<buc
	ket-name>"
	]
	}
	]
	}	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk impor objek dari Amazon S3 ke Aurora PostgreSQL yang kompatibel.	Gunakan kode berikut untuk membuat peran IAM bernama `aurora-s3-import-role` dengan hubungan AssumeRolekepercayaan. `AssumeRole` memungkinkan Aurora mengakses layanan AWS lainnya atas nama Anda. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow","Principal": { "Service": "rds.amazonaws.com" },"Action": "sts:AssumeRole" } ] }`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kaitkan peran IAM ke cluster.	Untuk mengaitkan peran IAM dengan cluster database yang kompatibel dengan Aurora PostgreSQL, jalankan perintah AWS CLI berikut. Ubah `<Account-ID>` ke ID akun AWS yang menghosting database yang kompatibel dengan Aurora PostgreSQL. Ini memungkinkan database yang kompatibel dengan Aurora PostgreSQL untuk mengakses bucket S3. `aws rds add-role-to-db-cluster --db-cluster-identifier aurora-postgres-cl --feature-name s3Import --role-arn arn:aws:iam::<Account-ID>:role/aurora-s3-import-role`	DBA
Unggah contoh ke Amazon S3.	1. Di bagian Informasi tambahan dari pola ini, salin kode template email ke dalam file bernama `salary.event.notification.email.vm` . 2. Unggah ke file ke bucket S3.	DBA, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan fungsi kustom.	1. Dari bagian Informasi tambahan, salin fungsi kustom konten file `load_file_into_clob` SQL ke dalam tabel sementara. 2. Masuk ke database Aurora PostgreSQL yang kompatibel dan terapkan ke dalam skema database dengan menggunakan klien pgAdmin.	Pemilik aplikasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan fungsi kustom untuk mengimpor data ke dalam database.	Jalankan perintah SQL berikut, ganti item dalam kurung sudut dengan nilai yang sesuai.
	`select load_file _into_clob('aws-s3 -import-test'::tex t,'us-west-1'::tex t,'employee.salary .event.notificatio n.email.vm'::text);`
	Ganti item dalam kurung sudut dengan nilai yang sesuai, seperti yang ditunjukkan pada contoh berikut, sebelum menjalankan perintah.
	`Select load_file _into_clob('aws-s3 -import-test'::tex t,'us-west-1'::tex t,'employee.salary .event.notificatio n.email.vm'::text);`
	Perintah memuat file dari Amazon S3 dan mengembal ikan output sebagai file. TEXT
		Pemilik aplikasi, DBA

Pendekatan 2: Ubah file template menjadi hex dump di sistem Linux lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah file template menjadi hex dump.	**Note** Utilitas Hexdump menampilkan isi file biner dalam heksadesimal, desimal, oktal, atau ASCII. `hexdump`Perintah adalah bagian dari `util-linux` paket dan sudah diinstal sebelumnya di distribusi Linux. Paket Hexdump RPM adalah bagian dari Amazon Linux 2 juga. (: Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs.) Untuk mengonversi isi file menjadi hex dump, jalankan perintah shell berikut. `xxd -p </path/file.vm> | tr -d '\n' > </path/file.hex>`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Ganti path dan file dengan nilai yang sesuai, seperti yang ditunjukkan pada contoh berikut. `xxd -p employee.salary.event.notification.email.vm | tr -d '\n' > employee.salary.event.notification.email.vm.hex`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Muat file hexdump ke dalam skema database.	Gunakan perintah berikut untuk memuat file hexdump ke dalam database Aurora PostgreSQL yang kompatibel.
	1. Masuk ke database Aurora PostgreSQL, dan buat tabel baru yang disebut. `email_template_hex`
	`CREATE TABLE email_template_hex(hex_data TEXT);`
	2. Muat file dari sistem file lokal ke dalam skema DB dengan menggunakan perintah berikut.
	`\copy email_template_hex FROM '/path/file.hex';`
	Ganti path dengan lokasi pada sistem file lokal Anda.
	`\copy email_template_hex FROM '/tmp/employee.salary.event.notification.email.vm.hex';`
	3. Buat satu tabel lagi yang disebut`email_template_bytea` .
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`CREATE TABLE email_template_bytea(hex_data bytea);`
	4. Masukkan data dari `email_template_hex` ke dalam `email_template_bytea` .
	`INSERT INTO email_template_bytea (hex_data)`
	`(SELECT decode(hex_data, 'hex') FROM email_template_hex limit 1);`
	5. Untuk mengembalikan kode hex bytea sebagai TEXT data, jalankan perintah berikut.
	`SELECT encode(hex_data::bytea, 'escape') FROM email_template_bytea;`	

Sumber daya terkait

Referensi

- Menggunakan database PostgreSQL sebagai target AWS Database Migration Service
- Oracle Database 19c ke Amazon Aurora dengan PostgreSQL Kompatibilitas (12.4) Playbook Migrasi

- Membuat kebijakan IAM
- Mengaitkan peran IAM dengan kluster DB MySQL Amazon Aurora
- pgAdmin

Tutorial

- Memulai dengan Amazon RDS
- Bermigrasi dari Oracle ke Amazon Aurora

Informasi tambahan

load_file_into_clob fungsi kustom

```
CREATE OR REPLACE FUNCTION load_file_into_clob(
    s3_bucket_name text,
    s3_bucket_region text,
    file_name text,
    file_delimiter character DEFAULT '&'::bpchar,
    file_encoding text DEFAULT 'UTF8'::text)
    RETURNS text
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
AS $BODY$
DECLARE
    blob_data BYTEA;
    clob_data TEXT;
    l_table_name CHARACTER VARYING(50) := 'file_upload_hex';
    l_column_name CHARACTER VARYING(50) := 'template';
    l_return_text TEXT;
    l_option_text CHARACTER VARYING(150);
    l_sql_stmt CHARACTER VARYING(500);

BEGIN

    EXECUTE format ('CREATE TEMPORARY TABLE %I (%I text, id_serial serial)',
 l_table_name, l_column_name);

    l_sql_stmt := 'select ''(format text, delimiter ''''' || file_delimiter || ''''',
 encoding ''''' || file_encoding ||  ''''')'' ';
```

```
    EXECUTE FORMAT(l_sql_stmt)
    INTO l_option_text;

    EXECUTE FORMAT('SELECT aws_s3.table_import_from_s3($1,$2,$6,
 aws_commons.create_s3_uri($3,$4,$5))')
    INTO l_return_text
    USING l_table_name, l_column_name, s3_bucket_name,
 file_name,s3_bucket_region,l_option_text;

    EXECUTE format('select array_to_string(array_agg(%I order by id_serial),E''\n'')
 from %I', l_column_name, l_table_name)
    INTO clob_data;

    drop table file_upload_hex;

    RETURN clob_data;
END;
$BODY$;
```

Template email

```
##############################################################################################
##
 ##
##    johndoe Template Type: email
 ##
##    File: johndoe.salary.event.notification.email.vm
 ##
##    Author: Aimée Étienne    Date 1/10/2021
      ##
##  Purpose: Email template used by EmplmanagerEJB to inform a johndoe they         ##
##        have been given access to a salary event
 ##
##    Template Attributes:
 ##
##        invitedUser - PersonDetails object for the invited user
  ##
##        salaryEvent - OfferDetails object for the event the user was given access
 ##
##        buyercollege - CompDetails object for the college owning the salary event
 ##
```

```
##        salaryCoordinator - PersonDetails of the salary coordinator for the event
 ##
##        idp - Identity Provider of the email recipient
 ##
##        httpWebRoot - HTTP address of the server
 ##
##
 ##
################################################################################

$!invitedUser.firstname $!invitedUser.lastname,

Ce courriel confirme que vous avez ete invite par $!salaryCoordinator.firstname $!
salaryCoordinator.lastname de $buyercollege.collegeName a participer a l'evenement
 "$salaryEvent.offeringtitle" sur johndoeMaster Sourcing Intelligence.

Votre nom d'utilisateur est $!invitedUser.username

Veuillez suivre le lien ci-dessous pour acceder a l'evenement.

${httpWebRoot}/myDashboard.do?idp=$!{idp}

Si vous avez oublie votre mot de passe, utilisez le lien "Mot de passe oublie" situe
 sur l'ecran de connexion et entrez votre nom d'utilisateur ci-dessus.

Si vous avez des questions ou des preoccupations, nous vous invitons a
 communiquer avec le coordonnateur de l'evenement $!salaryCoordinator.firstname $!
salaryCoordinator.lastname au ${salaryCoordinator.workphone}.

*******

johndoeMaster Sourcing Intelligence est une plateforme de soumission en ligne pour les
 equipements, les materiaux et les services.

Si vous avez des difficultes ou des questions, envoyez un courriel a
 support@johndoeMaster.com pour obtenir de l'aide.
```

Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dengan dan menggunakan dan AWS SCT AWS DMS AWS CLI CloudFormation

Pinesh Singal, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memigrasikan Amazon Relational Database Service (Amazon RDS) multi-terabyte untuk instans Oracle DB ke instans Amazon RDS for PostgreSQL DB dengan menggunakan (). AWS Command Line Interface AWS CLI Pendekatan ini memberikan waktu henti minimal dan tidak perlu masuk ke. Konsol Manajemen AWS

Pola ini membantu menghindari konfigurasi manual dan migrasi individual dengan menggunakan konsol AWS Schema Conversion Tool (AWS SCT) dan AWS Database Migration Service (AWS DMS). Solusinya menyiapkan konfigurasi satu kali untuk beberapa database dan melakukan migrasi dengan menggunakan AWS SCT dan AWS DMS di file. AWS CLI

Pola ini digunakan AWS SCT untuk mengonversi objek skema database dari Amazon RDS for Oracle ke Amazon RDS for PostgreSQL, dan kemudian digunakan untuk memigrasikan data. AWS DMS Menggunakan skrip Python di AWS CLI, Anda membuat AWS SCT objek dan AWS DMS tugas dengan template. CloudFormation

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Instans Amazon RDS for Oracle DB yang ada.
- Amazon RDS yang ada untuk instans DB Amazon RDS for PostgreSQL.
- Instans Amazon Elastic Compute Cloud (Amazon EC2) atau mesin lokal dengan OS Windows atau Linux untuk menjalankan skrip.
- Pemahaman tentang jenis tugas AWS DMS migrasi berikut:`full-load`,`cdc`,`full-load-and-cdc`. Untuk informasi selengkapnya, lihat Membuat tugas dalam AWS DMS dokumentasi.
- AWS SCT, diinstal dan dikonfigurasi dengan driver Java Database Connectivity (JDBC) untuk mesin database Oracle dan PostgreSQL. Untuk informasi selengkapnya, lihat Menginstal dan mengonfigurasi AWS SCT dalam AWS SCT dokumentasi.
- `AWSSchemaConversionToolBatch.jar`File dari AWS SCT folder yang diinstal, disalin ke direktori kerja Anda.

- `cli-sct-dms-cft.zip`File (terlampir), diunduh dan diekstraksi di direktori kerja Anda.
- Versi mesin contoh AWS DMS replikasi terbaru. Untuk informasi selengkapnya, lihat Bagaimana cara membuat instance AWS DMS replikasi dalam AWS Dukungan dokumentasi dan catatan AWS DMS rilis.
- AWS CLI versi 2, diinstal dan dikonfigurasi dengan ID kunci akses Anda, kunci akses rahasia, dan Wilayah AWS nama default untuk EC2 instance atau OS tempat skrip dijalankan. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui ke versi terbaru AWS CLI dan Mengkonfigurasi pengaturan untuk AWS CLI dokumentasi. AWS CLI
- Keakraban dengan CloudFormation template. Untuk informasi selengkapnya, lihat Cara CloudFormation kerja dalam CloudFormation dokumentasi.
- Python versi 3, diinstal dan dikonfigurasi pada EC2 instance atau OS tempat skrip dijalankan. Untuk informasi selengkapnya, lihat dokumentasi Python.

Batasan

- Persyaratan minimum untuk instans Amazon RDS for Oracle DB sumber Anda adalah:
  - Oracle versi 12c (12.1.0.2, 12.2.0.1), 18c (18.0.0.0), dan 19c (19.0.0.0) untuk edisi Enterprise, Standard, Standard One, dan Standard Two.
  - Meskipun Amazon RDS mendukung Oracle 18c (18.0.0.0), versi ini berada di jalur penghentian karena Oracle tidak lagi menyediakan tambalan untuk 18c setelah tanggal. end-of-support Untuk informasi selengkapnya, lihat Amazon RDS for Oracle di dokumentasi Amazon RDS.
  - Amazon RDS for Oracle 11g tidak lagi didukung.
- Persyaratan minimum untuk instans Amazon RDS for PostgreSQL DB target Anda adalah:
  - PostgreSQL versi 9 (9.5 dan 9.6), 10.x, 11.x, 12.x, dan 13.x

Versi produk

- Amazon RDS for Oracle DB instans versi 12.1.0.2 dan yang lebih baru
- Amazon RDS untuk instans PostgreSQL DB versi 11.5 dan yang lebih baru
- AWS CLI versi 2
- Versi terbaru dari AWS SCT
- Versi terbaru Python 3

Arsitektur

Tumpukan teknologi sumber

- Amazon RDS for Oracle

Tumpukan teknologi target

- Amazon RDS for PostgreSQL

Arsitektur sumber dan target

Diagram berikut menunjukkan migrasi instans Amazon RDS for Oracle DB ke instans Amazon RDS for PostgreSQL DB menggunakan skrip Python. AWS DMS

Diagram menunjukkan alur kerja migrasi berikut:

1. Skrip Python digunakan AWS SCT untuk terhubung ke sumber dan menargetkan instance DB.
2. Pengguna mulai AWS SCT dengan skrip Python, mengonversi kode Oracle ke kode PostgreSQL, dan menjalankannya pada instance DB target.
3. Skrip Python membuat tugas AWS DMS replikasi untuk instance DB sumber dan target.
4. Pengguna menyebarkan skrip Python untuk memulai AWS DMS tugas dan kemudian menghentikan tugas setelah migrasi data selesai.

Otomatisasi dan skala

Anda dapat mengotomatiskan migrasi ini dengan menambahkan parameter dan perubahan terkait keamanan ke skrip Python Anda, untuk menyediakan fungsionalitas tambahan.

Alat

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- CloudFormationmembantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah. Pola ini mengkonversi file `.csv` input ke file `.json` input dengan menggunakan script Python.

`.json`File ini digunakan dalam AWS CLI perintah untuk membuat CloudFormation tumpukan yang membuat beberapa tugas AWS DMS replikasi dengan Amazon Resource Names (ARNs), tipe migrasi, pengaturan tugas, dan pemetaan tabel.

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau di antara kombinasi pengaturan cloud dan lokal. Pola ini digunakan AWS DMS untuk membuat, memulai, dan menghentikan tugas dengan skrip Python yang berjalan pada baris perintah, dan untuk membuat template. CloudFormation
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target. Pola ini membutuhkan `AWSSchemaConversionToolBatch.jar` file dari AWS SCT direktori yang diinstal.

Kode

`cli-sct-dms-cft.zip`File (terlampir) berisi kode sumber lengkap untuk pola ini.

Epik

Konfigurasikan AWS SCT dan buat objek database di AWS CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
AWS SCT Konfigurasikan untuk menjalankan dari file AWS CLI.	1. Konfigurasikan detail konfigurasi lingkungan sumber dan target dalam `database_migration.txt` file dengan menggunakan format berikut: `#source_vendor,source_hostname,source_dbname,source_user,source_pwd,source_schema,source_port,source_sid,target_vendor,target_hostname,target_us`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`er,target_pwd,targ et_dbname,target_p ort ORACLE,myoracl edb.cokmvis0v46q.u s-east-1.rds.amazo naws.com,ORCL,orcl ,orcl1234,orcl,152 1,ORCL,POSTGRESQL, mypgdbinstance.cok mvis0v46q.us-east- 1.rds.amazonaws.co m,pguser,pgpasswor d,pgdb,5432` 2. Ubah parameter AWS SCT konfigurasi sesuai dengan kebutuhan Anda dalam file berikut:`project_s ettings.x ml` ,`Oracle_PG _Test_Batch.xml` , dan`ORACLE-orcl-to- POSTGRESQL.xml` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan `run_aws_sct.py` skrip Python.	Jalankan skrip `run_aws_sct.py` Python dengan menggunakan perintah berikut: `$ python run_aws_sct.py database_migration.txt` Script Python mengkonversi objek database dari Oracle ke PostgreSQL dan membuat file SQL dalam format PostgreSQL. Skrip ini juga membuat file PDF`Database migration assessment report` , yang memberi Anda rekomendasi terperinci dan statistik konversi untuk objek database.	DBA
Buat objek di Amazon RDS untuk PostgreSQL.	1. Secara manual memodifikasi file SQL yang dihasilkan oleh AWS SCT, jika diperlukan. 2. Jalankan file SQL dan buat objek di Amazon RDS for PostgreSQL DB instance.	DBA

Konfigurasikan dan buat AWS DMS tugas dengan menggunakan AWS CLI dan CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh AWS DMS replikasi.	Masuk ke Konsol Manajemen AWS, buka AWS DMS konsol, dan buat instance replikasi	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang dikonfigurasi sesuai dengan kebutuhan Anda. Untuk informasi selengkap nya, lihat Membuat instance replikasi dalam AWS DMS dokumentasi dan Bagaimana cara membuat instance AWS DMS replikasi dalam dokumentasi. AWS Dukungan	
Buat titik akhir sumber.	Di AWS DMS konsol, pilih Endpoint dan kemudian buat titik akhir sumber untuk database Oracle sesuai dengan kebutuhan Anda. **Note** Atribut koneksi tambahan harus `numberDataTypeScale` dengan `-2` nilai. Untuk informasi selengkap nya, lihat Membuat titik akhir sumber dan target dalam AWS DMS dokumentasi.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir target.	Di AWS DMS konsol, pilih Endpoint dan kemudian buat titik akhir target untuk database PostgreSQL sesuai dengan kebutuhan Anda. Untuk informasi selengkap nya, lihat Membuat titik akhir sumber dan target dalam AWS DMS dokumentasi.	DevOps insinyur

Konfigurasikan detail AWS DMS replikasi untuk dijalankan dari file. AWS CLI	Konfigurasikan titik akhir AWS DMS sumber dan target serta detail replikasi dalam `dms-arn-list.txt` file dengan ARN titik akhir sumber, ARN titik akhir target, dan ARN instance replikasi dengan menggunakan format berikut:
	
	`#sourceARN,targetARN,repARN`
	`arn:aws:dms:us-east-1:123456789012:endpoint:EH7AINRUDZ5GOYIY6HVMXECMCQ`
	`arn:aws:dms:us-east-1:123456789012:endpoint:HHJVUV57N7O3CQF4PJZKGIOYY5`
	`arn:aws:dms:us-east-1:123456789012:rep:LL57N77AQQAHHJF4PJFHNEDZ5G`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip `dms-create-task.py` Python untuk membuat tugas. AWS DMS	1. Jalankan skrip `dms-create-task.py` Python dengan menggunakan perintah berikut: `$ python dms-create-task.py database_migration.txt dms-arn-list.txt <cft-stack-name> <migration-type>` di mana: • `database_migration.txt` adalah file teks migrasi database. • `dms-arn-list.txt` adalah daftar ARN untuk. AWS DMS • `<cft-stack-name>`adalah nama CloudFormation tumpukan yang ditentukan pengguna. • `<migration-type>`adalah`full-load`,`cdc`, atau`full-load-and-cdc` . 2. Bergantung pada jenis migrasi, Anda dapat menggunakan perintah berikut untuk membuat tiga jenis AWS DMS tugas:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `$ python dms-create-task.py database_migration.txt dms-arn-list.txt dms-cli-cft-stack full-load` • `$ python dms-create-task.py database_migration.txt dms-arn-list.txt dms-cli-cft-stack cdc` • `$ python dms-create-task.py database_migration.txt dms-arn-list.txt dms-cli-cft-stack full-load-and-cdc`	
Verifikasi bahwa AWS DMS tugas sudah siap.	Di AWS DMS konsol, periksa apakah AWS DMS tugas Anda dalam `Ready` status di bagian Status.	DBA

Memulai dan menghentikan AWS DMS tugas dengan menggunakan AWS CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai AWS DMS tugas.	Jalankan skrip `dms-start-task.py` Python dengan menggunakan perintah berikut:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`$ python dms-start-task.py start '<cdc-start-datetime>'` **Note** Tanggal dan waktu mulai harus dalam `'YYYY-MM-DDTHH:MI:SS'` format `'DD-MON-YYYY'` atau (misalnya, `'01-Dec-2019'` atau `'2018-03-08T12:12:12'`). Anda dapat meninjau status AWS DMS tugas di tab Statistik tabel di halaman Tugas AWS DMS konsol.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi data.	1. Setelah migrasi beban penuh selesai, tugas terus berjalan untuk CDC. 2. Ketika CDC selesai atau tidak ada perubahan lagi yang perlu dimigrasikan, tinjau dan validasi hasil tugas migrasi dan data di database Oracle dan PostgreSQL Anda. Anda dapat memvalidasi data dengan memeriksa kolom status dan hitungan (**`Validation state Validation pending Validation failed`** ,**`Validation suspended`** ,,, dan**`Validation details`**) di tab Statistik tabel tugas migrasi database Anda di halaman Tugas AWS DMS konsol. Untuk informasi selengkapnya, lihat validasi AWS DMS data dalam AWS DMS dokumentasi.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan AWS DMS tugas.	Jalankan skrip Python dengan menggunakan perintah berikut: `$ python dms-start-task.py stop` **Note** AWS DMS tugas mungkin berhenti dengan `failed` status, tergantung pada status validasi. Untuk informasi lebih lanjut, lihat bagian selanjutnya.	DBA

Pemecahan masalah

Isu	Solusi
AWS SCT koneksi uji sumber dan target gagal.	Konfigurasikan versi driver JDBC dan aturan masuk grup keamanan VPC untuk menerima lalu lintas masuk.
Uji coba sumber atau target titik akhir gagal.	Periksa apakah pengaturan titik akhir dan contoh replikasi berada dalam `Available` status. Periksa apakah status koneksi titik akhir adalah`Successful` . Untuk informasi selengkapnya, lihat Bagaimana cara memecahkan masalah kegagalan

Isu	Solusi
	konektivitas titik akhir AWS DMS dalam dokumentasi. AWS Dukungan
Proses beban penuh gagal.	Periksa apakah basis data sumber dan target memiliki tipe dan ukuran data yang cocok. Untuk informasi selengkapnya, lihat Memecahkan masalah tugas migrasi AWS DMS dalam dokumentasi. AWS DMS
Anda menemukan kesalahan menjalankan validasi.	Periksa apakah tabel memiliki kunci primer karena tabel kunci non-primer tidak divalidasi. Jika tabel memiliki kunci utama dan kesalahan, periksa apakah atribut koneksi tambahan di titik akhir sumber memilikinumberDataTypeScale=-2 . Untuk informasi selengkapnya, lihat Pengaturan titik akhir saat menggunakan Oracle sebagai sumber untuk AWS DMSOracleSettings, dan Pemecahan Masalah dalam dokumentasi. AWS DMS

Sumber daya terkait

- Menginstal dan mengkonfigurasi AWS SCT
- Pengantar AWS DMS (video)
- Contoh perintah operasi CloudFormation tumpukan untuk AWS CLI dan PowerShell
- Menavigasi antarmuka pengguna AWS SCT
- Menggunakan database Oracle sebagai sumber AWS DMS
- Menghubungkan ke database Oracle dengan AWS SCT
- Menggunakan database PostgreSQL sebagai target AWS DMS
- Sumber untuk migrasi data

- Target untuk migrasi data
- cloudformation (dokumentasi)AWS CLI
- buat-tumpukan (dokumentasi)AWS CLI
- dms (AWS CLI dokumentasi)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dalam mode SSL dengan menggunakan AWS DMS

Pinesh Singal, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan Amazon Relational Database Service (Amazon RDS) untuk instans database Oracle ke database Amazon RDS for PostgreSQL di Amazon Web Services (AWS) Cloud. Untuk mengenkripsi koneksi antar database, pola menggunakan otoritas sertifikat (CA) dan mode SSL di Amazon RDS dan AWS Database Migration Service (AWS DMS).

Pola ini menggambarkan strategi migrasi online dengan sedikit atau tanpa downtime untuk database sumber Oracle multi-terabyte dengan jumlah transaksi yang tinggi. Untuk keamanan data, pola menggunakan SSL saat mentransfer data.

Pola ini menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema database Amazon RDS for Oracle ke skema Amazon RDS for PostgreSQL. Kemudian pola menggunakan AWS DMS untuk memigrasikan data dari database Amazon RDS for Oracle ke database Amazon RDS for PostgreSQL.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Otoritas sertifikat database Amazon RDS (CA) dikonfigurasi hanya dengan rds-ca-rsa2048-g1
  - Sertifikat rds-ca-2019 berakhir pada Agustus 2024.
  - Sertifikat rds-ca-2015 berakhir pada 5 Maret 2020.
- AWS SCT
- AWS DMS
- pgAdmin
- Alat SQL (misalnya, Pengembang SQL atau SQL* Plus)

Batasan

- Amazon RDS for Oracle database — Persyaratan minimum adalah untuk Oracle versi 19c untuk edisi Enterprise dan Standard Two.

- Amazon RDS for PostgreSQL database - Persyaratan minimum adalah untuk PostgreSQL versi 12 dan yang lebih baru (untuk versi 9.x dan yang lebih baru).

Versi produk

- Amazon RDS for Oracle database versi 12.1.0.2 misalnya
- Amazon RDS untuk database PostgreSQL versi 11.5 contoh

Arsitektur

Tumpukan teknologi sumber

- Instans database Amazon RDS for Oracle dengan versi 12.1.0.2.v18.

Tumpukan teknologi target

- AWS DMS
- Instans database Amazon RDS for PostgreSQL dengan versi 11.5.

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk arsitektur migrasi data antara Oracle (sumber) dan PostgreSQL (target) database. Arsitekturnya meliputi:

- Awan pribadi virtual (VPC)
- Zona Ketersediaan
- Subnet pribadi
- Database Amazon RDS for Oracle
- Instans replikasi AWS DMS
- Sebuah RDS untuk database PostgreSQL

Untuk mengenkripsi koneksi untuk basis data sumber dan target, mode CA dan SSL harus diaktifkan di Amazon RDS dan AWS DMS.

Alat

Layanan AWS

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di AWS Cloud.
- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di AWS Cloud.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Layanan lainnya

- pgAdmin adalah alat manajemen open source untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.

Praktik terbaik

Amazon RDS menyediakan sertifikat CA baru sebagai praktik terbaik keamanan AWS. Untuk informasi tentang sertifikat baru dan Wilayah AWS yang didukung, lihat Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans atau cluster DB.

Jika instans RDS Anda saat ini menggunakan sertifikat CA`rds-ca-2019`, dan Anda ingin meningkatkan k`erds-ca-rsa2048-g1`, ikuti petunjuk dalam Memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB Anda atau Memperbarui sertifikat CA Anda dengan menerapkan pemeliharaan.

Epik

Konfigurasikan Amazon RDS for Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance database Oracle.	Masuk ke akun AWS Anda, buka AWS Management	AWS Umum, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Console, dan navigasikan ke konsol Amazon RDS. Di konsol, pilih Buat database, lalu pilih Oracle.	
Konfigurasikan grup keamanan.	Konfigurasikan grup keamanan masuk dan keluar.	AWS Umum
Buat grup opsi.	Buat grup opsi di VPC dan grup keamanan yang sama dengan database Amazon RDS for Oracle. Untuk Opsi, pilih SSL. Untuk Port, pilih 2484 (untuk koneksi SSL).	AWS Umum
Konfigurasikan pengaturan opsi.	Gunakan pengaturan berikut: • `SQLNET.CIPHER_SUITE : SSL_RSA_WITH_AES_256_CBC_SHA` • `SQLNET.SSL_VERSION : 1.2 or 1.0`	AWS Umum
Ubah RDS untuk instans Oracle DB.	Tetapkan sertifikat CA sebagai rds-ca-rsa2048-g1. Di bawah grup Opsi, lampirkan grup opsi yang dibuat sebelumnya.	DBA, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan bahwa instans RDS untuk Oracle DB tersedia.	Pastikan bahwa instans database Amazon RDS for Oracle aktif dan berjalan dan skema database dapat diakses.
	
	Untuk terhubung ke RDS untuk Oracle DB, gunakan `sqlplus` perintah dari baris perintah.
	
	`$ sqlplus orcl/**** @myoracledb.cokmvis0v46q.us-east-1.rds.amazonaws.com:1521/ORCL`
	`SQL*Plus: Release 12.1.0.2.0 Production on Tue Oct 15 18:11:07 2019`
	`Copyright (c) 1982, 2016, Oracle. All rights reserved.`
	`Last Successful login time: Mon Dec 16 2019 23:17:31 +05:30`
	`Connected to:`
	`Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production`
	`With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options`
	`SQL>`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat objek dan data dalam database RDS untuk Oracle.	Buat objek dan masukkan data dalam skema.	DBA

Konfigurasikan Amazon RDS untuk instance PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat RDS untuk database PostgreSQL.	Pada halaman database Buat konsol Amazon RDS, pilih PostgreSQL untuk membuat instance database Amazon RDS for PostgreSQL.	DBA, AWS Umum
Konfigurasikan grup keamanan.	Konfigurasikan grup keamanan masuk dan keluar.	AWS Umum
Buat grup parameter.	Jika Anda menggunakan PostgreSQL versi 11.x, buat grup parameter untuk mengatur parameter SSL. Di PostgreSQL versi 12, grup parameter SSL diaktifkan secara default.	AWS Umum
Edit parameter.	Ubah `rds.force_ssl` parameter menjadi 1 (on). Secara default, `ssl` parameter nya adalah 1 (on). Dengan mengatur `rds.force_ssl` parameter ke1, Anda memaksa semua koneksi untuk terhubung melalui mode SSL saja.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memodifikasi RDS untuk PostgreSQL DB instance.	Tetapkan sertifikat CA sebagai rds-ca-rsa2048-g1. Lampirkan grup parameter default atau grup parameter yang dibuat sebelumnya, tergantung pada versi PostgreSQL Anda.	DBA, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan bahwa RDS untuk instance PostgreSQL DB tersedia.	Pastikan database Amazon RDS for PostgreSQL aktif dan berjalan.
	
	`psql`Perintah membuat koneksi SSL dengan `sslmode` set dari baris perintah.
	
	Salah satu opsi adalah mengatur `sslmode=1` dalam grup parameter dan menggunakan `psql` koneksi tanpa menyertakan `sslmode` parameter dalam perintah.
	
	Output berikut menunjukkan bahwa koneksi SSL dibuat.
	
	`$ psql -h mypgdbinstance.cokmvis0v46q.us-east-1.rds.amazonaws.com -p 5432 "dbname=pgdb user=pguser"`
	`Password for user pguser:`
	`psql (11.3, server 11.5)`
	`SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)`
	`Type "help" for help.`
	`pgdb=>`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Opsi kedua adalah mengatur `sslmode=1` dalam grup parameter dan memasukkan `sslmode` parameter dalam `psql` perintah.
	
	Output berikut menunjukkan bahwa koneksi SSL dibuat.
	
	`$ psql -h mypgdbinstance.cokmvis0v46q.us-east-1.rds.amazonaws.com -p 5432 "dbname=pgdb user=pguser sslmode=require"`
	`Password for user pguser:`
	`psql (11.3, server 11.5)`
	`SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)`
	`Type "help" for help.`
	`pgdb=>`	

Konfigurasikan dan jalankan AWS SCT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS SCT.	Instal versi terbaru aplikasi AWS SCT.	AWS Umum
Konfigurasikan AWS SCT dengan driver JDBC.	Unduh driver Java Database Connectivity (JDBC) untuk Oracle (ojdbc8.jar) dan	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	PostgreSQL (postgresql-42.2.5.jar). Untuk mengonfigurasi driver di AWS SCT, pilih Pengaturan, Pengaturan global, Driver.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat proyek AWS SCT.	Buat proyek AWS SCT dan laporkan, menggunakan Oracle sebagai mesin DB sumber dan Amazon RDS for PostgreSQL sebagai mesin DB target: 1. Uji koneksi ke database Oracle sumber dan targetkan Amazon RDS for PostgreSQL database dengan memberikan detail koneksi. Untuk database Oracle sumber, izin atau hak istimewa berikut diperlukan: • `CONNECT` • `SELECT_CATALOG_ROLE` • `SELECT ANY DICTIONARY` • `SELECT on SYS.USER$ TO <sct_user>` Untuk informasi selengkapnya, lihat Menggunakan Oracle Database sebagai sumber AWS SCT. Koneksi sumber dan target harus berhasil sebelum AWS SCT dapat memulai laporan migrasi.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Setelah laporan, masukkan skema yang akan dikonversi, dan pilih Selesai.	
Validasi objek database.	1. Pilih skema Muat. AWS SCT menampilkan sumber dan objek target yang dikonversi, termasuk objek yang memiliki kesalahan. Perbarui objek yang salah pada database target. 2. Tinjau kesalahan, dan bersihkan dengan menggunakan intervensi manual. 3. Setelah semua kesalahan dihapus, pilih Muat skema lagi. 4. Pilih Terapkan ke database. 5. Connect ke pgAdmin, atau alat apa pun yang mendukung koneksi PostgreSQL DB, dan periksa skema dan objek.	DBA, AWS Umum

Konfigurasikan dan jalankan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh replikasi.	1. Masuk ke akun Anda, buka AWS Management	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Console, dan navigasikan ke konsol AWS DMS. 2. Buat instance replikasi dengan pengaturan yang valid untuk VPC, grup keamanan, Availability Zone, dan atribut koneksi tambahan.	
Impor sertifikat.	Unduh paket sertifikat (PEM) untuk Wilayah AWS Anda. Bundel berisi sertifikat `rds-ca-2019` perantara dan root. Bundel ini juga berisi sertifikat `rds-ca-rsa2048-g1` `rds-ca-rsa4096-g1` ,, dan `rds-ca-ecc384-g1` root CA. Toko kepercayaan aplikasi Anda hanya perlu mendaftar kan sertifikat root CA.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir sumber.	1. Buat titik akhir sumber untuk Amazon RDS for Oracle dengan memilih instans Select RDS DB dan kemudian memilih instans RDS untuk Oracle DB yang Anda buat. Detail konfigurasi titik akhir akan diisi secara otomatis. 2. Pilih Berikan informasi akses secara manual. Untuk Port, pastikan Anda memasukkan 2484. 3. Di bawah mode Secure Socket Layer (SSL)`verify-ca` , pilih, lalu pilih sertifikat CA yang Anda buat sebelumnya. 4. Di bawah Pengaturan titik akhir, tambahkan atribut koneksi tambahan `NumberDataTypeScale=-2` untuk mendukung tipe `NUMBER` data tanpa ukuran. Untuk informasi selengkapnya, lihat Menggunakan database Oracle sebagai sumber AWS Database Migration Service.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir target.	1. Buat titik akhir target untuk Amazon RDS for PostgreSQL dengan memilih instans Select RDS DB dan kemudian memilih RDS Anda untuk instans PostgreSQL DB. Detail konfigurasi titik akhir akan diisi secara otomatis. 2. Pilih Berikan informasi akses secara manual. Untuk Port, pastikan Anda memasukkan 2484. Untuk informasi selengkapnya, lihat Menggunakan database PostgreSQL sebagai target AWS Database Migration Service.	AWS Umum
Uji titik akhir.	1. Uji sumber dan titik akhir target untuk mengonfirmasi bahwa keduanya berhasil dan tersedia. 2. Jika pengujian gagal, pastikan aturan masuk grup keamanan valid.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas migrasi.	Untuk membuat tugas migrasi untuk memuat penuh dan mengubah pengambilan data (CDC) atau untuk validasi data, lakukan hal berikut: 1. Untuk membuat tugas migrasi database, pilih contoh replikasi, titik akhir basis data sumber, titik akhir basis data target. Tentukan jenis migrasi sebagai salah satu dari berikut ini: • Migrasi data yang ada (beban penuh) • Replikasi perubahan data saja (CDC) • Migrasi data yang ada dan replikasi perubahan yang sedang berlangsung (full load dan CDC) 2. Di bawah Pemetaan tabel, Anda dapat mengonfigurasi aturan pemilihan dan aturan transformasi dalam format GUI atau JSON: • Di bawah Aturan pemilihan, pilih skema, masukkan nama tabel, dan pilih tindakan (Sertakan atau Kecualikan) yang akan dikonfigurasi; misalnya, Skema	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	ORCL, Nama tabel%, Tindakan Termasuk.
	• Di bawah aturan Transformasi, lakukan salah satu hal berikut:
	  • Pilih skema dan pilih tindakan (kasus, awalan, akhiran); misalnya, Target Schema ORCL, Action Make huruf kecil.
	  • Pilih skema, masukkan nama tabel, dan pilih tindakan (kasus, awalan, akhiran); misalnya, Target Skema ORCL, Tabel %, Tindakan Buat huruf kecil.
	3. Aktifkan pemantauan Amazon CloudWatch Logs.
	4. Untuk aturan pemetaan, tambahkan kode JSON berikut.
	
	`{ "rules": [ { "rule-type": "transformation", "rule-id": "1", "rule-name": "1",`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"rule-target": "table", "object-locator": { "schema-name": "%", "table-name": "%" }, "rule-action": "convert-lowercase", "value": null, "old-value": null }, { "rule-type": "transformation", "rule-id": "2", "rule-name": "2", "rule-target": "schema", "object-locator": { "schema-name": "ORCL", "table-name": "%" }, "rule-action": "convert-lowercase", "value": null, "old-value": null	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`}, { "rule-type": "selection", "rule-id": "3", "rule-name": "3", "object-locator": { "schema-name": "ORCL", "table-name": "DEPT" }, "rule-action": "include", "filters": [] } ] }`	
Rencanakan proses produksi.	Konfirmasikan waktu henti dengan pemangku kepentingan seperti pemilik aplikasi untuk menjalankan AWS DMS dalam sistem produksi.	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan tugas migrasi .	1. Mulai tugas AWS DMS yang memiliki status Siap, dan pantau log tugas migrasi di Amazon CloudWatch untuk mengetahui kesalahan apa pun. Jika Anda memilih Migrasi data yang ada dan mereplikasi perubahan yang sedang berlangsung sebagai jenis migrasi, dan statusnya adalah Muat replikasi berkelanjutan lengkap, beban penuh dengan migrasi data CDC selesai dan validasi sedang berlangsung. 2. Setelah memulai migrasi, Anda dapat memperoleh informasi koneksi SSL tambahan di. CloudWatch Untuk Oracle, CloudWatch menunjukkan string koneksi berikut. `2019-12-17T09:15:11 [SOURCE_UNLOAD ]I: Connecting to Oracle: Beginning session (oracle_endpoint_conn.c:834)`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	String koneksi PostgreSQL akan mirip dengan contoh berikut. `2019-12-17T09:15:11 [TARGET_LOAD ]I: Going to connect to ODBC connection string: PROTOCOL=7.4-0;DRIVER={PostgreSQL};SERVER=mypgdbinstance.cokmvis0v46q.us-east-1.rds.amazonaws.com;DATABASE=pgdb;PORT=5432;sslmode=require;UID=pguser; (odbc_endpoint_imp.c:2218)`	
Validasi data.	Tinjau hasil tugas migrasi dan data di sumber Oracle dan target database PostgreSQL: 1. Connect ke pgAdmin dan periksa data dalam database PostgreSQL Anda dengan skema. `ORCL` 2. Untuk CDC, periksa perubahan yang sedang berlangsung dengan memasukkan atau memperbarui data dalam database Oracle sumber.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan tugas migrasi.	Setelah Anda berhasil menyelesaikan validasi data, hentikan tugas migrasi.	AWS Umum

Bersihkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus tugas AWS DMS.	1. Di konsol AWS DMS, navigasikan ke tugas migrasi Database, dan hentikan tugas AWS DMS yang sedang berlangsung atau berjalan. 2. Pilih tugas atau tugas, pilih Tindakan, dan pilih Hapus.	AWS Umum
Hapus titik akhir AWS DMS.	Pilih titik akhir sumber dan target yang Anda buat, pilih Tindakan, dan pilih Hapus.	AWS Umum
Hapus instans replikasi AWS DMS.	Pilih contoh replikasi, pilih Tindakan, lalu pilih Hapus.	AWS Umum
Hapus database PostgreSQL.	1. Di konsol Amazon RDS, pilih Database. 2. Pilih instance database PostgreSQL yang Anda buat, pilih Tindakan, lalu pilih Hapus.	AWS Umum
Hapus database Oracle.	Di konsol Amazon RDS, pilih instance database Oracle, pilih Actions, lalu pilih Delete.	AWS Umum

Pemecahan masalah

Isu	Solusi
Sumber AWS SCT dan koneksi pengujian target gagal.	Konfigurasikan versi driver JDBC dan aturan masuk grup keamanan VPC untuk menerima lalu lintas masuk.
Uji titik akhir sumber Oracle gagal.	Periksa pengaturan titik akhir dan apakah instance replikasi tersedia.
Proses beban penuh tugas AWS DMS gagal.	Periksa apakah basis data sumber dan target memiliki tipe dan ukuran data yang cocok.
Tugas migrasi validasi AWS DMS mengembal ikan kesalahan.	1. Periksa apakah tabel memiliki kunci utama. Tabel tanpa kunci primer tidak divalidasi. 2. Jika tabel memiliki kunci primer tetapi mengembalikan kesalahan, periksa atribut koneksi tambahan di titik akhir sumber. Atribut koneksi tambahan `numberDataTypeScale=-2` harus mendukung tipe `NUMBER` data tanpa ukuran dinamis berdasarkan data yang tersedia dalam tabel.

Sumber daya terkait

Basis Data

- Amazon RDS for Oracle
- Amazon RDS untuk PostgreSQL

Koneksi SSL DB

- Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans DB
  - Menggunakan SSL dengan RDS untuk instans Oracle DB
  - Mengamankan koneksi ke RDS untuk PostgreSQL dengan SSL/TLS
  - Unduh bundel sertifikat untuk Wilayah AWS tertentu

    - Unduh sertifikat root CA-2019 (kedaluwarsa pada Agustus 2024)
- Bekerja dengan grup opsi
    - Menambahkan opsi ke instans Oracle DB
    - Lapisan Soket Aman Oracle
- Bekerja dengan kelompok parameter
- Parameter koneksi sslmode PostgreSQL
- Menggunakan SSL dari JDBC
- Memutar sertifikat Anda SSL/TLS
    - Memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB
    - Memperbarui sertifikat CA Anda dengan menerapkan pemeliharaan

AWS SCT

- Alat Konversi Skema AWS
- Panduan Pengguna Alat Konversi Skema AWS
- Menggunakan antarmuka pengguna AWS SCT
- Menggunakan Oracle Database sebagai sumber AWS SCT

AWS DMS

- AWS Database Migration Service
- Panduan Pengguna AWS Database Migration Service
    - Menggunakan database Oracle sebagai sumber AWS DMS
    - Menggunakan database PostgreSQL sebagai target AWS DMS
- Menggunakan SSL dengan AWS Database Migration Service
- Migrasi Aplikasi Menjalankan Database Relasional ke AWS

Informasi tambahan

Sertifikat Otoritas Sertifikat Amazon RDS `rds-ca-2019` kedaluwarsa pada Agustus 2024. Jika Anda menggunakan atau berencana menggunakan SSL atau TLS dengan verifikasi sertifikat untuk terhubung ke instans RDS DB atau klaster DB multi-AZ, pertimbangkan untuk menggunakan salah satu sertifikat CA baru:,, atau. `rds-ca-rsa2048-g1` `rds-ca-rsa4096-g1` `rds-ca-ecc384-g1`

Migrasikan paket pragma Oracle SERIALLY_REUSABLE ke PostgreSQL

Vinay Paladi, Amazon Web Services

Ringkasan

Pola ini menyediakan step-by-step pendekatan untuk memigrasikan paket Oracle yang didefinisikan sebagai SERIALLY_REUSABLE pragma ke PostgreSQL di Amazon Web Services (AWS). Pendekatan ini mempertahankan fungsionalitas pragma SERIALLY_REUSABLE.

PostgreSQL tidak mendukung konsep paket dan pragma SERIALLY_REUSABLE. Untuk mendapatkan fungsionalitas serupa di PostgreSQL, Anda dapat membuat skema untuk paket dan menyebarkan semua objek terkait (seperti fungsi, prosedur, dan tipe) di dalam skema. Untuk mencapai fungsionalitas pragma SERIALLY_REUSABLE, contoh skrip fungsi pembungkus yang disediakan dalam pola ini menggunakan paket ekstensi AWS Schema Conversion Tool (AWS SCT).

Untuk informasi selengkapnya, lihat SERIALLY_REUSABLE Pragma dalam dokumentasi Oracle.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Versi terbaru AWS SCT dan driver yang diperlukan
- Database Edisi yang kompatibel dengan Amazon Aurora PostgreSQL atau Amazon Relational Database Service (Amazon RDS) untuk database PostgreSQL

Versi produk

- Oracle Database versi 10g dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database Oracle di tempat

Tumpukan teknologi target

- Aurora PostgreSQL kompatibel atau Amazon RDS untuk PostgreSQL
- AWS SCT

Arsitektur migrasi

Alat

Layanan AWS

- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.
- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di AWS Cloud.

Alat-alat lainnya

- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.

Epik

Migrasikan paket Oracle dengan menggunakan AWS SCT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS SCT.	Konfigurasikan konektivitas AWS SCT ke database sumber. Untuk informasi selengkapnya, lihat Menggunakan Oracle Database sebagai sumber AWS SCT.	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konversi skrip.	Gunakan AWS SCT untuk mengonversi paket Oracle dengan memilih database target sebagai Aurora PostgreSQL kompatibel.	DBA, Pengembang
Simpan file.sql.	Sebelum Anda menyimpan file.sql, ubah opsi Pengatura n Proyek di AWS SCT ke File tunggal per tahap. AWS SCT akan memisahkan file.sql menjadi beberapa file.sql berdasarkan jenis objek.	DBA, Pengembang
Ubah kodenya.	Buka `init` fungsi yang dihasilkan oleh AWS SCT, dan ubah seperti yang ditunjukk an pada contoh di bagian Informasi tambahan. Ini akan menambahkan variabel untuk mencapai fungsiona litas`pg_serialize = 0` .	DBA, Pengembang
Uji konversi.	Menyebarkan `init` fungsi ke database yang kompatibe l dengan Aurora PostgreSQL, dan uji hasilnya.	DBA, Pengembang

Sumber daya terkait

- Alat Konversi Skema AWS
- Amazon RDS
- Fitur Amazon Aurora
- SERIALLY_REUSABLE Pragma

Informasi tambahan

```
Source Oracle Code:

CREATE OR REPLACE PACKAGE test_pkg_var
IS
PRAGMA SERIALLY_REUSABLE;
PROCEDURE function_1
 (test_id number);
PROCEDURE function_2
 (test_id number
 );
END;

CREATE OR REPLACE PACKAGE BODY test_pkg_var
IS
PRAGMA SERIALLY_REUSABLE;
v_char VARCHAR2(20) := 'shared.airline';
v_num number := 123;

PROCEDURE function_1(test_id number)
IS
begin
dbms_output.put_line( 'v_char-'|| v_char);
dbms_output.put_line( 'v_num-'||v_num);
v_char:='test1';
function_2(0);
END;

PROCEDURE function_2(test_id number)
is
begin
dbms_output.put_line( 'v_char-'|| v_char);
dbms_output.put_line( 'v_num-'||v_num);
END;
END test_pkg_var;

Calling the above functions

set serveroutput on


EXEC test_pkg_var.function_1(1);
```

```
EXEC test_pkg_var.function_2(1);


Target Postgresql Code:


CREATE SCHEMA test_pkg_var;

CREATE OR REPLACE FUNCTION test_pkg_var.init(pg_serialize IN INTEGER DEFAULT 0)

RETURNS void
AS
$BODY$

DECLARE

BEGIN

if aws_oracle_ext.is_package_initialized( 'test_pkg_var' ) AND pg_serialize = 0

then

return;

end if;

PERFORM aws_oracle_ext.set_package_initialized( 'test_pkg_var' );

PERFORM aws_oracle_ext.set_package_variable( 'test_pkg_var', 'v_char',
 'shared.airline.basecurrency'::CHARACTER

VARYING(100));

PERFORM aws_oracle_ext.set_package_variable('test_pkg_var', 'v_num', 123::integer);

END;

$BODY$

LANGUAGE plpgsql;


CREATE OR REPLACE FUNCTION test_pkg_var.function_1(pg_serialize int default 1)
```

```
RETURNS void
AS

$BODY$
DECLARE

BEGIN

PERFORM test_pkg_var.init(pg_serialize);

raise notice 'v_char%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_char');

raise notice 'v_num%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_num');

PERFORM aws_oracle_ext.set_package_variable( 'test_pkg_var', 'v_char',
 'test1'::varchar);

PERFORM test_pkg_var.function_2(0);
END;

$BODY$
LANGUAGE plpgsql;


CREATE OR REPLACE FUNCTION test_pkg_var.function_2(IN pg_serialize integer default 1)

RETURNS void

AS

$BODY$

DECLARE

BEGIN

PERFORM test_pkg_var.init(pg_serialize);

raise notice 'v_char%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_char');

raise notice 'v_num%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_num');

END;
```

```
$BODY$
LANGUAGE plpgsql;


Calling the above functions

select test_pkg_var.function_1()

 select test_pkg_var.function_2()
```

Migrasikan tabel eksternal Oracle ke Amazon Aurora PostgreSQL yang kompatibel

anuradha chintha dan Rakesh Raghav, Amazon Web Services

Ringkasan

Tabel eksternal memberikan Oracle kemampuan untuk query data yang disimpan di luar database dalam file datar. Anda dapat menggunakan driver ORACLE_LOADER untuk mengakses data apa pun yang disimpan dalam format apa pun yang dapat dimuat oleh utilitas SQL* Loader. Anda tidak dapat menggunakan Data Manipulation Language (DHTML) pada tabel eksternal, tetapi Anda dapat menggunakan tabel eksternal untuk operasi kueri, bergabung, dan mengurutkan.

Amazon Aurora PostgreSQL Compatible Edition tidak menyediakan fungsionalitas yang mirip dengan tabel eksternal di Oracle. Sebaliknya, Anda harus menggunakan modernisasi untuk mengembangkan solusi terukur yang memenuhi persyaratan fungsional dan hemat.

Pola ini menyediakan langkah-langkah untuk memigrasikan berbagai jenis tabel eksternal Oracle ke Aurora PostgreSQL Compatible Edition di Amazon Web Services (AWS) Cloud dengan menggunakan ekstensi. `aws_s3`

Kami merekomendasikan pengujian solusi ini secara menyeluruh sebelum menerapkannya di lingkungan produksi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Antarmuka Baris Perintah AWS (AWS CLI)
- Sebuah instance database Aurora PostgreSQL yang kompatibel dengan Aurora.
- Database Oracle lokal dengan tabel eksternal
- PG.Client API
- File data

Batasan

- Pola ini tidak menyediakan fungsionalitas untuk bertindak sebagai pengganti tabel eksternal Oracle. Namun, langkah-langkah dan kode sampel dapat ditingkatkan lebih lanjut untuk mencapai tujuan modernisasi database Anda.

- File tidak boleh berisi karakter yang diteruskan sebagai pembatas dalam fungsi `aws_s3` ekspor dan impor.

Versi produk

- Untuk mengimpor dari Amazon S3 ke RDS untuk PostgreSQL database harus menjalankan PostgreSQL versi 10.7 atau yang lebih baru.

Arsitektur

Tumpukan teknologi sumber

- Oracle

Arsitektur sumber

Tumpukan teknologi target

- Kompatibel dengan Amazon Aurora PostgreSQL
- Amazon CloudWatch
- AWS Lambda
- AWS Secrets Manager
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Storage Service (Amazon S3)

Arsitektur target

Diagram berikut menunjukkan representasi tingkat tinggi dari solusi.

1. File diunggah ke bucket S3.
2. Fungsi Lambda dimulai.
3. Fungsi Lambda memulai panggilan fungsi DB.
4. Secrets Manager menyediakan kredensyal untuk akses database.
5. Tergantung pada fungsi DB, alarm SNS dibuat.

Otomatisasi dan skala

Setiap penambahan atau perubahan pada tabel eksternal dapat ditangani dengan pemeliharaan metadata.

Alat

- Kompatibel dengan Amazon Aurora PostgreSQL — Amazon Aurora PostgreSQL Compatible Edition adalah mesin basis data relasional relasional yang dikelola sepenuhnya, kompatibel dengan PostgreSQL, dan sesuai dengan Asam yang menggabungkan kecepatan dan keandalan database komersial kelas atas dengan efektivitas biaya database sumber terbuka.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat terpadu untuk mengelola layanan AWS Anda. Dengan hanya satu alat untuk mengunduh dan mengonfigurasi, Anda dapat mengontrol beberapa layanan AWS dari baris perintah dan mengotomatiskannya melalui skrip.
- Amazon CloudWatch — Amazon CloudWatch memantau sumber daya dan pemanfaatan Amazon S3.
- AWS Lambda — AWS Lambda adalah layanan komputasi tanpa server yang mendukung menjalankan kode tanpa menyediakan atau mengelola server, membuat logika penskalaan klaster yang sadar beban kerja, mempertahankan integrasi peristiwa, atau mengelola runtime. Dalam pola ini, Lambda menjalankan fungsi database setiap kali file diunggah ke Amazon S3.
- AWS Secrets Manager — AWS Secrets Manager adalah layanan untuk penyimpanan dan pengambilan kredensyal. Dengan menggunakan Secrets Manager, Anda dapat mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service menyediakan lapisan penyimpanan untuk menerima dan menyimpan file untuk konsumsi dan transmisi ke dan dari cluster yang kompatibel dengan Aurora PostgreSQL.
- aws_s3 - Ekstensi `aws_s3` mengintegrasikan Amazon S3 dan Aurora PostgreSQL kompatibel.
- Amazon SNS - Amazon Simple Notification Service (Amazon SNS) mengoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien. Dalam pola ini, Amazon SNS digunakan untuk mengirim notifikasi.

Kode

Setiap kali file ditempatkan di bucket S3, fungsi DB harus dibuat dan dipanggil dari aplikasi pemrosesan atau fungsi Lambda. Untuk detailnya, lihat kode (terlampir).

Epik

Buat file eksternal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan file eksternal ke database sumber.	Buat file eksternal, dan pindahkan ke `oracle` direktori .	DBA

Konfigurasikan target (kompatibel dengan Aurora PostgreSQL)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database Aurora PostgreSQL.	Buat instans DB di klaster yang kompatibel dengan Amazon Aurora PostgreSQL.	DBA
Buat skema, ekstensi aws_s3, dan tabel.	Gunakan kode di bawah `ext_tbl_scripts` di bagian Informasi tambahan. Tabel termasuk tabel aktual, tabel pementasan, kesalahan dan tabel log, dan metatable.	DBA, Pengembang
Buat fungsi DB.	Untuk membuat fungsi DB, gunakan kode di bawah `load_external_tabl e_latest` fungsi di bagian Informasi tambahan.	DBA, Pengembang

Buat fungsi Lambda dengan konsol

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran.	Buat peran dengan izin untuk mengakses Amazon S3 dan Amazon Relational Database Service (Amazon RDS). Peran ini akan ditugaskan ke Lambda untuk menjalankan pola.	DBA
Buat fungsi Lambda.	Buat fungsi Lambda yang membaca nama file dari Amazon S3 (misalnya `file_key = info.get( 'object', {}).get(' key') ` ,) dan memanggil fungsi DB (misalnya `curs.callproc("loa d_external_tables" , [file_key]) ` ,) dengan nama file sebagai parameter input. Bergantung pada hasil pemanggilan fungsi, pemberita huan SNS akan dimulai (misalnya,`client.pu blish(TopicArn='ar n:',Message='filel oadsuccess',Subjec t='fileloadsuccess ') )`. Berdasarkan kebutuhan bisnis Anda, Anda dapat membuat fungsi Lambda dengan kode	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tambahan jika diperlukan. Untuk informasi selengkapnya, lihat dokumentasi Lambda.	
Konfigurasikan pemicu peristiwa bucket S3.	Konfigurasikan mekanisme untuk memanggil fungsi Lambda untuk semua peristiwa pembuatan objek di bucket S3.	DBA
Buat rahasia.	Buat nama rahasia untuk kredensyal database dengan menggunakan Secrets Manager. Lewati rahasia dalam fungsi Lambda.	DBA
Unggah file pendukung Lambda.	Unggah file.zip yang berisi paket dukungan Lambda dan skrip Python terlampir untuk menghubungkan ke Aurora PostgreSQL yang kompatibel. Kode Python memanggil fungsi yang Anda buat dalam database.	DBA
Membuat sebuah topik SNS.	Buat topik SNS untuk mengirim email untuk keberhasilan atau kegagalan pemuatan data.	DBA

Tambahkan integrasi dengan Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Di konsol Amazon S3, buat bucket S3 dengan nama unik yang tidak mengandung garis miring di depan. Nama bucket S3 unik secara global, dan namespace dibagikan oleh semua akun AWS.	DBA
Buat kebijakan IAM.	Untuk membuat kebijakan AWS Identity and Access Management (IAM), gunakan kode `s3bucketpolicy_for_import` di bagian Informasi tambahan.	DBA
Buat peran.	Buat dua peran untuk Aurora PostgreSQL kompatibel, satu peran untuk Impor dan satu peran untuk Ekspor. Tetapkan kebijakan yang sesuai untuk peran.	DBA
Lampirkan peran ke cluster yang kompatibel dengan Aurora PostgreSQL.	Di bawah Kelola peran, lampirkan peran Impor dan Ekspor ke klaster PostgreSQL Aurora.	DBA
Buat objek pendukung untuk Aurora PostgreSQL kompatibel.	Untuk skrip tabel, gunakan kode di bawah `ext_tbl_scripts` di bagian Informasi tambahan. Untuk fungsi kustom, gunakan kode di bawah `load_exte`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`rnal_Table_latest` di bagian Informasi tambahan.	

Memproses file uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file ke bucket S3.	Untuk mengunggah file pengujian ke bucket S3, gunakan konsol atau perintah berikut di AWS CLI. `aws s3 cp /Users/Desktop/ukpost/exttbl/"testing files"/aps s3://s3importtest/inputext/aps` Segera setelah file diunggah, peristiwa bucket memulai fungsi Lambda, yang menjalankan fungsi yang kompatibel dengan Aurora PostgreSQL.	DBA
Periksa data dan file log dan kesalahan.	Fungsi Aurora PostgreSQL yang kompatibel memuat file ke dalam tabel utama, dan itu membuat `.log` dan `.bad` file di bucket S3.	DBA
Pantau solusinya.	Di CloudWatch konsol Amazon, pantau fungsi Lambda.	DBA

Sumber daya terkait

- Integrasi Amazon S3
- Amazon S3
- Bekerja dengan Amazon Aurora PostgreSQL Edisi yang kompatibel
- AWS Lambda
- Amazon CloudWatch
- AWS Secrets Manager
- Menyiapkan notifikasi Amazon SNS

Informasi tambahan

ext_table_scripts

```
CREATE EXTENSION aws_s3 CASCADE;
CREATE TABLE IF NOT EXISTS meta_EXTERNAL_TABLE
(
    table_name_stg character varying(100) ,
    table_name character varying(100)  ,
    col_list character varying(1000)  ,
    data_type character varying(100)  ,
    col_order numeric,
    start_pos numeric,
    end_pos numeric,
    no_position character varying(100)  ,
    date_mask character varying(100)  ,
    delimeter character(1)  ,
    directory character varying(100)  ,
    file_name character varying(100)  ,
    header_exist character varying(5)
);
CREATE TABLE IF NOT EXISTS ext_tbl_stg
(
    col1 text
);
CREATE TABLE IF NOT EXISTS error_table
(
    error_details text,
    file_name character varying(100),
    processed_time timestamp without time zone
```

```
);
CREATE TABLE IF NOT EXISTS log_table
(
    file_name character varying(50) COLLATE pg_catalog."default",
    processed_date timestamp without time zone,
    tot_rec_count numeric,
    proc_rec_count numeric,
    error_rec_count numeric
);
sample insert scripts of meta data:
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'source_filename', 'character varying', 2, 8, 27, NULL, NULL, NULL, 'databasedev',
 'externalinterface/loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'record_type_identifier', 'character varying', 3, 28, 30, NULL, NULL, NULL,
 'databasedev', 'externalinterface/loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'fad_code', 'numeric', 4, 31, 36, NULL, NULL, NULL, 'databasedev', 'externalinterface/
loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'session_sequence_number', 'numeric', 5, 37, 42, NULL, NULL, NULL, 'databasedev',
 'externalinterface/loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'transaction_sequence_number', 'numeric', 6, 43, 48, NULL, NULL, NULL, 'databasedev',
 'externalinterface/loaddir/APS', 'NO');
```

s3bucketpolicy_untuk impor

```
---Import role policy
--Create an IAM policy to allow, Get,  and list actions on S3 bucket
 {
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "s3import",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::s3importtest",
                "arn:aws:s3:::s3importtest/*"
            ]
        }
    ]
}
--Export Role policy
--Create an IAM policy to allow, put,  and list actions on S3 bucket
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3export",
            "Action": [
                "S3:PutObject",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::s3importtest/*"
            ]
        }
    ]
}
```

Contoh fungsi DB load_external_tables_latest

```
CREATE OR REPLACE FUNCTION public.load_external_tables(pi_filename text)
 RETURNS character varying
 LANGUAGE plpgsql
AS $function$
/* Loading data from S3 bucket into a APG table */
DECLARE
 v_final_sql TEXT;
 pi_ext_table TEXT;
```

```
r refCURSOR;
v_sqlerrm text;
v_chunk numeric;
i integer;
v_col_list TEXT;
v_postion_list CHARACTER VARYING(1000);
v_len  integer;
v_delim varchar;
v_file_name CHARACTER VARYING(1000);
v_directory CHARACTER VARYING(1000);
v_table_name_stg CHARACTER VARYING(1000);
v_sql_col TEXT;
v_sql TEXT;
v_sql1 TEXT;
v_sql2 TEXT;
v_sql3 TEXT;
v_cnt integer;
v_sql_dynamic TEXT;
v_sql_ins TEXT;
proc_rec_COUNT integer;
error_rec_COUNT integer;
tot_rec_COUNT integer;
v_rec_val integer;
rec record;
v_col_cnt integer;
kv record;
v_val text;
v_header text;
j integer;
ERCODE VARCHAR(5);
v_region text;
cr CURSOR FOR
SELECT distinct DELIMETER,
  FILE_NAME,
  DIRECTORY
FROM  meta_EXTERNAL_TABLE
WHERE table_name = pi_ext_table
  AND DELIMETER IS NOT NULL;


cr1 CURSOR FOR
  SELECT   col_list,
  data_type,
  start_pos,
```

```
   END_pos,
   concat_ws('',' ',TABLE_NAME_STG) as TABLE_NAME_STG,
   no_position,date_mask
 FROM  meta_EXTERNAL_TABLE
 WHERE table_name = pi_ext_table
 order by col_order asc;
cr2 cursor FOR
SELECT  distinct table_name,table_name_stg
   FROM  meta_EXTERNAL_TABLE
   WHERE upper(file_name) = upper(pi_filename);


BEGIN
 -- PERFORM utl_file_utility.init();
   v_region := 'us-east-1';
   /* find tab details from file name */


   --DELETE FROM  ERROR_TABLE WHERE file_name= pi_filename;
  -- DELETE FROM  log_table WHERE file_name= pi_filename;


 BEGIN


   SELECT distinct table_name,table_name_stg INTO strict pi_ext_table,v_table_name_stg
   FROM  meta_EXTERNAL_TABLE
   WHERE upper(file_name) = upper(pi_filename);
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
    raise notice 'error 1,%',sqlerrm;
    pi_ext_table := null;
    v_table_name_stg := null;
      RAISE USING errcode = 'NTFIP' ;
    when others then
        raise notice 'error others,%',sqlerrm;
 END;
 j :=1 ;

for rec in  cr2
 LOOP
```

```
 pi_ext_table     := rec.table_name;
 v_table_name_stg := rec.table_name_stg;
 v_col_list := null;


IF pi_ext_table IS NOT NULL
 THEN
   --EXECUTE concat_ws('','truncate table  ' ,pi_ext_table) ;
  EXECUTE concat_ws('','truncate table  ' ,v_table_name_stg) ;




      SELECT distinct DELIMETER INTO STRICT v_delim
      FROM  meta_EXTERNAL_TABLE
      WHERE table_name = pi_ext_table;


      IF v_delim IS NOT NULL THEN
    SELECT distinct DELIMETER,
      FILE_NAME,
      DIRECTORY ,
      concat_ws('',' ',table_name_stg),
      case  header_exist when 'YES' then 'CSV HEADER' else 'CSV' end as header_exist
    INTO STRICT v_delim,v_file_name,v_directory,v_table_name_stg,v_header
    FROM  meta_EXTERNAL_TABLE
    WHERE table_name = pi_ext_table
      AND DELIMETER IS NOT NULL;


    IF    upper(v_delim) = 'CSV'
    THEN
      v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3 ( ''',
      v_table_name_stg,''','''',
      ''DELIMITER '''','''' CSV HEADER QUOTE '''''''''', aws_commons.create_s3_uri
( ''',
      v_directory,''','',v_file_name,''', ''',v_region,'''))');
      ELSE
      v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3(''',
          v_table_name_stg, ''','''', ''DELIMITER AS ''''^''''',''',','
         aws_commons.create_s3_uri
          ( ''',v_directory, ''','',
          v_file_name, ''',',
```

```
            ''',v_region,''')
         )');
         raise notice 'v_sql , %',v_sql;
      begin
       EXECUTE  v_sql;
      EXCEPTION
        WHEN OTHERS THEN
          raise notice 'error 1';
        RAISE USING errcode = 'S3IMP' ;
      END;


      select count(col_list) INTO v_col_cnt
      from  meta_EXTERNAL_TABLE where table_name = pi_ext_table;







       -- raise notice 'v_sql 2, %',concat_ws('','update ',v_table_name_stg, ' set
col1 = col1||''',v_delim,''''');


      execute concat_ws('','update ',v_table_name_stg, ' set col1 =
col1||''',v_delim,''''');




      i :=1;
      FOR rec in cr1
      loop
      v_sql1 := concat_ws('',v_sql1,'split_part(col1,''',v_delim,''',', i,')',' as
',rec.col_list,',');
      v_sql2 := concat_ws('',v_sql2,rec.col_list,',');
  --    v_sql3 := concat_ws('',v_sql3,'rec.',rec.col_list,'::',rec.data_type,',');


      case
        WHEN upper(rec.data_type) = 'NUMERIC'
        THEN v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
```

```
                ELSE
                coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))):NUMERIC,0)::',rec.data_type,' END as ',rec.col_list,',') ;
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDD'
        THEN v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))),''99990101''),''YYYYMMDD''):',rec.data_type,' END as ',rec.col_list,',');
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'MM/DD/YYYY hh24:mi:ss'
        THEN v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))),''01/01/9999 0024:00:00''),''MM/DD/YYYY hh24:mi:ss''):',rec.data_type,' END as
',rec.col_list,',');
         ELSE
       v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
                ELSE
                 coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))),''''):',rec.data_type,' END as ',rec.col_list,',') ;
      END case;


      i :=i+1;
      end loop;


        -- raise notice 'v_sql 3, %',v_sql3;


      SELECT trim(trailing ' ' FROM v_sql1) INTO v_sql1;
      SELECT trim(trailing ',' FROM v_sql1) INTO v_sql1;


      SELECT trim(trailing ' ' FROM v_sql2) INTO v_sql2;
      SELECT trim(trailing ',' FROM v_sql2) INTO v_sql2;
```

```
       SELECT trim(trailing ' ' FROM v_sql3) INTO v_sql3;
       SELECT trim(trailing ',' FROM v_sql3) INTO v_sql3;


       END IF;
      raise notice 'v_delim , %',v_delim;


     EXECUTE concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg)  INTO v_cnt;


    raise notice 'stg cnt , %',v_cnt;


    /* if upper(v_delim) = 'CSV' then
       v_sql_ins := concat_ws('', ' SELECT * from ' ,v_table_name_stg );
     else
      -- v_sql_ins := concat_ws('',' SELECT ',v_sql1,'  from (select col1 from
 ' ,v_table_name_stg , ')sub ');
       v_sql_ins := concat_ws('',' SELECT ',v_sql3,'  from (select col1 from
 ' ,v_table_name_stg , ')sub ');
       END IF;*/


v_chunk := v_cnt/100;




for i in 1..101
loop
     BEGIN
    -- raise notice 'v_sql , %',v_sql;
       -- raise notice 'Chunk number , %',i;
       v_sql_ins := concat_ws('',' SELECT ',v_sql3,'  from (select col1 from
 ' ,v_table_name_stg , ' offset ',v_chunk*(i-1), ' limit ',v_chunk,') sub ');


     v_sql := concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins);
     -- raise notice 'select statement , %',v_sql_ins;
          -- v_sql := null;
     -- EXECUTE concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins, 'offset
 ',v_chunk*(i-1), ' limit ',v_chunk );
```

```
    --v_sql := concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins );


    -- raise notice 'insert statement , %',v_sql;


   raise NOTICE 'CHUNK START %',v_chunk*(i-1);
  raise NOTICE 'CHUNK END %',v_chunk;


    EXECUTE v_sql;


 EXCEPTION
      WHEN OTHERS THEN
      -- v_sql_ins := concat_ws('',' SELECT ',v_sql1, '  from (select col1 from
' ,v_table_name_stg , ' )sub ');
        -- raise notice 'Chunk number for cursor , %',i;


   raise NOTICE 'Cursor - CHUNK START %',v_chunk*(i-1);
  raise NOTICE 'Cursor -  CHUNK END %',v_chunk;
        v_sql_ins := concat_ws('',' SELECT ',v_sql3, '  from (select col1 from
' ,v_table_name_stg , ' )sub ');


        v_final_sql := REPLACE (v_sql_ins, ''''::text, ''''''::text);
       -- raise notice 'v_final_sql %',v_final_sql;
        v_sql :=concat_ws('','do $a$ declare  r refcursor;v_sql text; i
numeric;v_conname text;  v_typ  ',pi_ext_table,'[]; v_rec  ','record',';
          begin







          open r for execute ''select col1 from ',v_table_name_stg ,'  offset
',v_chunk*(i-1), ' limit ',v_chunk,''';
          loop
          begin
          fetch r into v_rec;
          EXIT WHEN NOT FOUND;
```

```
          v_sql := concat_ws('''','insert into  ',pi_ext_table,' SELECT ',REPLACE
 (v_sql3, ''''::text, ''''''::text) , '  from ( select ''''''',v_rec.col1,''''''' as
 col1) v'');
           execute v_sql;


          exception
           when others then
         v_sql := ''INSERT INTO  ERROR_TABLE VALUES (concat_ws(''''''''','''''Error
 Name: ''''',$$''||SQLERRM||''$$,''''Error State: ''''',''''''||
SQLSTATE||''''''',''''record : ''''',$$''||v_rec.col1||''$$),'''''||
pi_filename||''''',now())'';


             execute v_sql;
            continue;
          end ;
          end loop;
          close r;
          exception
          when others then
        raise;
          end ; $a$');
     -- raise notice ' inside excp v_sql %',v_sql;
         execute v_sql;
     --  raise notice 'v_sql %',v_sql;
      END;
  END LOOP;
     ELSE


     SELECT distinct DELIMETER,FILE_NAME,DIRECTORY ,concat_ws('',' ',table_name_stg),
       case  header_exist when 'YES' then 'CSV HEADER' else 'CSV' end as header_exist
       INTO STRICT v_delim,v_file_name,v_directory,v_table_name_stg,v_header
     FROM  meta_EXTERNAL_TABLE
     WHERE table_name = pi_ext_table                  ;
     v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3(''',
       v_table_name_stg, ''','''', ''DELIMITER AS ''''#'''' ',v_header,' '','','
      aws_commons.create_s3_uri
       ( ''',v_directory, ''','''',
       v_file_name, ''',',
```

```
       ''''',v_region,''')
     )');
        EXECUTE  v_sql;


    FOR rec in cr1
    LOOP


     IF rec.start_pos IS NULL AND rec.END_pos IS NULL AND rec.no_position = 'recnum'
     THEN
       v_rec_val := 1;
     ELSE


      case
        WHEN upper(rec.data_type) = 'NUMERIC'
        THEN v_sql1 := concat_ws('',' case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1)))::NUMERIC,0)::',rec.data_type,' END as
',rec.col_list,',') ;
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDD'
        THEN v_sql1 := concat_ws('','case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1))),''99990101''),''YYYYMMDD''):',rec.data_type,'
END as ',rec.col_list,',');
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDDHH24MISS'
        THEN v_sql1 := concat_ws('','case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1))),''9999010100240000''),''YYYYMMDDHH24MISS''):',rec.data_
END as ',rec.col_list,',');
         ELSE
```

```
       v_sql1 := concat_ws('',' case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                 coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1))),''''):',rec.data_type,' END as
',rec.col_list,',') ;
      END case;


     END IF;
     v_col_list := concat_ws('',v_col_list ,v_sql1);
    END LOOP;




          SELECT trim(trailing ' ' FROM v_col_list) INTO v_col_list;
          SELECT trim(trailing ',' FROM v_col_list) INTO v_col_list;


          v_sql_col   :=  concat_ws('',trim(trailing ',' FROM v_col_list) , ' FROM
',v_table_name_stg,' WHERE col1 IS NOT NULL AND length(col1)>0 ');




          v_sql_dynamic := v_sql_col;


          EXECUTE  concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg) INTO v_cnt;




        IF v_rec_val = 1 THEN
            v_sql_ins := concat_ws('',' select row_number() over(order by ctid) as
line_number ,' ,v_sql_dynamic) ;


        ELSE
              v_sql_ins := concat_ws('',' SELECT' ,v_sql_dynamic) ;
          END IF;
```

```
    BEGIN
      EXECUTE concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins);
          EXCEPTION
             WHEN OTHERS THEN
         IF v_rec_val = 1 THEN
                v_final_sql := ' select row_number() over(order by ctid) as
line_number ,col1 from ';
               ELSE
                v_final_sql := ' SELECT col1 from';
              END IF;
      v_sql :=concat_ws('','do $a$ declare  r refcursor;v_rec_val numeric :=
',coalesce(v_rec_val,0),';line_number numeric; col1 text; v_typ  ',pi_ext_table,'[];
v_rec  ',pi_ext_table,';
            begin
            open r for execute ''',v_final_sql, ' ',v_table_name_stg,' WHERE col1 IS
NOT NULL AND length(col1)>0 '' ;
            loop
            begin
            if   v_rec_val = 1 then
            fetch r into line_number,col1;
            else
            fetch r into col1;
            end if;


            EXIT WHEN NOT FOUND;
             if v_rec_val = 1 then
             select line_number,',trim(trailing ',' FROM v_col_list) ,' into v_rec;
             else
               select ',trim(trailing ',' FROM v_col_list) ,' into v_rec;
             end if;


            insert into  ',pi_ext_table,' select v_rec.*;
             exception
             when others then
              INSERT INTO  ERROR_TABLE VALUES (concat_ws('''',''Error Name:
'',SQLERRM,''Error State: '',SQLSTATE,''record : '',v_rec),''',pi_filename,''',now());
              continue;
             end ;
              end loop;
            close r;
             exception
```

```
            when others then
            raise;
            end ; $a$');
        execute v_sql;


    END;


        END IF;


  EXECUTE concat_ws('','SELECT COUNT(*) FROM  ' ,pi_ext_table)   INTO proc_rec_COUNT;


  EXECUTE concat_ws('','SELECT COUNT(*) FROM  error_table WHERE file_name
='''',pi_filename,''' and processed_time::date = clock_timestamp()::date')  INTO
error_rec_COUNT;


  EXECUTE concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg)   INTO tot_rec_COUNT;


  INSERT INTO  log_table values(pi_filename,now(),tot_rec_COUNT,proc_rec_COUNT,
error_rec_COUNT);


  raise notice 'v_directory, %',v_directory;


  raise notice 'pi_filename, %',pi_filename;


  raise notice 'v_region, %',v_region;


 perform aws_s3.query_export_to_s3('SELECT
replace(trim(substring(error_details,position(''('' in
error_details)+1),'')''),'','','';''),file_name,processed_time FROM  error_table WHERE
file_name = '''||pi_filename||'''',
  aws_commons.create_s3_uri(v_directory, pi_filename||'.bad', v_region),
  options :='FORmat csv, header, delimiter $$,$$'
  );
```

```
raise notice 'v_directory, %',v_directory;


   raise notice 'pi_filename, %',pi_filename;


   raise notice 'v_region, %',v_region;


  perform aws_s3.query_export_to_s3('SELECT * FROM  log_table WHERE file_name = '''||
pi_filename||'''',
   aws_commons.create_s3_uri(v_directory, pi_filename||'.log', v_region),
   options :='FORmat csv, header, delimiter $$,$$'
   );




   END IF;
 j := j+1;
 END LOOP;


       RETURN 'OK';
EXCEPTION
    WHEN  OTHERS THEN
  raise notice 'error %',sqlerrm;
   ERCODE=SQLSTATE;
   IF ERCODE = 'NTFIP' THEN
     v_sqlerrm := concat_Ws('',sqlerrm,'No data for the filename');
   ELSIF ERCODE = 'S3IMP' THEN
    v_sqlerrm := concat_Ws('',sqlerrm,'Error While exporting the file from S3');
   ELSE
      v_sqlerrm := sqlerrm;
   END IF;


 select distinct directory into v_directory from  meta_EXTERNAL_TABLE;




 raise notice 'exc v_directory, %',v_directory;
```

```
   raise notice 'exc pi_filename, %',pi_filename;


   raise notice 'exc v_region, %',v_region;


  perform aws_s3.query_export_to_s3('SELECT * FROM  error_table WHERE file_name = '''||
pi_filename||'''',
   aws_commons.create_s3_uri(v_directory, pi_filename||'.bad', v_region),
   options :='FORmat csv, header, delimiter $$,$$'
   );
    RETURN null;
END;
$function$
```

Migrasikan indeks berbasis fungsi dari Oracle ke PostgreSQL

Veeranjaneyulu Grandhi dan Navakanth Talluri, Amazon Web Services

Ringkasan

Indeks adalah cara umum untuk meningkatkan kinerja database. Indeks memungkinkan server database untuk menemukan dan mengambil baris tertentu jauh lebih cepat daripada yang bisa tanpa indeks. Tetapi indeks juga menambahkan overhead ke sistem database secara keseluruhan, sehingga mereka harus digunakan dengan bijaksana. Indeks berbasis fungsi, yang didasarkan pada fungsi atau ekspresi, dapat melibatkan beberapa kolom dan ekspresi matematika. Indeks berbasis fungsi meningkatkan kinerja kueri yang menggunakan ekspresi indeks.

Secara native, PostgreSQL tidak mendukung pembuatan indeks berbasis fungsi menggunakan fungsi yang memiliki volatilitas yang didefinisikan sebagai stabil. Namun, Anda dapat membuat fungsi serupa dengan volatilitas `IMMUTABLE` dan menggunakannya dalam pembuatan indeks.

Sebuah `IMMUTABLE` fungsi tidak dapat memodifikasi database, dan itu dijamin untuk mengembalikan hasil yang sama diberikan argumen yang sama selamanya. Kategori ini memungkinkan pengoptimal untuk melakukan pra-evaluasi fungsi saat kueri memanggilnya dengan argumen konstan.

Pola ini membantu dalam memigrasikan indeks berbasis fungsi Oracle saat digunakan dengan fungsi seperti,`to_char`, `to_date` dan ke PostgreSQL yang setara. `to_number`

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif
- Sebuah instance database Oracle sumber dengan layanan listener yang disiapkan dan dijalankan
- Keakraban dengan database PostgreSQL

Batasan

- Batas ukuran database adalah 64 TB.
- Fungsi yang digunakan dalam pembuatan indeks harus IMMUTABLE.

Versi produk

- Semua edisi database Oracle untuk versi 11g (versi 11.2.0.3.v1 dan yang lebih baru) dan hingga 12.2, dan 18c
- PostgreSQL versi 9.6 dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database Oracle di tempat atau di instans Amazon Elastic Compute Cloud EC2 (Amazon), atau Amazon RDS for Oracle DB instans

Tumpukan teknologi target

- Mesin PostgreSQL apa pun

Alat

- pgAdmin 4 adalah alat manajemen open source untuk Postgres. Alat pgAdmin 4 menyediakan antarmuka grafis untuk membuat, memelihara, dan menggunakan objek database.
- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi (IDE) untuk mengembangkan dan mengelola Oracle Database baik dalam penerapan tradisional maupun cloud.

Epik

Buat indeks berbasis fungsi menggunakan fungsi default

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat indeks berbasis fungsi pada kolom menggunakan fungsi to_char.	Gunakan kode berikut untuk membuat indeks berbasis fungsi.
	`postgres=# create table funcindex( col1 timestamp without time zone);`
	`CREATE TABLE`
		DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`postgres=# insert into funcindex values (now()); INSERT 0 1 postgres=# select * from funcindex; col1 ---------- 2022-08-09 16:00:57.77414 (1 rows) postgres=# create index funcindex_idx on funcindex(to_char(col1,'DD-MM-YYYY HH24:MI:SS')); ERROR: functions in index expression must be marked IMMUTABLE` **Note** PostgreSQL tidak mengizinkan pembuatan indeks berbasis fungsi tanpa klausa. `IMMUTABLE`	
Periksa volatilitas fungsi.	Untuk memeriksa volatilitas fungsi, gunakan kode di bagian Informasi tambahan.	DBA

Buat indeks berbasis fungsi menggunakan fungsi pembungkus

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi pembungkus.	Untuk membuat fungsi pembungkus, gunakan kode di bagian Informasi tambahan.	Pengembang PostgreSQL
Buat indeks dengan menggunakan fungsi pembungkus.	Gunakan kode di bagian Informasi tambahan untuk membuat fungsi yang ditentukan pengguna dengan kata kunci `IMMUTABLE` dalam skema yang sama dengan aplikasi, dan merujuknya dalam skrip pembuatan indeks. Jika fungsi yang ditentukan pengguna dibuat dalam skema umum (dari contoh sebelumnya), perbarui seperti yang `search_path` ditunjukkan. `ALTER ROLE <ROLENAME> set search_path=$user, COMMON;`	DBA, pengembang PostgreSQL

Validasi pembuatan indeks

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi pembuatan indeks.	Validasi bahwa indeks perlu dibuat, berdasarkan pola akses kueri.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi bahwa indeks dapat digunakan.	Untuk memeriksa apakah indeks berbasis fungsi diambil oleh PostgreSQL Optimizer , jalankan pernyataan SQL menggunakan menjelaskan atau menjelaskan analisis. Gunakan kode di bagian Informasi tambahan. Jika memungkinkan, kumpulkan juga statistik tabel. **Note** Jika Anda melihat rencana penjelasan, pengoptimal PostgreSQL telah memilih indeks berbasis fungsi karena kondisi predikat.	DBA

Sumber daya terkait

- Indeks berbasis fungsi (dokumentasi Oracle)
- Indeks pada Ekspresi (dokumentasi PostgreSQL)
- Volatilitas PostgreSQL (dokumentasi PostgreSQL)
- PostgreSQL search_path (dokumentasi PostgreSQL)
- Oracle Database 19c ke Amazon Aurora PostgreSQL Playbook Migrasi

Informasi tambahan

Buat fungsi pembungkus

```
CREATE OR REPLACE FUNCTION myschema.to_char(var1 timestamp without time zone, var2
 varchar) RETURNS varchar AS $BODY$ select to_char(var1, 'YYYYMMDD'); $BODY$ LANGUAGE
 sql IMMUTABLE;
```

Buat indeks dengan menggunakan fungsi pembungkus

```
postgres=# create function common.to_char(var1 timestamp without time zone, var2
 varchar) RETURNS varchar AS $BODY$ select to_char(var1, 'YYYYMMDD'); $BODY$ LANGUAGE
 sql IMMUTABLE;
CREATE FUNCTION
postgres=# create index funcindex_idx on funcindex(common.to_char(col1,'DD-MM-YYYY
 HH24:MI:SS'));
CREATE INDEX
```

Periksa volatilitas fungsi

```
SELECT DISTINCT p.proname as "Name",p.provolatile as "volatility" FROM
 pg_catalog.pg_proc p
 LEFT JOIN pg_catalog.pg_namespace n ON n.oid = p.pronamespace
 LEFT JOIN pg_catalog.pg_language l ON l.oid = p.prolang
 WHERE n.nspname OPERATOR(pg_catalog.~) '^(pg_catalog)$' COLLATE pg_catalog.default AND
 p.proname='to_char'GROUP BY p.proname,p.provolatile
ORDER BY 1;
```

Validasi bahwa indeks dapat digunakan

```
explain analyze <SQL>


postgres=# explain select col1 from funcindex where common.to_char(col1,'DD-MM-YYYY
 HH24:MI:SS') = '09-08-2022 16:00:57';
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------
 Index Scan using funcindex_idx on funcindex  (cost=0.42..8.44 rows=1 width=8)
   Index Cond: ((common.to_char(col1, 'DD-MM-YYYY HH24:MI:SS'::character
 varying))::text = '09-08-2022 16:00:57'::text)
(2 rows)
```

Migrasikan fungsi asli Oracle ke PostgreSQL menggunakan ekstensi

Pinesh Singal, Amazon Web Services

Ringkasan

Pola migrasi ini memberikan step-by-step panduan untuk memigrasikan Amazon Relational Database Service (Amazon RDS) untuk instance database Oracle ke Amazon RDS for PostgreSQL atau database Amazon Aurora PostgreSQL Compatible Edition dengan memodifikasi dan ekstensi ke kode bawaan bawaan PostgreSQL () bawaan bawaan. `aws_oracle_ext` `orafce` `psql` Ini akan menghemat waktu pemrosesan.

Pola ini menggambarkan strategi migrasi manual offline tanpa downtime untuk database sumber Oracle multi-terabyte dengan jumlah transaksi yang tinggi.

Proses migrasi menggunakan AWS Schema Conversion Tool (AWS SCT) dengan `aws_oracle_ext` ekstensi `orafce` dan untuk mengonversi skema database Amazon RDS for Oracle ke Amazon RDS for PostgreSQL atau skema database yang kompatibel dengan Aurora PostgreSQL. Kemudian kode secara manual diubah menjadi PostgreSQL didukung kode bawaan asli. `psql` Ini karena ekstensi memanggil pemrosesan kode dampak pada server database PostgreSQL, dan tidak semua kode ekstensi sepenuhnya dikeluhkan atau kompatibel dengan kode PostgreSQL.

Pola ini terutama berfokus pada migrasi kode SQL secara manual menggunakan AWS SCT dan ekstensi dan. `aws_oracle_ext` `orafce` Anda mengonversi ekstensi yang sudah digunakan menjadi bawaan `psql` PostgreSQL () bawaan bawaan. Kemudian Anda menghapus semua referensi ke ekstensi dan mengonversi kode yang sesuai.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sistem operasi (Windows atau Mac) atau EC2 instans Amazon (aktif dan berjalan)
- Orafce

Batasan

Tidak semua fungsi Oracle menggunakan `aws_oracle_ext` atau `orafce` ekstensi dapat dikonversi ke fungsi PostgreSQL asli. Mungkin perlu pengerjaan ulang manual untuk mengompilasinya dengan pustaka PostgreSQL.

Salah satu kelemahan menggunakan ekstensi AWS SCT adalah kinerjanya yang lambat dalam menjalankan dan mengambil hasilnya. Biayanya dapat dipahami dari rencana PostgreSQL EXPLORE sederhana (rencana eksekusi pernyataan) pada migrasi fungsi *SYSDATE* Oracle ke fungsi *NOW()* PostgreSQL antara ketiga *aws_oracle_ext* kode *orafce* (,, dan default)*psql*, seperti yang dijelaskan di bagian pemeriksaan perbandingan Kinerja dalam dokumen terlampir.

Versi produk

- Sumber: Amazon RDS for Oracle database 10.2 dan yang lebih baru (untuk 10.x), 11g (11.2.0.3.v1 dan yang lebih baru) dan hingga 12.2, 18c, dan 19c (dan yang lebih baru) untuk Enterprise Edition, Standard Edition, Standard Edition 1, dan Standard Edition 2
- Target: Amazon RDS untuk database yang kompatibel dengan PostgreSQL atau Aurora PostgreSQL 9.4 dan yang lebih baru (untuk 9.x), 10.x, 11.x, 12.x, 13.x, dan 14.x (dan yang lebih baru)
- AWS SCT: Versi terbaru (pola ini diuji dengan 1.0.632)
- Orafce: Versi terbaru (pola ini diuji dengan 3.9.0)

Arsitektur

Tumpukan teknologi sumber

- Instans database Amazon RDS for Oracle dengan versi 12.1.0.2.v18

Tumpukan teknologi target

- Instans database yang kompatibel dengan Amazon RDS for PostgreSQL atau Aurora PostgreSQL dengan versi 11.5

Arsitektur migrasi database

Diagram berikut merupakan arsitektur migrasi database antara sumber Oracle dan database PostgreSQL target. Arsitekturnya melibatkan AWS Cloud, virtual private cloud (VPC), Availability Zones, subnet pribadi, database Amazon RDS for Oracle, AWS SCT, Amazon RDS untuk

PostgreSQL atau database yang kompatibel dengan Aurora PostgreSQL, ekstensi untuk file Oracle (dan), dan bahasa kueri terstruktur (SQL). `aws_oracle_ext` `orafce`

1. Luncurkan Amazon RDS for Oracle DB instance (sumber DB).
2. Gunakan AWS SCT dengan paket `orafce` ekstensi `aws_oracle_ext` dan untuk mengonversi kode sumber dari Oracle ke PostreSQL.
3. Konversi menghasilkan file.sql bermigrasi yang didukung PostgreSQL.
4. Secara manual mengonversi kode ekstensi Oracle yang tidak dikonversi ke kode PostgreSQL (). `psql`
5. Konversi manual menghasilkan file.sql dikonversi yang didukung PostgreSQL.
6. Jalankan file.sql ini di Amazon RDS untuk instans DB Amazon RDS for PostgreSQL (target DB).

Alat

Alat

Layanan AWS

- AWS SCT - AWS Schema Conversion Tool (AWS SCT) mengonversi skema database Anda yang ada dari satu mesin database ke mesin database lainnya. Anda dapat mengonversi skema Pemrosesan Transaksional Online (OLTP) relasional, atau skema gudang data. Skema konversi Anda cocok untuk instans Amazon RDS for MySQL DB, kluster Amazon Aurora DB, instans Amazon RDS for PostgreSQL DB, atau cluster Amazon Redshift. Skema yang dikonversi juga dapat digunakan dengan database pada EC2 instans Amazon atau disimpan sebagai data dalam bucket Amazon S3.

  AWS SCT menyediakan antarmuka pengguna berbasis proyek untuk secara otomatis mengonversi skema database database sumber Anda menjadi format yang kompatibel dengan instans Amazon RDS target Anda.

  Anda dapat menggunakan AWS SCT untuk melakukan migrasi dari database sumber Oracle ke salah satu target yang tercantum sebelumnya. Menggunakan AWS SCT, Anda dapat mengekspor definisi objek database sumber seperti skema, tampilan, prosedur tersimpan, dan fungsi.

  Anda dapat menggunakan AWS SCT untuk mengonversi data dari Oracle ke Amazon RDS untuk PostgreSQL atau Amazon Aurora PostgreSQL Edisi yang kompatibel dengan Amazon Aurora.

Dalam pola ini, Anda menggunakan AWS SCT untuk mengonversi dan memigrasikan kode Oracle ke PostgreSQL menggunakan ekstensi `aws_oracle_ext` dan`orafce`, dan memigrasikan kode ekstensi secara manual ke kode bawaan default atau asli. `psql`

- Paket ekstensi AWS SCT adalah modul add-on yang mengemulasi fungsi yang ada dalam database sumber yang diperlukan saat mengonversi objek ke database target. Sebelum Anda dapat menginstal paket ekstensi AWS SCT, Anda perlu mengonversi skema database Anda.

  Saat Anda mengonversi skema database atau gudang data, AWS SCT menambahkan skema tambahan ke basis data target Anda. Skema ini mengimplementasikan fungsi sistem SQL dari database sumber yang diperlukan saat menulis skema yang dikonversi ke database target Anda. Skema tambahan ini disebut skema paket ekstensi.

  Skema paket ekstensi untuk database OLTP diberi nama sesuai dengan database sumber. Untuk database Oracle, skema paket ekstensi adalah. `AWS_ORACLE_EXT`

Alat-alat lainnya

- Orafce - Orafce adalah modul yang mengimplementasikan fungsi, tipe data, dan paket yang kompatibel dengan Oracle. Ini adalah alat sumber terbuka dengan lisensi Berkeley Source Distribution (BSD) sehingga siapa pun dapat menggunakannya. `orafce`Modul ini berguna untuk migrasi dari Oracle ke PostgreSQL karena memiliki banyak fungsi Oracle yang diimplementasikan di PostgreSQL.

Kode

Untuk daftar semua kode yang umum digunakan dan dimigrasi dari Oracle ke PostgreSQL guna menghindari penggunaan kode ekstensi AWS SCT, lihat dokumen terlampir.

Epik

Konfigurasikan database sumber Amazon RDS for Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance database Oracle.	Buat instance database yang kompatibel dengan Amazon	AWS Umum, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	RDS for Oracle atau Aurora PostgreSQL dari konsol Amazon RDS.	
Konfigurasikan grup keamanan.	Konfigurasikan grup keamanan masuk dan keluar.	AWS Umum
Buat database.	Buat database Oracle dengan pengguna dan skema yang dibutuhkan.	AWS Umum, DBA
Buat objek.	Buat objek dan masukkan data dalam skema.	DBA

Konfigurasikan database target Amazon RDS for PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance database PostgreSQL.	Buat instance database Amazon RDS for PostgreSQ L atau Amazon Aurora PostgreSQL dari konsol Amazon RDS.	AWS Umum, DBA
Konfigurasikan grup keamanan.	Konfigurasikan grup keamanan masuk dan keluar.	AWS Umum
Buat database.	Buat database PostgreSQL dengan pengguna dan skema yang dibutuhkan.	AWS Umum, DBA
Validasi ekstensi.	Pastikan bahwa `aws_oracl e_ext` dan diinstal dan `orafce` dikonfigurasi dengan benar di database PostgreSQ L.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa database PostgreSQL tersedia.	Pastikan database PostgreSQL aktif dan berjalan.	DBA

Migrasikan skema Oracle ke PostgreSQL menggunakan AWS SCT dan ekstensi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS SCT.	Instal AWS SCT versi terbaru.	DBA
Konfigurasikan AWS SCT.	Konfigurasikan AWS SCT dengan driver Java Database Connectivity (JDBC) untuk Oracle () `ojdbc8.jar` dan PostgreSQL (). `postgresql-42.2.5.jar`	DBA
Aktifkan paket atau templat ekstensi AWS SCT.	Di bawah AWS SCT Project Settings, aktifkan implementasi fungsi bawaan dengan `aws_oracle_ext` dan `orafce` ekstensi untuk skema database Oracle.	DBA
Konversi skema.	Di AWS SCT, pilih Konversi Skema untuk mengonversi skema dari Oracle ke PostgreSQL dan buat file.sql.	DBA

Ubah kode ekstensi AWS SCT ke kode psql

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konversi kode secara manual.	Secara manual mengonversi setiap baris kode yang	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	didukung ekstensi menjadi kode bawaan `psql` default, seperti yang dijelaskan dalam dokumen terlampir. Misalnya, ubah `AWS_ORACLE_EXT.SYSDATE()` atau `ORACLE.SYSDATE()` ke`NOW()`.	
Validasi kode	(Opsional) Validasi setiap baris kode dengan menjalankannya sementara di database PostgreSQL.	DBA
Buat objek dalam database PostgreSQL.	Untuk membuat objek dalam database PostgreSQL, jalankan file.sql yang dihasilkan oleh AWS SCT dan dimodifikasi dalam dua langkah sebelumnya.	DBA

Sumber daya terkait

- Basis Data
  - Oracle di Amazon RDS
  - PostgreSQL di Amazon RDS
  - Bekerja dengan Amazon Aurora PostgreSQL
  - PostgreSQL JELASKAN rencana
- AWS SCT
  - Ikhtisar Alat Konversi Schema AWS
  - Panduan Pengguna AWS SCT
  - Menggunakan antarmuka pengguna AWS SCT
  - Menggunakan Oracle Database sebagai sumber AWS SCT

- Ekstensi untuk AWS SCT
  - Menggunakan paket ekstensi AWS SCT
  - Fungsionalitas Oracle
  - PGXN orafce
  - GitHub orafce

Informasi tambahan

Untuk informasi lebih lanjut, ikuti perintah terperinci, dengan sintaks dan contoh, untuk mengonversi kode secara manual dalam dokumen terlampir.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Migrasikan database Db2 dari Amazon ke EC2 Aurora MySQL yang kompatibel dengan menggunakan AWS DMS

Pinesh Singal, Amazon Web Services

Ringkasan

Setelah memigrasikan database IBM Db2 untuk LUW ke Amazon Elastic Compute Cloud (EC2Amazon), pertimbangkan untuk merancang ulang database dengan pindah ke database cloud-native Amazon Web Services (AWS). Pola ini mencakup migrasi database IBM Db2 untuk LUW yang berjalan pada instans Amazon ke database Amazon Aurora MySQL EC2 Edition yang kompatibel dengan AWS.

Pola ini menggambarkan strategi migrasi online dengan downtime minimal untuk database sumber Db2 multi-terabyte dengan jumlah transaksi yang tinggi.

Pola ini menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema database Db2 menjadi skema yang kompatibel dengan Aurora MySQL. Kemudian pola menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data dari database Db2 ke database yang kompatibel dengan Aurora MySQL. Konversi manual akan diperlukan untuk kode yang tidak dikonversi oleh AWS SCT.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif dengan virtual private cloud (VPC)
- AWS SCT
- AWS DMS

Versi produk

- AWS SCT versi terbaru
- Db2 untuk Linux versi 11.1.4.4 dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- DB2/Linux x86-64 bit dipasang pada sebuah instance EC2

Tumpukan teknologi target

- Instans database Edisi yang kompatibel dengan Amazon Aurora MySQL

Arsitektur sumber dan target

Diagram berikut menunjukkan arsitektur migrasi data antara sumber Db2 dan target database yang kompatibel dengan Aurora MySQL. Arsitektur pada AWS Cloud mencakup virtual private cloud (VPC) (Virtual Private Cloud), Availability Zone, subnet publik untuk instans Db2 dan instans replikasi AWS DMS, dan subnet pribadi untuk database yang kompatibel dengan Aurora MySQL.

Alat

Layanan AWS

- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target. AWS SCT mendukung sebagai sumber IBM Db2 untuk LUW versi 9.1, 9.5, 9.7, 10.1, 10.5, 11.1, dan 11.5.

Praktik terbaik

Untuk praktik terbaik, lihat Praktik terbaik untuk AWS Database Migration Service.

Epik

Konfigurasikan sumber database IBM Db2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database IBM Db2 di Amazon. EC2	Anda dapat membuat database IBM Db2 pada EC2 instans dengan menggunakan Amazon Machine Image (AMI) dari AWS Marketplace atau dengan menginstal perangkat lunak Db2 pada instans. EC2 Luncurkan EC2 instance dengan memilih AMI untuk IBM Db2 (misalnya, IBM Db2 v11.5.7 RHEL 7.9), yang mirip dengan database lokal.	DBA, AWS Umum
Konfigurasikan grup keamanan.	Konfigurasikan aturan masuk grup keamanan VPC untuk SSH (Secure Shell) dan TCP dengan port 22 dan 50000, masing-masing.	AWS Umum
Buat instance database.	Buat instance baru (pengguna) dan database (skema), atau gunakan `db2inst1` contoh default dan database sampel. 1. Connect ke EC2 instance dengan menggunakan terminal untuk terhubung ke database Db2. Atau, Anda dapat menginstal perangkat lunak klien DB apa pun	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang akan terhubung ke database Db2. 2. Untuk mengatur kata sandi pengguna db2inst1, jalankan perintah. `sudo passwd db2inst1` 3. Untuk terhubung ke instance db2inst1, jalankan perintah. `sudo su - db2inst1` 4. Untuk terhubung ke database Db2, jalankan perintah. `db2` 5. Untuk terhubung ke database sampel, gunakan perintah`connect to sample`. Atau, sambungkan ke database yang Anda buat. 6. Setelah Anda terhubung ke instance database, buat objek dan masukkan data ke dalam objek ini dengan menggunakan pernyataan SQL Db2.	
Konfirmasikan bahwa instans Db2 DB tersedia.	Untuk mengonfirmasi bahwa instance database Db2 aktif dan berjalan, gunakan perintah. `Db2pd -`	DBA

Konfigurasikan database target yang kompatibel dengan Aurora MySQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database yang kompatibel dengan Aurora MySQL.	Buat Amazon Aurora dengan Database kompatibilitas MySQL dari layanan AWS RDS • Buat database di Amazon Aurora dengan kompatibi litas MySQL dan versi pilihan Anda, misalnya Aurora (MySQL) —5.6.10a • Instal aplikasi MySQL Workbench atau perangkat lunak klien DB pilihan Anda yang memungkinkan Anda terhubung ke database MySQL	DBA, AWS Umum
Konfigurasikan grup keamanan.	Konfigurasikan aturan masuk grup keamanan VPC untuk koneksi SSH dan TCP.	AWS Umum
Konfirmasikan bahwa database Aurora tersedia.	Untuk memastikan bahwa database yang kompatibel dengan Aurora MySQL aktif dan berjalan, lakukan hal berikut: 1. Connect ke EC2 instance melalui SSH. 2. Konfigurasikan dan sambungkan ke instance yang kompatibel dengan Aurora MySQL dari MySQL Workbench. Gunakan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	endpoint sebagai nama host, seperti yang ditunjukkan pada contoh berikut. `mysql-cluster-instance-1.cokmvis0v46q.us-east-1.rds.amazonaws.com` 3. Buat dan sambungkan ke skema baru (misalnya ,`mysql-sample-db2`). 4. Jalankan pernyataan MySQL untuk memeriksa skema dan objek dalam database.	

Konfigurasikan dan jalankan AWS SCT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS SCT.	Unduh dan instal versi terbaru AWS SCT (versi terbaru saat ini 1.0.628).	AWS Umum
Konfigurasikan AWS SCT.	1. Unduh driver Java Database Connectivity (JDBC) untuk IBM Db2 (versi 4.22.X) dan MySQL (8.x). 2. Untuk mengonfigurasi driver di AWS SCT, pilih Pengaturan, Pengaturan global, Driver.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat proyek AWS SCT.	Buat proyek AWS SCT dan laporkan yang menggunakan Db2 untuk LUW sebagai mesin DB sumber dan Aurora MySQL yang kompatibel dengan mesin DB target. Untuk mengidentifikasi hak istimewa yang diperlukan untuk terhubung ke database Db2 untuk LUW, lihat Menggunakan Db2 LUW sebagai sumber AWS SCT.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi objek.	Pilih skema Muat, validasi objek. Perbarui objek yang salah pada database target: 1. Sambungkan ke server Amazon Aurora MySQL yang kompatibel dengan menyediakan detail koneksi, dan pilih Uji koneksi. Koneksi sumber dan target harus berhasil sebelum AWS SCT dapat memulai laporan migrasi. 2. Setelah laporan selesai, masukkan skema yang akan dikonversi, dan pilih Selesai. AWS SCT mencantumkan objek sumber dan target apa pun yang dikonversi dan memiliki kesalahan. 3. Tinjau kesalahan, dan bersihkan secara manual. 4. Setelah semua kesalahan dihapus, buka menu konteks (klik kanan) untuk skema, dan pilih Muat skema. 5. Pilih Terapkan ke database. 6. Di MySQL Workbench, sambungkan ke database	DBA, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang kompatibel dengan Aurora MySQL, dan periksa skema dan objek.	

Konfigurasikan dan jalankan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh replikasi.	Masuk ke AWS Management Console, navigasikan ke layanan AWS DMS, dan buat instance replikasi dengan pengaturan yang valid untuk grup keamanan VPC yang Anda konfigurasikan untuk basis data sumber dan target.	AWS Umum
Buat titik akhir.	Buat titik akhir sumber untuk database Db2, dan buat titik akhir target untuk database yang kompatibel dengan Aurora MySQL: 1. Buat endpoint untuk IBM Db2 sebagai sumber dengan memilih Select RDS DB instance dan kemudian memilih instance Db2 yang Anda buat. Detail konfigurasi titik akhir akan diisi secara otomatis. 2. Dalam pengaturan khusus titik akhir, tambahkan atribut koneksi tambahan berikut.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`CurrentLSN=<scan>; MaxKBytesPerRead=64; SetDataCaptureChanges=true` Jika Anda tidak menyebutkan atribut ini, koneksi pengujian titik akhir sumber tidak akan berhasil. Untuk informasi selengkapnya, lihat Menggunakan IBM Db2 LUW sebagai sumber AWS DMS. 3. Buat titik akhir untuk Aurora MySQL yang kompatibel dengan target dengan memilih instans Select RDS DB dan kemudian memilih instance Aurora MySQL yang kompatibel dengan yang Anda buat. Detail konfigurasi titik akhir akan diisi secara otomatis. Untuk informasi selengkapnya, lihat Menggunakan database yang kompatibel dengan MySQL sebagai target AWS Database Migration Service. 4. Uji sumber dan titik akhir target. Konfirmasikan bahwa keduanya berhasil dan tersedia	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Jika pengujian gagal, pastikan aturan masuk grup keamanan valid.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas migrasi.	Buat tugas migrasi tunggal atau beberapa tugas migrasi untuk beban penuh dan validasi CDC atau Data: 1. Untuk membuat tugas migrasi database, pilih contoh replikasi, titik akhir basis data sumber, titik akhir basis data target. Tentukan jenis migrasi sebagai Migrasikan data yang ada (beban penuh), Replikasi perubahan data saja (CDC), atau Migrasi data yang ada dan replikasi perubahan yang sedang berlangsung (beban penuh dan CDC). 2. Di bawah Pemetaan tabel, Anda dapat mengonfigurasi aturan pemilihan dan aturan transformasi dalam format GUI atau JSON. 3. Di bawah Aturan pemilihan , pilih skema, masukkan nama tabel, dan pilih Tindakan (Sertakan/ Kecualikan) yang akan dikonfigurasi (misalnya , Skema: SAMPEL; Nama tabel:%, Tindakan: Sertakan).	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Di bawah Aturan transformasi, pilih target (Skema, Tabel, atau Kolom). Pilih nama skema, dan pilih tindakan (kasus, awalan, akhiran); misalnya, Target: Skema;`mysql-sample-db`; Tindakan: Buat huruf kecil. 5. Aktifkan pemantauan Amazon CloudWatch Logs.	
Rencanakan proses produksi.	Konfirmasikan waktu henti dengan pemangku kepentingan seperti pemilik aplikasi untuk menjalankan AWS DMS dalam sistem produksi.	Pimpin migrasi
Jalankan tugas migrasi.	1. Mulai tugas AWS DMS yang memiliki status Siap. 2. Pantau log tugas migrasi di Amazon CloudWatch Logs untuk mengetahui kesalahan apa pun.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi data.	Tinjau hasil tugas migrasi dan data di sumber Db2 dan target database MySQL: 1. Jika statusnya Memuat replikasi berkelanjutan lengkap, beban penuh dengan migrasi data CDC selesai, dan validasi sedang berlangsung. 2. Connect ke database yang kompatibel dengan Aurora MySQL, dan periksa datanya. 3. Periksa perubahan yang sedang berlangsung dengan memasukkan atau memperbarui data dalam database Db2.	DBA
Hentikan tugas migrasi.	Setelah validasi data berhasil diselesaikan, hentikan tugas migrasi validasi.	AWS Umum

Pemecahan masalah

Isu	Solusi
Sumber AWS SCT dan koneksi pengujian target gagal.	Konfigurasikan versi driver JDBC dan aturan masuk grup keamanan VPC untuk menerima lalu lintas masuk.
Uji titik akhir sumber Db2 gagal.	Konfigurasikan pengaturan koneksi tambahan`CurrentLSN=<scan>;` .

Isu	Solusi

AWSDMS Tugas gagal terhubung ke sumber Db2, dan kesalahan berikut dikembalikan.
 	
`database is recoverable if either or both of the database configuration parameters LOGARCHMETH1 and LOGARCHMETH2 are set to ON`
	Untuk menghindari kesalahan, jalankan perintah berikut:
	
	1. `$ db2 update db cfg for sample using LOGARCHMETH1 DISK:/home/db2inst1/logs`
	2. `$ db2stop`
	3. `$ db2start`
	4. `$ db2 connect to sample`
	
	`SQL1116N A connection to or activation of database "SAMPLE" cannot be made because of BACKUP PENDING. SQLSTATE=57019`
	
	5. `$ db2 backup database sample to ../logs`
	
	`SQL2036N The path for the file or device "../logs" is not valid`
	
	6. `$ cd`
	7. `$ pwd`
	
	`/home/db2inst1`
	
	8. `$ mkdir /tmp/backup`
	9. `$ db2 backup database sample to /tmp/backup`
	
	`Backup successful. The timestamp for this backup image is : 20190530084921`
	
	10. `$ db2 connect to sample`
	
	`Database Connection Information`

Isu	Solusi

	`Database server = DB2/LINUX 9.7.1`
	`SQL authorization ID = DB2INST1`
	`Local database alias = SAMPLE`

Sumber daya terkait

Amazon EC2

- Amazon EC2
- Panduan EC2 Pengguna Amazon

Basis Data

- Basis Data IBM Db2
- Amazon Aurora
- Bekerja dengan Amazon Aurora MySQL

AWS SCT

- Konversi Skema AWS DMS
- Panduan Pengguna Alat Konversi Skema AWS
- Menggunakan antarmuka pengguna AWS SCT
- Menggunakan IBM Db2 LUW sebagai sumber AWS SCT

AWS DMS

- AWS Database Migration Service
- Panduan Pengguna AWS Database Migration Service
- Sumber untuk migrasi data
- Target untuk migrasi data
- AWS Database Migration Service dan AWS Schema Conversion Tool sekarang mendukung IBM Db2 LUW sebagai sumber (posting blog)

- Migrasi Aplikasi Menjalankan Database Relasional ke AWS

Migrasi database Microsoft SQL Server dari Amazon EC2 ke Amazon DocumentDB dengan menggunakan AWS DMS

Perusahaan Nooka, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan database Microsoft SQL Server yang dihosting di instans Amazon Elastic Compute Cloud (Amazon EC2) ke database Amazon DocumentDB (dengan kompatibilitas MongoDB).

Tugas replikasi AWS DMS membaca struktur tabel database SQL Server, membuat koleksi yang sesuai di Amazon DocumentDB, dan melakukan migrasi beban penuh.

Anda juga dapat menggunakan pola ini untuk memigrasikan SQL Server lokal atau Amazon Relational Database Service (Amazon RDS) untuk instans SQL Server DB ke Amazon DocumentDB. Untuk informasi selengkapnya, lihat panduan Memigrasi database Microsoft SQL Server ke AWS Cloud di situs web AWS Prescriptive Guidance.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Database SQL Server yang ada pada sebuah EC2 instance.
- Peran database tetap (db_owner) yang ditetapkan ke AWS DMS dalam database SQL Server. Untuk informasi selengkapnya, lihat Peran tingkat database dalam dokumentasi SQL Server.
- Keakraban dengan menggunakan`mongodump`,, `mongorestore``mongoexport`, dan `mongoimport` utilitas untuk memindahkan data masuk dan keluar dari cluster Amazon DocumentDB.
- Microsoft SQL Server Management Studio, diinstal dan dikonfigurasi.

Batasan

- Batas ukuran cluster di Amazon DocumentDB adalah 64 TB. Untuk informasi selengkapnya, lihat Batas klaster dalam dokumentasi Amazon DocumentDB.
- AWS DMS tidak mendukung penggabungan beberapa tabel sumber ke dalam satu koleksi Amazon DocumentDB.

- Jika AWS DMS memproses perubahan apa pun dari tabel sumber tanpa kunci utama, AWS akan mengabaikan kolom objek besar (LOB) di tabel sumber.

Arsitektur

Tumpukan teknologi sumber

- Amazon EC2

Tumpukan teknologi target

- Amazon DocumentDB

Arsitektur target

Alat

- AWS DMS — AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan database dengan mudah dan aman.
- Amazon DocumentDB - Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan database yang cepat, andal, dan dikelola sepenuhnya.
- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud.
- Microsoft SQL Server — SQL Server adalah sistem manajemen basis data relasional.
- SQL Server Management Studio (SSMS) — SSMS adalah alat untuk mengelola SQL Server, termasuk mengakses, mengkonfigurasi, dan mengelola komponen SQL Server.

Epik

Buat dan konfigurasikan VPC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Masuk ke AWS Managemen t Console dan buka konsol	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon VPC. Buat virtual private cloud (VPC) dengan rentang blok IPv4 CIDR.	
Buat grup keamanan dan jaringan ACLs.	Di konsol VPC Amazon, buat grup keamanan dan daftar kontrol akses jaringan (jaringan ACLs) untuk VPC Anda, sesuai dengan kebutuhan Anda. Anda juga dapat menggunakan pengaturan default untuk konfigurasi ini. Untuk informasi selengkapnya tentang ini dan cerita lainnya, lihat bagian “Sumber daya terkait”.	Administrator sistem

Buat dan konfigurasikan cluster Amazon DocumentDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster Amazon DocumentDB.	Buka konsol Amazon DocumentDB dan pilih “Clusters.” Pilih “Buat,” dan buat cluster Amazon DocumentDB dengan satu instance. Penting: Pastikan Anda mengonfigurasi klaster ini dengan grup keamanan VPC Anda.	Administrator sistem
Instal cangkang mongo.	Mongo shell adalah utilitas baris perintah yang Anda gunakan untuk menghubun gkan dan menanyakan cluster	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon DocumentDB Anda. Untuk menginstalnya, jalankan perintah “/etc/yum.repos.d/mongodb-org-3.6.repo” untuk membuat file repositori. Jalankan perintah “sudo yum install -y mongodb-org-shell” untuk menginstal shell mongo. Untuk mengenkripsi data dalam perjalanan, unduh kunci publik untuk Amazon DocumentDB, lalu sambungkan ke instans Amazon DocumentDB Anda. Untuk informasi selengkapnya tentang langkah-langkah ini, lihat bagian “Sumber daya terkait”.	
Buat database di cluster Amazon DocumentDB.	Jalankan perintah “gunakan” dengan nama database Anda untuk membuat database di cluster Amazon DocumentDB Anda.	Administrator sistem

Membuat dan mengonfigurasi instans replikasi AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance replikasi AWS DMS.	Buka konsol AWS DMS dan pilih “Buat instance replikasi.” Masukkan nama dan deskripsi untuk tugas replikasi Anda. Pilih kelas instans, versi mesin, penyimpanan, VPC,	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Multi-AZ, dan membuatnya dapat diakses publik. Pilih tab “Advanced” untuk mengatur pengaturan jaringan dan enkripsi. Tentukan pengaturan pemeliharaan, lalu pilih “Buat contoh replikasi.”	
Konfigurasikan database SQL Server.	Masuk ke Microsoft SQL Server dan tambahkan aturan masuk untuk komunikasi antara titik akhir sumber dan instans replikasi AWS DMS. Gunakan alamat IP pribadi instance replikasi sebagai sumbernya. Penting: Instance replikasi dan titik akhir target harus berada di VPC yang sama. Gunakan sumber alternatif dalam grup keamanan jika VPCs berbeda untuk contoh sumber dan replikasi.	Administrator sistem

Membuat dan menguji sumber dan target titik akhir di AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir basis data sumber dan target.	Buka konsol AWS DMS dan pilih “Connect source and target database endpoint.” Tentukan informasi koneksi untuk basis data sumber dan target. Jika diperlukan, pilih tab “Advanced” untuk	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menetapkan nilai untuk “Atribut koneksi tambahan.” Unduh dan gunakan bundel sertifikat dalam konfigurasi titik akhir Anda.	
Uji koneksi titik akhir.	Pilih “Jalankan tes” untuk menguji koneksi. Memecahkan masalah pesan kesalahan apa pun dengan memverifikasi pengaturan grup keamanan dan koneksi ke instans replikasi AWS DMS dari instans database sumber dan target.	Administrator sistem

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas migrasi AWS DMS.	Pada konsol AWS DMS, pilih “Tugas,” “Buat tugas.” Tentukan opsi tugas, termasuk nama titik akhir sumber dan tujuan, dan nama instance replikasi. Di bawah “Jenis migrasi” pilih “Migrasi data yang ada,” dan “Replikasi perubahan data saja.” Pilih “Mulai tugas.”	Administrator sistem
Jalankan tugas migrasi AWS DMS.	Di bawah “Pengaturan tugas,” tentukan pengaturan untuk mode persiapan tabel, seperti “Jangan lakukan apa-apa,”	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	“Jatuhkan tabel pada target,” “Pemotongan,” dan “Sertakan kolom LOB dalam replikasi .” Tetapkan ukuran LOB maksimum yang akan diterima AWS DMS dan pilih “Aktifkan logging.” Biarkan “Pengaturan lanjutan” pada nilai defaultnya dan pilih “Buat tugas.”	
Pantau migrasi.	Di konsol AWS DMS, pilih “Tugas” dan pilih tugas migrasi Anda. Pilih “Pemantauan tugas” untuk memantau tugas Anda. Tugas berhenti ketika migrasi beban penuh selesai dan perubahan cache diterapkan.	Administrator sistem

Uji dan verifikasi migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke cluster Amazon DocumentDB dengan menggunakan shell mongo.	Buka konsol Amazon DocumentDB, pilih cluster Anda di bawah “Clusters .” Di tab “Konektivitas dan Keamanan”, pilih “Connect to this cluster with the mongo shell.”	Administrator sistem
Verifikasi hasil migrasi Anda.	Jalankan perintah “gunakan” dengan nama database Anda dan kemudian jalankan perintah “tampilkan koleksi”.	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jalankan perintah “db. .count ();” dengan nama database Anda. Jika hasilnya cocok dengan basis data sumber Anda, maka migrasi Anda berhasil.	

Sumber daya terkait

Buat dan konfigurasikan VPC

- Buat grup keamanan untuk VPC Anda
- Buat jaringan ACL

Buat dan konfigurasikan cluster Amazon DocumentDB

- Buat klaster Amazon DocumentDB
- Instal shell mongo untuk Amazon DocumentDB
- Connect ke klaster Amazon DocumentDB

Membuat dan mengonfigurasi instans replikasi AWS DMS

- Gunakan contoh replikasi publik dan pribadi

Membuat dan menguji sumber dan target titik akhir di AWS DMS

- Gunakan Amazon DocumentDB sebagai target AWS DMS
- Gunakan database SQL Server sebagai sumber AWS DMS
- Gunakan titik akhir AWS DMS

Migrasikan data

- Migrasi ke Amazon DocumentDB

Sumber daya lainnya

- Batasan dalam menggunakan SQL Server sebagai sumber AWS DMS
- Cara menggunakan Amazon DocumentDB untuk membangun dan mengelola aplikasi dalam skala

Memigrasi database ThoughtSpot Falcon lokal ke Amazon Redshift

Battulga Purevragchaa dan Antony Prasad Thevaraj, Amazon Web Services

Ringkasan

Gudang data lokal memerlukan waktu administrasi dan sumber daya yang signifikan, terutama untuk kumpulan data yang besar. Biaya finansial untuk membangun, memelihara, dan menumbuhkan gudang ini juga sangat tinggi. Untuk membantu mengelola biaya, menjaga kompleksitas ekstrak, transformasi, dan pemuatan (ETL) tetap rendah, dan memberikan kinerja seiring pertumbuhan data Anda, Anda harus terus-menerus memilih data mana yang akan dimuat dan data mana yang akan diarsipkan.

Dengan memigrasikan database ThoughtSpot Falcon lokal ke Amazon Web Services (AWS) Cloud, Anda dapat mengakses data lake dan gudang data berbasis cloud yang meningkatkan kelincahan bisnis, keamanan, dan keandalan aplikasi, selain mengurangi biaya infrastruktur secara keseluruhan. Amazon Redshift membantu menurunkan biaya dan overhead operasional gudang data secara signifikan. Anda juga dapat menggunakan Amazon Redshift Spectrum untuk menganalisis sejumlah besar data dalam format aslinya tanpa memuat data.

Pola ini menjelaskan langkah-langkah dan proses untuk memigrasikan database ThoughtSpot Falcon dari pusat data lokal ke database Amazon Redshift di AWS Cloud.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database ThoughtSpot Falcon yang dihosting di pusat data lokal

Versi produk

- ThoughtSpot versi 7.0.1

Arsitektur

Diagram menunjukkan alur kerja berikut:

1. Data dihosting di database relasional lokal.
2. AWS Schema Conversion Tool (AWS SCT) mengonversi bahasa definisi data (DDL) yang kompatibel dengan Amazon Redshift.
3. Setelah tabel dibuat, Anda dapat memigrasikan data menggunakan AWS Database Migration Service (AWS DMS).
4. Data dimuat ke Amazon Redshift.
5. Data disimpan di Amazon Simple Storage Service (Amazon S3) Simple Storage S3) jika Anda menggunakan Redshift Spectrum atau sudah meng-host data di Amazon S3.

Alat

- AWS DMS — AWS Data Migration Service (AWS DMS) membantu Anda dengan cepat dan aman memigrasikan database ke AWS.
- Amazon Redshift - Amazon Redshift adalah layanan gudang data berskala petabyte yang cepat, terkelola sepenuhnya, yang membuatnya sederhana dan hemat biaya untuk menganalisis semua data Anda secara efisien menggunakan alat intelijen bisnis yang ada.
- AWS SCT — AWS Schema Conversion Tool (AWS SCT) mengonversi skema database Anda yang ada dari satu mesin database ke mesin database lainnya.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi konfigurasi Amazon Redshift yang sesuai.	Identifikasi konfigurasi klaster Amazon Redshift yang sesuai berdasarkan kebutuhan dan volume data Anda. Untuk informasi selengkapnya, lihat klaster Amazon Redshift di dokumentasi Amazon Redshift.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Teliti Amazon Redshift untuk mengevaluasi apakah itu memenuhi kebutuhan Anda.	Gunakan Amazon Redshift FAQs untuk memahami dan mengevaluasi apakah Amazon Redshift memenuhi kebutuhan Anda.	DBA

Siapkan cluster Amazon Redshift target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster Amazon Redshift.	Masuk ke AWS Management Console, buka konsol Amazon Redshift, lalu buat klaster Amazon Redshift di cloud pribadi virtual (VPC). Untuk informasi selengkap nya, lihat Membuat klaster di VPC di dokumentasi Amazon Redshift.	DBA
Lakukan PoC untuk desain database Amazon Redshift Anda.	Ikuti praktik terbaik Amazon Redshift dengan melakukan proof of concept (PoC) untuk desain database Anda. Untuk informasi selengkapnya, lihat Melakukan bukti konsep untuk Amazon Redshift di dokumentasi Amazon Redshift.	DBA
Buat pengguna database.	Buat pengguna di database Amazon Redshift Anda dan berikan peran yang sesuai	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk akses ke skema dan tabel. Untuk informasi selengkap nya, lihat Memberikan hak akses untuk pengguna atau grup pengguna di dokumenta si Amazon Redshift.	
Terapkan pengaturan konfigurasi ke database target.	Terapkan pengaturan konfigurasi ke database Amazon Redshift sesuai dengan kebutuhan Anda. Untuk informasi selengkap nya tentang mengaktifkan parameter database, sesi, dan tingkat server, lihat referensi Konfigurasi di dokumentasi Amazon Redshift.	DBA

Buat objek di cluster Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel secara manual dengan DDL di Amazon Redshift.	(Opsional) Jika Anda menggunakan AWS SCT, tabel dibuat secara otomatis. Namun, jika ada kegagalan saat mereplikasi DDLs, Anda harus membuat tabel secara manual	DBA
Buat tabel eksternal untuk Redshift Spectrum.	Buat tabel eksternal dengan skema eksternal untuk Amazon Redshift Spectrum.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk membuat tabel eksternal, Anda harus menjadi pemilik skema eksternal atau superuser database. Untuk informasi selengkapnya, lihat Membuat tabel eksternal untuk Amazon Redshift Spectrum di dokumentasi Amazon Redshift.	

Migrasi data menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan AWS DMS untuk memigrasikan data.	Setelah Anda membuat DDL tabel di database Amazon Redshift, migrasi data Anda ke Amazon Redshift menggunak an AWS DMS. Untuk langkah dan petunjuk terperinci, lihat Menggunak an database Amazon Redshift sebagai target AWS DMS dalam dokumentasi AWS DMS.	DBA
Gunakan perintah COPY untuk memuat data.	Gunakan `COPY` perintah Amazon Redshift untuk memuat data dari Amazon S3 ke Amazon Redshift. Untuk informasi selengkapnya, lihat Menggunakan perintah COPY untuk memuat dari	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon S3 di dokumentasi Amazon Redshift.	

Validasi cluster Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi sumber dan catatan target.	Validasi jumlah tabel untuk sumber dan catatan target yang dimuat dari sistem sumber Anda.	DBA
Menerapkan praktik terbaik Amazon Redshift untuk penyetelan kinerja.	Menerapkan praktik terbaik Amazon Redshift untuk desain tabel dan database. Untuk informasi lebih lanjut, lihat posting blog 10 teknik penyetelan kinerja teratas untuk Amazon Redshift.	DBA
Optimalkan kinerja kueri.	Amazon Redshift menggunakan kueri berbasis SQL untuk berinteraksi dengan data dan objek dalam sistem. Bahasa manipulasi data (DHTML) adalah bagian dari SQL yang dapat Anda gunakan untuk melihat, menambah, mengubah, dan menghapus data. DDL adalah bagian dari SQL yang Anda gunakan untuk menambah, mengubah, dan menghapus	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	objek database seperti tabel dan tampilan. Untuk informasi selengkap nya, lihat Menyetel kinerja kueri di dokumentasi Amazon Redshift.	
Menerapkan WLM.	Anda dapat menggunakan manajemen beban kerja (WLM) untuk menentukan beberapa antrian kueri dan merutekan kueri ke antrian yang sesuai saat runtime. Untuk informasi selengkapnya, lihat Menerapkan manajemen beban kerja di dokumentasi Amazon Redshift.	DBA
Bekerja dengan penskalaan konkurensi.	Dengan menggunakan fitur Penskalaan Konkurensi, Anda dapat mendukung pengguna bersamaan dan kueri bersamaan yang hampir tidak terbatas, dengan kinerja kueri yang cepat secara konsisten. Untuk informasi selengkap nya, lihat Bekerja dengan penskalaan konkurensi di dokumentasi Amazon Redshift.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan praktik terbaik Amazon Redshift untuk desain tabel.	Ketika Anda merencanakan database Anda, keputusan desain tabel penting tertentu dapat sangat mempengaruhi kinerja kueri secara keseluruhan. Untuk informasi selengkapnya tentang memilih opsi desain tabel yang paling tepat, lihat praktik terbaik Amazon Redshift untuk mendesain tabel di dokumentasi Amazon Redshift.	DBA
Buat tampilan terwujud di Amazon Redshift.	Tampilan terwujud berisi kumpulan hasil yang dihitung sebelumnya berdasarkan kueri SQL di atas satu atau beberapa tabel dasar. Anda dapat mengeluarkan `SELECT` pernyataan untuk menanyakan tampilan terwujud dengan cara yang sama seperti Anda menanyakan tabel atau tampilan lain dalam database. Untuk informasi selengkapnya, lihat Membuat tampilan terwujud di Amazon Redshift di dokumentasi Amazon Redshift.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan gabungan antara tabel.	Untuk mencari lebih dari satu tabel pada saat yang sama ThoughtSpot, Anda harus menentukan gabungan antara tabel dengan menentukan kolom yang berisi data yang cocok di dua tabel. Kolom ini `foreign key` mewakili `primary key` dan bergabung . Anda dapat mendefinisikannya dengan menggunakan `ALTER TABLE` perintah di Amazon Redshift atau. ThoughtSpot Untuk informasi selengkap nya, lihat ALTER TABLE di dokumentasi Amazon Redshift.	DBA

Siapkan ThoughtSpot koneksi ke Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan koneksi Amazon Redshift.	Tambahkan sambungan Amazon Redshift ke database Falcon lokal ThoughtSpot Anda. Untuk informasi selengkap nya, lihat Menambahkan sambungan Amazon Redshift di dokumentasi. ThoughtSpot	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Edit koneksi Amazon Redshift.	Anda dapat mengedit koneksi Amazon Redshift untuk menambahkan tabel dan kolom. Untuk informasi selengkapnya, lihat Mengedit sambungan Amazon Redshift di dokumentasi. ThoughtSpot	DBA
Memetakan ulang koneksi Amazon Redshift.	Ubah parameter koneksi dengan mengedit file pemetaan sumber .yaml yang dibuat saat Anda menambahk an koneksi Amazon Redshift. Misalnya, Anda dapat memetakan ulang tabel atau kolom yang ada ke tabel atau kolom yang berbeda dalam koneksi database yang ada. ThoughtSpot merekomen dasikan agar Anda memeriksa dependensi sebelum dan sesudah Anda memetakan ulang tabel atau kolom dalam koneksi untuk memastikan bahwa mereka ditampilkan sesuai kebutuhan. Untuk informasi selengkap nya, lihat Memetakan ulang koneksi Amazon Redshift di ThoughtSpot dokumentasi.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus tabel dari koneksi Amazon Redshift.	(Opsional) Jika Anda mencoba menghapus tabel dalam koneksi Amazon Redshift, ThoughtSpot memeriksa dependensi dan menampilkan daftar objek dependen. Anda dapat memilih objek yang terdaftar untuk menghapusnya atau menghapus ketergant ungan. Anda kemudian dapat menghapus tabel. Untuk informasi selengkapnya, lihat Menghapus tabel dari sambungan Amazon Redshift di dokumentasi. ThoughtSpot	DBA
Hapus tabel dengan objek dependen dari koneksi Amazon Redshift.	(Opsional) Jika Anda mencoba menghapus tabel dengan objek dependen, operasi diblokir. Sebuah `Cannot delete` jendela muncul, dengan daftar tautan ke objek dependen. Ketika semua dependensi dihapus, Anda kemudian dapat menghapus tabel Untuk informasi selengkapnya, lihat Menghapus tabel dengan objek dependen dari koneksi Amazon Redshift dalam dokumentasi. ThoughtSpot	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus koneksi Amazon Redshift.	(Opsional) Karena koneksi dapat digunakan di beberapa sumber data atau visualisasi, Anda harus menghapus semua sumber dan tugas yang menggunakan koneksi tersebut sebelum Anda dapat menghapus koneksi Amazon Redshift. Untuk informasi selengkapnya, lihat Menghapus sambungan Amazon Redshift di dokumentasi. ThoughtSpot	DBA
Periksa referensi koneksi untuk Amazon Redshift.	Pastikan Anda memberikan informasi yang diperlukan untuk koneksi Amazon Redshift Anda dengan menggunakan referensi Koneksi dalam dokumentasi. ThoughtSpot	DBA

Informasi tambahan

- Analisis berbasis AI pada skala apa pun dengan ThoughtSpot dan Amazon Redshift
- Harga Amazon Redshift
- Memulai AWS SCT
- Memulai dengan Amazon Redshift
- Menggunakan agen ekstraksi data
- Chick-fil-A meningkatkan kecepatan wawasan dengan ThoughtSpot dan AWS

Migrasi dari Oracle Database ke Amazon RDS for PostgreSQL dengan menggunakan Oracle GoldenGate

Dhairya Jindani, Sindhusha Paturu, dan Rajeshkumar Sabankar, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memigrasikan database Oracle ke Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL dengan menggunakan Oracle Cloud Infrastructure (OCI). GoldenGate

Dengan menggunakan Oracle GoldenGate, Anda dapat mereplikasi data antara database sumber Anda dan satu atau lebih database tujuan dengan downtime minimal.

> **Note**
>
> Database Oracle sumber dapat berupa lokal atau di instans Amazon Elastic Compute Cloud (Amazon EC2). Anda dapat menggunakan prosedur serupa saat menggunakan alat replikasi lokal.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Lisensi Oracle GoldenGate
- Driver Java Database Connectivity (JDBC) untuk terhubung ke database PostgreSQL
- Skema dan tabel yang dibuat dengan AWS Schema Conversion Tool (AWS SCT) pada target Amazon RDS for PostgreSQL database

Batasan

- Oracle GoldenGate dapat mereplikasi data tabel yang ada (pemuatan awal) dan perubahan yang sedang berlangsung (mengubah pengambilan data) saja

Versi produk

- Oracle Database Enterprise Edition 10g atau versi yang lebih baru

- Oracle GoldenGate 12.2.0.1.1 untuk Oracle atau versi yang lebih baru
- Oracle GoldenGate 12.2.0.1.1 untuk PostgreSQL atau versi yang lebih baru

Arsitektur

Diagram berikut menunjukkan contoh alur kerja untuk memigrasikan database Oracle ke Amazon RDS for PostgreSQL dengan menggunakan Oracle: GoldenGate

Diagram menunjukkan alur kerja berikut:

1. Proses Oracle GoldenGate Extract berjalan melawan database sumber untuk mengekstrak data.
2. Proses Oracle GoldenGate Replicat mengirimkan data yang diekstraksi ke database Amazon RDS for PostgreSQL target.

Alat

- Oracle GoldenGate membantu Anda merancang, menjalankan, mengatur, dan memantau replikasi data dan mengalirkan solusi pemrosesan data di Oracle Cloud Infrastructure.
- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di AWS Cloud.

Epik

Unduh dan instal Oracle GoldenGate

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh Oracle GoldenGate.	Unduh versi Oracle GoldenGate berikut: • Oracle GoldenGate 12.2.0.1.1 untuk Oracle atau versi yang lebih baru • Oracle GoldenGate 12.2.0.1.1 untuk PostgreSQL atau versi yang lebih baru	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk mengunduh perangkat lunak, lihat GoldenGate Unduhan Oracle di situs web Oracle.	
Instal Oracle GoldenGate untuk Oracle di server Oracle Database sumber.	Untuk instruksi, lihat GoldenGate dokumentasi Oracle.	DBA
Instal Oracle GoldenGate untuk database PostgreSQL di instans Amazon. EC2	Untuk instruksi, lihat GoldenGate dokumentasi Oracle.	DBA

Konfigurasikan Oracle GoldenGate pada basis data sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Oracle GoldenGate untuk Oracle Database pada database sumber.	Untuk instruksi, lihat GoldenGate dokumentasi Oracle. Pastikan Anda mengonfigurasi yang berikut ini: • Penebangan tambahan • Pengguna Oracle GoldenGate • Setiap hibah dan izin yang diperlukan • File parameter • Proses manajer • Direktori • Berkas GLOBALS • Dompet Oracle	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Oracle GoldenGate untuk PostgreSQL pada database target.	Untuk petunjuk, lihat Bagian VI Menggunakan Oracle GoldenGate untuk PostgreSQL di situs web Oracle. Pastikan Anda mengonfigurasi yang berikut ini: • Proses manajer • Berkas GLOBALS • Dompet Oracle	DBA

Konfigurasikan pengambilan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan proses Ekstrak di database sumber.	Di sumber Oracle Database, buat file ekstrak untuk mengekstrak data. Untuk petunjuk, lihat ADD EXTRACT dalam dokumentasi Oracle. **Note** File ekstrak mencakup pembuatan file parameter ekstrak dan direktori file jejak.	DBA
Siapkan pompa data untuk mentransfer file jejak dari sumber ke database target.	Buat file parameter EXTRACT dan direktori file jejak dengan mengikuti instruksi di	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	PARFILE di Database Utilities di situs web Oracle. Untuk informasi lebih lanjut, lihat Apa itu Jejak? di Fusion Middleware Memahami Oracle GoldenGate di situs web Oracle.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan replikasi pada EC2 instance Amazon.	Buat file parameter replikasi dan direktori file jejak. Untuk informasi selengkap nya tentang membuat file parameter replikasi, lihat bagian 3.5 Memvalidasi file parameter dalam dokumentasi Oracle Database. Untuk informasi selengkapnya tentang membuat direktori file trail, lihat Membuat jejak di dokumentasi Oracle Cloud. ⚠ Important Pastikan Anda menambahkan entri tabel pos pemeriksaan di file GLOBALS pada target. Untuk informasi lebih lanjut, lihat Apa itu Replika? di Fusion Middleware Memahami Oracle GoldenGate di situs web Oracle.	DBA

Konfigurasikan replikasi data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dalam database sumber, buat file parameter untuk mengekstrak data untuk beban awal.	Ikuti petunjuk dalam Membuat file parameter di GGSCI dalam dokumentasi Oracle Cloud. **Important** Pastikan bahwa Manajer berjalan pada target.	DBA
Dalam database target, buat file parameter untuk mereplika si data untuk pemuatan awal.	Ikuti petunjuk dalam Membuat file parameter di GGSCI dalam dokumentasi Oracle Cloud. **Important** Pastikan Anda menambahkan dan memulai proses Replicat.	DBA

Potong ke database Amazon RDS for PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan proses Replicat dan pastikan database sumber dan target sinkron.	Bandingkan jumlah baris antara basis data sumber dan target untuk memastikan bahwa replikasi data berhasil.	DBA
Konfigurasikan dukungan bahasa definisi data (DDL).	Jalankan skrip DDL untuk membuat pemicu, urutan,	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sinonim, dan kunci referensial pada PostgreSQL. **Note** Anda dapat menggunakan aplikasi klien SQL standar apa pun untuk terhubung ke database di cluster DB Anda. Misalnya, Anda dapat menggunak an pgAdmin untuk terhubung ke instans DB Anda.	

Sumber daya terkait

- Amazon RDS untuk PostgreSQL (Panduan Pengguna Amazon RDS)
- EC2 Dokumentasi Amazon
- Oracle GoldenGate mendukung metode pemrosesan dan database (dokumentasi Oracle)

Migrasikan tabel partisi Oracle ke PostgreSQL menggunakan AWS DMS

Saurav Mishra dan Eduardo Valentim, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara mempercepat pemuatan tabel yang dipartisi dari Oracle ke PostgreSQL dengan menggunakan AWS Database Migration Service (AWS DMS), yang tidak mendukung partisi asli. Basis data PostgreSQL target dapat diinstal di Amazon Elastic Compute Cloud (Amazon), atau dapat berupa Amazon Relational Database Service (EC2Amazon RDS) untuk PostgreSQL atau instans DB Edisi Amazon Aurora PostgreSQL yang kompatibel.

Mengunggah tabel yang dipartisi mencakup langkah-langkah berikut:

1. Buat tabel induk yang mirip dengan tabel partisi Oracle, tetapi jangan sertakan partisi apa pun.
2. Buat tabel anak yang akan mewarisi dari tabel induk yang Anda buat di langkah 1.
3. Buat fungsi prosedur dan pemicu untuk menangani sisipan di tabel induk.

Namun, karena pemicu diaktifkan untuk setiap sisipan, beban awal menggunakan AWS DMS bisa sangat lambat.

Untuk mempercepat pemuatan awal dari Oracle ke PostgreSQL 9.0, pola ini membuat tugas AWS DMS terpisah untuk setiap partisi dan memuat tabel turunan yang sesuai. Anda kemudian membuat pemicu selama cutover.

PostgreSQL versi 10 mendukung partisi asli. Namun, Anda mungkin memutuskan untuk menggunakan partisi yang diwariskan dalam beberapa kasus. Untuk informasi selengkapnya, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Oracle sumber dengan tabel yang dipartisi
- Database PostgreSQL di AWS

Versi produk

- PostgreSQL 9.0

Arsitektur

Tumpukan teknologi sumber

- Tabel yang dipartisi di Oracle

Tumpukan teknologi target

- Tabel yang dipartisi di PostgreSQL (di Amazon, Amazon RDS untuk PostgreSQL, atau Aurora EC2 PostgreSQL)

Arsitektur target

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.

Epik

Mengatur AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel di PostgreSQL.	Buat tabel induk dan anak yang sesuai di PostgreSQL dengan kondisi pemeriksaan yang diperlukan untuk partisi.	DBA
Buat tugas AWS DMS untuk setiap partisi.	Sertakan kondisi filter partisi dalam tugas AWS DMS. Petakan partisi ke tabel anak PostgreSQL yang sesuai.	DBA
Jalankan tugas AWS DMS menggunakan full load dan change data capture (CDC).	Pastikan bahwa `StopTaskCachedChangesApplied` parameter diatur ke	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`true` dan `StopTaskCachedChangesNotApplied` parameter diatur ke`false`.	

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan tugas replikasi.	Sebelum Anda menghentikan tugas, konfirmasikan bahwa sumber dan tujuan sinkron.	DBA
Buat pemicu pada tabel induk.	Karena tabel induk akan menerima semua perintah insert dan update, buat pemicu yang akan merutekan perintah ini ke tabel turunan masing-masing berdasarkan kondisi partisi.	DBA

Sumber daya terkait

- AWS DMS
- Partisi Tabel (dokumentasi PostgreSQL)

Informasi tambahan

Meskipun PostgreSQL versi 10 mendukung partisi asli, Anda mungkin memutuskan untuk menggunakan partisi yang diwariskan untuk kasus penggunaan berikut:

- Partisi memberlakukan aturan bahwa semua partisi harus memiliki kumpulan kolom yang sama dengan induknya, tetapi pewarisan tabel mendukung anak-anak yang memiliki kolom tambahan.
- Warisan tabel mendukung beberapa pewarisan.

- Partisi deklaratif hanya mendukung partisi daftar dan rentang. Dengan pewarisan tabel, Anda dapat membagi data sesuai keinginan. Namun, jika pengecualian kendala tidak dapat memangkas partisi secara efektif, kinerja kueri akan terganggu.
- Beberapa operasi membutuhkan kunci yang lebih kuat saat menggunakan partisi deklaratif daripada saat menggunakan pewarisan tabel. Misalnya, menambahkan atau menghapus partisi ke atau dari tabel yang dipartisi memerlukan `ACCESS EXCLUSIVE` kunci pada tabel induk, sedangkan `SHARE UPDATE EXCLUSIVE` kunci cukup untuk pewarisan reguler.

Saat Anda menggunakan partisi pekerjaan terpisah, Anda juga dapat memuat ulang partisi jika ada masalah validasi AWS DMS. Untuk kontrol kinerja dan replikasi yang lebih baik, jalankan tugas pada instance replikasi terpisah.

Bermigrasi dari Amazon RDS for Oracle ke Amazon RDS untuk MySQL

Jitender Kumar, Srini Ramaswamy, dan Neha Sharma, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan Amazon Relational Database Service (Amazon RDS) untuk instans Oracle DB ke instans Amazon RDS for MySQL DB di Amazon Web Services (AWS). Pola ini menggunakan AWS Database Migration Service (AWS DMS) dan AWS Schema Conversion Tool (AWS SCT).

Pola ini memberikan praktik terbaik untuk menangani migrasi prosedur tersimpan. Ini juga mencakup dan perubahan kode untuk mendukung lapisan aplikasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Database sumber Amazon RDS for Oracle.
- Basis data target Amazon RDS for MySQL. Database sumber dan target harus berada dalam virtual private cloud (VPC) yang sama. Jika Anda menggunakan beberapa VPCs, atau Anda harus memiliki izin akses yang diperlukan.
- Grup keamanan yang memungkinkan konektivitas antara basis data sumber dan target, AWS SCT, server aplikasi, dan AWS DMS.
- Akun pengguna dengan hak istimewa yang diperlukan untuk menjalankan AWS SCT pada database sumber.
- Pencatatan tambahan diaktifkan untuk menjalankan AWS DMS pada database sumber.

Batasan

- Batas ukuran basis data Amazon RDS sumber dan target adalah 64 TB. Untuk informasi ukuran Amazon RDS, lihat dokumentasi AWS.
- Oracle tidak peka huruf besar/kecil untuk objek database, tetapi MySQL tidak. AWS SCT dapat menangani masalah ini saat membuat objek. Namun, beberapa pekerjaan manual diperlukan untuk mendukung ketidakpekaan kasus penuh.

- Migrasi ini tidak menggunakan ekstensi MySQL untuk mengaktifkan fungsi asli Oracle. AWS SCT menangani sebagian besar konversi, tetapi beberapa pekerjaan diperlukan untuk mengubah kode secara manual.
- Perubahan driver Java Database Connectivity (JDBC) diperlukan dalam aplikasi.

Versi produk

- Amazon RDS for Oracle 12.2.0.1 dan yang lebih baru. Untuk RDS yang saat ini didukung untuk versi Oracle, lihat dokumentasi AWS.
- Amazon RDS for MySQL 8.0.15 dan yang lebih baru. Untuk RDS yang saat ini didukung untuk versi MySQL, lihat dokumentasi AWS.
- AWS DMS versi 3.3.0 dan yang lebih baru. Lihat dokumentasi AWS untuk informasi selengkapnya tentang titik akhir sumber dan titik akhir target yang didukung AWS DMS.
- AWS SCT versi 1.0.628 dan yang lebih baru. Lihat sumber AWS SCT dan matriks dukungan titik akhir target dalam dokumentasi AWS.

Arsitektur

Tumpukan teknologi sumber

- Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Menggunakan database Oracle sebagai sumber AWS DMS.

Tumpukan teknologi target

- Amazon RDS for MySQL. Untuk informasi selengkapnya, lihat Menggunakan database yang kompatibel dengan MySQL sebagai target AWS DMS.

Arsitektur migrasi

Dalam diagram berikut, AWS SCT menyalin dan mengonversi objek skema dari database sumber Amazon RDS for Oracle dan mengirimkan objek ke database target Amazon RDS for MySQL. AWS DMS mereplikasi data dari database sumber dan mengirimkannya ke Amazon RDS for MySQL instance.

Alat

- AWS Data Migration Service membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud. Pola ini menggunakan Amazon RDS for Oracle dan Amazon RDS untuk MySQL.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi dan mesin basis data sumber dan target.		DBA
Identifikasi persyaratan perangkat keras untuk instance server target.		DBA, SysAdmin
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, SysAdmin
Pilih jenis instans yang tepat (kapasitas, fitur penyimpanan, fitur jaringan).		DBA, SysAdmin
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih strategi migrasi aplikasi.	Pertimbangkan apakah Anda menginginkan waktu henti penuh atau sebagian waktu henti untuk aktivitas cutover.	DBA, SysAdmin, Pemilik aplikasi

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC dan subnet.		SysAdmin
Buat grup keamanan dan daftar kontrol akses jaringan (ACLs).		SysAdmin
Konfigurasikan dan mulai Amazon RDS for Oracle instance.		DBA, SysAdmin
Konfigurasikan dan mulai Amazon RDS for MySQL instance.		DBA, SysAdmin
Siapkan kasus uji untuk validasi konversi kode.	Ini akan membantu dalam pengujian unit untuk kode yang dikonversi.	DBA, Pengembang
Konfigurasikan instans AWS DMS.		
Konfigurasikan titik akhir sumber dan target di AWS DMS.		

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skrip basis data target menggunakan AWS SCT.	Periksa keakuratan kode yang dikonversi oleh AWS SCT. Beberapa pekerjaan manual akan diperlukan.	DBA, Pengembang
Di AWS SCT, pilih pengaturan “Case Insensitive”.	Di AWS SCT, pilih Pengatura n Proyek, Sensitivitas Kasus Target, Case Insensitive.	DBA, Pengembang
Di AWS SCT, pilih untuk tidak menggunakan fungsi asli Oracle.	Di Pengaturan Proyek, periksa fungsi TO_ CHAR/TO_N UMBER/TO _DATE.	DBA, Pengembang
Buat perubahan untuk kode “sql%notfound”.	Anda mungkin harus mengonversi kode secara manual.	
Kueri pada tabel dan objek dalam prosedur tersimpan (gunakan kueri huruf kecil).		DBA, Pengembang
Buat skrip utama setelah semua perubahan dibuat, dan kemudian gunakan skrip utama pada database target.		DBA, Pengembang
Prosedur tersimpan unit-test dan panggilan aplikasi menggunakan data sampel.		
Bersihkan data yang dibuat selama pengujian unit.		DBA, Pengembang
Jatuhkan kendala kunci asing pada database target.	Langkah ini diperlukan untuk memuat data awal. Jika	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda tidak ingin menghapus batasan kunci asing, Anda harus membuat tugas migrasi untuk data khusus untuk tabel primer dan sekunder.	
Jatuhkan kunci utama dan kunci unik pada database target.	Langkah ini menghasilkan kinerja yang lebih baik untuk beban awal.	DBA, Pengembang
Aktifkan logging tambahan pada database sumber.		DBA
Buat tugas migrasi untuk pemuatan awal di AWS DMS, lalu jalankan.	Pilih opsi untuk memigrasi data yang ada.	DBA
Tambahkan kunci utama dan kunci asing ke database target.	Kendala perlu ditambahkan setelah beban awal.	DBA, Pengembang
Buat tugas migrasi untuk replikasi yang sedang berlangsung.	Replikasi yang sedang berlangsung membuat database target disinkronkan dengan database sumber.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ganti fungsi asli Oracle dengan fungsi asli MySQL.		Pemilik aplikasi
Pastikan bahwa hanya nama huruf kecil yang digunakan		DBA, SysAdmin, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
untuk objek database dalam query SQL.		

Potong ke database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan server aplikasi.		Pemilik aplikasi
Validasi bahwa basis data sumber dan target sinkron.		DBA, Pemilik aplikasi
Hentikan instans Amazon RDS for Oracle DB.		DBA
Hentikan tugas migrasi.	Ini akan berhenti secara otomatis setelah Anda menyelesaikan langkah sebelumnya.	DBA
Ubah koneksi JDBC dari Oracle ke MySQL.		Pemilik aplikasi, DBA
Mulai aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Meninjau dan memvalidasi dokumen proyek.		DBA, SysAdmin
Kumpulkan metrik tentang waktu untuk bermigrasi, persentase tugas manual		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
versus alat, penghematan biaya, dll.		
Hentikan dan hapus instans AWS DMS.		DBA
Hapus titik akhir sumber dan target.		DBA
Hapus tugas migrasi.		DBA
Ambil snapshot dari instans Amazon RDS for Oracle DB.		DBA
Hapus instans Amazon RDS for Oracle DB.		DBA
Matikan dan hapus sumber daya AWS sementara lainnya yang Anda gunakan.		DBA, SysAdmin
Tutup proyek dan berikan umpan balik apa pun.		DBA

Sumber daya terkait

- AWS DMS
- AWS SCT
- Harga Amazon RDS
- Memulai dengan AWS DMS
- Memulai dengan Amazon RDS

Bermigrasi dari IBM Db2 di Amazon EC2 ke Aurora PostgreSQL yang kompatibel dengan menggunakan AWS DMS dan AWS SCT

Sirsendu Halder dan Abhimanyu Chhabra, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan database IBM Db2 pada instans Amazon Elastic Compute Cloud (Amazon) ke instans DB Edisi EC2 Amazon Aurora PostgreSQL yang kompatibel. Pola ini menggunakan AWS Database Migration Service (AWS DMS) dan AWS Schema Conversion Tool (AWS SCT) untuk migrasi data dan konversi skema.

Pola ini menargetkan strategi migrasi online dengan sedikit atau tanpa downtime untuk database IBM Db2 multi-terabyte yang memiliki jumlah transaksi yang tinggi. Kami menyarankan Anda mengonversi kolom dalam kunci utama (PKs) dan kunci asing (FKs) dengan tipe data `NUMERIC` ke `INT` atau `BIGINT` di PostgreSQL untuk kinerja yang lebih baik.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber IBM Db2 pada sebuah instance EC2

Versi produk

- DB2/LINUXX8664 versi 11.1.4.4 dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database Db2 pada sebuah instance EC2

Tumpukan teknologi target

- Aurora PostgreSQL versi 10.18 atau instans DB yang kompatibel dengan Aurora

Arsitektur migrasi database

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan database ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal. Database sumber tetap beroperasi penuh selama migrasi, meminimalkan waktu henti ke aplikasi yang bergantung pada database. Anda dapat menggunakan AWS DMS untuk memigrasikan data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan. AWS DMS mendukung migrasi heterogen antara platform database yang berbeda, seperti IBM Db2 ke Aurora PostgreSQL yang kompatibel dengan versi 10.18 atau lebih tinggi. Untuk detailnya, lihat Sumber untuk Migrasi Data dan Target untuk Migrasi Data dalam dokumentasi AWS DMS.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema database sumber dan sebagian besar objek kode database, termasuk tampilan, prosedur tersimpan, dan fungsi, ke format yang kompatibel dengan database target. Objek apa pun yang tidak dikonversi secara otomatis ditandai dengan jelas sehingga dapat dikonversi secara manual untuk menyelesaikan migrasi. AWS SCT juga dapat memindai kode sumber aplikasi untuk pernyataan SQL yang disematkan dan mengonversinya.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance DB yang kompatibel dengan Aurora PostgreSQL.	Untuk membuat instans DB, ikuti petunjuk dalam dokumentasi AWS. Untuk jenis mesin, pilih Amazon Aurora. Untuk edisi, pilih Amazon Aurora PostgreSQL Compatible Edition. Aurora PostgreSQL yang kompatibel dengan versi 10.18 atau instans DB yang lebih baru harus berada di cloud pribadi virtual (VPC)	Amazon RDS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang sama dengan sumber database IBM Db2.	

Konversikan skema basis data Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan verifikasi AWS SCT.	1. Instal AWS SCT dengan mengikuti langkah-langkah dalam dokumentasi AWS SCT. 2. Verifikasi penginstalan dengan mengikuti prosedur dalam dokumentasi AWS SCT.	Administrator AWS, DBA, insinyur migrasi
Mulai AWS SCT dan buat proyek.	Untuk memulai alat AWS SCT dan membuat proyek baru untuk menjalankan laporan penilaian migrasi database, ikuti petunjuk dalam dokumentasi AWS SCT.	Insinyur migrasi
Tambahkan server database dan buat aturan pemetaan.	1. Tambahkan server basis data sumber dan target dengan mengikuti petunjuk dalam dokumentasi AWS SCT. 2. Buat aturan pemetaan untuk menentukan platform basis data target untuk basis data sumber Anda. Untuk petunjuk, lihat dokumentasi AWS SCT.	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat laporan penilaian migrasi database.	Buat laporan penilaian migrasi database dengan mengikuti langkah-langkah dalam dokumentasi AWS SCT.	Insinyur migrasi
Lihat laporan penilaian.	Gunakan tab Ringkasan laporan penilaian migrasi database untuk melihat laporan dan menganalisis data. Analisis ini akan membantu Anda menentukan kompleksitas migrasi. Untuk informasi selengkapnya, lihat dokumentasi AWS SCT.	Insinyur migrasi
Konversi skema.	Untuk mengonversi skema basis data sumber Anda: 1. Di konsol AWS SCT, pilih View, lalu Main view. 2. Pilih objek atau node induk dari skema sumber Anda, buka menu konteks (klik kanan), lalu pilih Konversi skema. Untuk informasi selengkapnya, lihat dokumentasi AWS SCT.	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan skema database yang dikonversi ke instans DB target.	1. Pilih elemen skema di panel kanan proyek Anda yang menampilkan skema yang direncanakan untuk instans DB target Anda. 2. Buka menu konteks (klik kanan) untuk elemen skema, lalu pilih Terapkan ke database. Untuk informasi selengkapnya, lihat dokumentasi AWS SCT.	Insinyur migrasi

Migrasi data Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan grup parameter VPC dan DB.	Siapkan grup parameter VPC dan DB, dan konfigura sikan aturan dan parameter masuk yang diperlukan untuk migrasi. Untuk petunjuk, lihat dokumentasi AWS DMS. Untuk grup keamanan VPC, pilih EC2 instance untuk Db2 dan instans DB yang kompatibel dengan Aurora PostgreSQL. Instance replikasi ini harus dalam VPC yang sama dengan instans DB sumber dan target.	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan instans DB sumber dan target.	Siapkan instans DB sumber dan target untuk migrasi. Dalam lingkungan produksi, database sumber sudah ada. Untuk database sumber, nama server harus menjadi Public Domain Name System (DNS) dari EC2 contoh di mana Db2 berjalan. Untuk nama pengguna, Anda dapat menggunakan `db2inst1` diikuti oleh port, yang akan menjadi 5000 untuk IBM Db2.	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat EC2 klien dan titik akhir Amazon.	1. Buat EC2 klien Amazon. Anda menggunakan klien ini untuk mengisi basis data sumber Anda dengan data untuk mereplikasi. Anda juga menggunakan klien ini untuk memverifikasi replikasi dengan menjalankan kueri pada basis data target. 2. Buat titik akhir untuk database sumber dan target instans DB untuk digunakan untuk langkah selanjutnya. Untuk petunjuk, lihat dokumentasi AWS DMS. Anda harus membuat endpoint terpisah untuk database sumber dan target. Untuk Aurora PostgreSQL yang kompatibel dengan versi 10.18 atau yang lebih baru, portnya adalah 5432, dan Anda bisa mendapatkan nama server dari titik akhir instans DB.	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh replikasi.	Buat instance replikasi dengan menggunakan konsol AWS DMS dan tentukan titik akhir sumber dan target. Instance replikasi melakukan migrasi data antara titik akhir. Untuk informasi selengkapnya, lihat dokumentasi AWS DMS.	Insinyur migrasi
Buat tugas AWS DMS untuk memigrasikan data.	Buat tugas untuk memuat tabel IBM Db2 sumber ke instans PostgreSQL DB target dengan mengikuti langkah-langkah dalam dokumentasi AWS DMS. • Untuk sumber dan target, gunakan nama titik akhir sumber dan tujuan. • Jenis migrasi dapat memuat penuh. • Untuk aturan skema, Anda dapat menggunakan `inst1` skema dari database Db2. • Untuk nama tabel, tentukan `%` untuk memigrasikan semua tabel. Ketika beban selesai, Anda akan melihat tabel Db2 dari `inst1` skema yang muncul di database Aurora PostgreSQL yang kompatibel.	Insinyur migrasi

Sumber daya terkait

Referensi

- Dokumentasi Amazon Aurora
- Dokumentasi pembungkus data asing PostgreSQL (FDW)
- PostgreSQL IMPOR dokumentasi SKEMA ASING
- Dokumentasi AWS DMS
- Dokumentasi AWS SCT

Tutorial dan video

- Memulai dengan AWS DMS (panduan)
- Pengantar Amazon EC2 - Elastic Cloud Server & Hosting dengan AWS (video)

Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Amazon RDS for PostGresQL dan AWS DMS SharePlex

Kumar Babu P G, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Oracle 8i atau 9i lokal ke Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL atau Amazon Aurora PostgreSQL. AWS Database Migration Service (AWS DMS) tidak mendukung Oracle 8i atau 9i sebagai sumber, jadi Quest SharePlex mereplikasi data dari database 8i atau 9i lokal ke database Oracle perantara (Oracle 10g atau 11g), yang kompatibel dengan AWS DMS.

Dari instans Oracle perantara, skema dan data dimigrasikan ke database PostgreSQL di AWS dengan menggunakan AWS Schema Conversion Tool (AWS SCT) dan AWS DMS. Metode ini membantu mencapai streaming data secara terus menerus dari database sumber Oracle ke instans PostgreSQL DB target dengan lag replikasi minimum. Dalam implementasi ini, downtime terbatas pada lamanya waktu yang diperlukan untuk membuat atau memvalidasi semua kunci asing, pemicu, dan urutan pada database PostgreSQL target.

Migrasi menggunakan instans Amazon Elastic Compute Cloud (Amazon EC2) dengan Oracle 10g atau 11g diinstal untuk meng-host perubahan dari database Oracle sumber. AWS DMS menggunakan instans Oracle perantara ini sebagai sumber untuk mengalirkan data ke Amazon RDS for PostgreSQL atau Aurora PostgreSQL. Replikasi data dapat dijeda dan dilanjutkan dari database Oracle lokal ke instance Oracle perantara. Ini juga dapat dijeda dan dilanjutkan dari instance Oracle perantara ke database PostgreSQL target sehingga Anda dapat memvalidasi data dengan menggunakan validasi data AWS DMS atau alat validasi data kustom.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sumber database Oracle 8i atau 9i di pusat data lokal
- AWS Direct Connect dikonfigurasi antara pusat data lokal dan AWS
- Driver Java Database Connectivity (JDBC) untuk konektor AWS SCT diinstal baik pada mesin lokal atau pada instance di EC2 mana AWS SCT diinstal
- Keakraban dengan menggunakan database Oracle sebagai sumber AWS DMS
- Keakraban dengan menggunakan database PostgreSQL sebagai target AWS DMS

- Keakraban dengan replikasi SharePlex data Quest

Batasan

- Batas ukuran database adalah 64 TB
- Database Oracle lokal harus Enterprise Edition

Versi produk

- Oracle 8i atau 9i untuk database sumber
- Oracle 10g atau 11g untuk database perantara
- PostgreSQL 9.6 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database Oracle 8i atau 9i
- Pencarian SharePlex

Tumpukan teknologi target

- Amazon RDS untuk PostgreSQL atau Aurora PostgreSQL

Arsitektur sumber dan target

Alat

- AWS DMS — AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan database dengan cepat dan aman. Database sumber tetap beroperasi penuh selama migrasi, meminimalkan waktu henti ke aplikasi yang bergantung pada database. AWS DMS dapat memigrasikan data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan.

- AWS SCT — AWS Schema Conversion Tool (AWS SCT) membuat migrasi database heterogen dapat diprediksi dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar objek kode basis data, termasuk tampilan, prosedur tersimpan, dan fungsi, ke format yang kompatibel dengan basis data target. Objek yang tidak dapat dikonversi secara otomatis ditandai dengan jelas sehingga dapat dikonversi secara manual untuk menyelesaikan migrasi. AWS SCT juga dapat memindai kode sumber aplikasi Anda untuk pernyataan SQL yang disematkan dan mengonversinya sebagai bagian dari proyek konversi skema database. Selama proses ini, AWS SCT melakukan pengoptimalan kode cloud-native dengan mengonversi fungsi Oracle dan SQL Server lama menjadi setara AWS mereka, untuk membantu Anda memodernisasi aplikasi saat memigrasikan database Anda. Saat konversi skema selesai, AWS SCT dapat membantu memigrasikan data dari berbagai gudang data ke Amazon Redshift dengan menggunakan agen migrasi data bawaan.
- Quest SharePlex — Quest SharePlex adalah alat replikasi Oracle-to-Oracle data untuk memindahkan data dengan downtime minimal dan tidak ada kehilangan data.

Epik

Buat EC2 instance dan instal Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan jaringan untuk Amazon EC2.	Buat virtual private cloud (VPC), subnet, gateway internet, tabel rute, dan grup keamanan.	AWS SysAdmin
Buat EC2 instance baru.	Pilih Amazon Machine Image (AMI) untuk EC2 instance. Pilih ukuran instans dan konfigurasikan detail instans: jumlah instance (1), VPC dan subnet dari langkah sebelumnya, tetapkan IP publik secara otomatis, dan opsi lainnya. Tambahkan penyimpanan, konfigura sikan grup keamanan, dan	AWS SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	luncurkan instance. Saat diminta, buat dan simpan key pair untuk langkah selanjutn ya.	
Instal Oracle pada EC2 instance.	Dapatkan lisensi dan binari Oracle yang diperlukan, dan instal Oracle 10g atau 11g pada instans. EC2	DBA

Siapkan SharePlex pada sebuah EC2 instance dan konfigurasikan replikasi data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur SharePlex.	Buat EC2 instance Amazon dan instal SharePlex binari yang kompatibel dengan Oracle 8i atau 9i.	AWS SysAdmin, DBA
Konfigurasikan replikasi data.	Ikuti praktik SharePlex terbaik untuk mengonfigurasi replikasi data dari database Oracle 8i/9i lokal ke instans Oracle 10g/11g.	DBA

Mengkonversi skema database Oracle ke PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS SCT.	Buat laporan baru, lalu sambungkan ke Oracle sebagai sumber dan PostgreSQL sebagai target. Dalam pengaturan proyek,	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	buka tab SQL Scripting dan ubah skrip SQL target ke Multiple Files.	
Mengkonversi skema database Oracle.	Di tab Action, pilih Generate Report, Convert Schema, dan kemudian Save as SQL.	DBA
Ubah skrip SQL yang dihasilkan oleh AWS SCT.		DBA

Membuat dan mengonfigurasi instans Amazon RDS DB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon RDS DB.	Di konsol Amazon RDS, buat instance PostgreSQL DB baru.	AWS SysAdmin, DBA
Konfigurasikan instans DB.	Tentukan versi mesin DB, kelas instans DB, penyebaran multi-AZ, jenis penyimpanan, dan penyimpanan yang dialokasikan. Masukkan pengidentifikasi instans DB, nama pengguna master, dan kata sandi utama.	AWS SysAdmin, DBA
Konfigurasikan jaringan dan keamanan.	Tentukan VPC, grup subnet, aksesibilitas publik, preferensi Availability Zone, dan grup keamanan.	AWS SysAdmin, DBA
Konfigurasikan opsi basis data.	Tentukan nama database, port, grup parameter, enkripsi, dan kunci master.	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan cadangan.	Tentukan periode retensi cadangan, jendela cadangan, waktu mulai, durasi, dan apakah akan menyalin tag ke snapshot.	AWS SysAdmin, DBA
Konfigurasikan opsi pemantauan.	Mengaktifkan atau menonaktifkan pemantauan dan wawasan kinerja yang disempurnakan.	AWS SysAdmin, DBA
Konfigurasikan opsi pemeliharaan.	Tentukan peningkatan versi minor otomatis, jendela pemeliharaan, dan hari mulai, waktu, dan durasi.	AWS SysAdmin, DBA
Jalankan skrip pra-migrasi dari AWS SCT.	Pada instance Amazon RDS, jalankan skrip ini: create_database.sql, create_sequence.sql, create_table.sql, create_view.sql, dan create_function.sql.	AWS SysAdmin, DBA

Migrasi data dengan menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance replikasi di AWS DMS.	Lengkapi bidang untuk nama, kelas instance, VPC (sama seperti untuk EC2 instance), Multi-AZ, dan aksesibilitas publik. Di bagian konfigurasi lanjutan, tentukan penyimpanan yang dialokasikan, grup subnet, Availability Zone, grup	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	keamanan VPC, dan kunci root AWS Key Management Service (AWS KMS).	
Buat titik akhir basis data sumber.	Tentukan nama titik akhir, jenis, mesin sumber (Oracle), nama server (nama DNS EC2 pribadi Amazon), port, mode SSL, nama pengguna, kata sandi, SID, VPC (tentukan VPC yang memiliki instance replikasi), dan contoh replikasi . Untuk menguji koneksi, pilih Run Test, lalu buat endpoint. Anda juga dapat mengonfig urasi pengaturan lanjutan berikut: maxFileSize dan numberDataType Skala.	AWS SysAdmin, DBA
Buat tugas replikasi AWS DMS.	Tentukan nama tugas, contoh replikasi, titik akhir sumber dan target, dan contoh replikasi. Untuk jenis migrasi, pilih “Migrasi data yang ada dan replikasi perubahan yang sedang berlangsung.” Kosongkan kotak centang “Mulai tugas saat membuat”.	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan tugas replikasi AWS DMS.	Untuk mode persiapan tabel target, pilih “Jangan lakukan apa-apa.” Hentikan tugas setelah beban penuh selesai untuk membuat kunci utama. Tentukan mode LOB terbatas atau penuh, dan aktifkan tabel kontrol. Secara opsional, Anda dapat mengonfigurasi pengaturan CommitRate lanjutan.	DBA
Konfigurasikan pemetaan tabel.	Di bagian pemetaan tabel, buat aturan Sertakan untuk semua tabel di semua skema yang disertakan dalam migrasi, lalu buat aturan Kecualikan. Tambahkan tiga aturan transformasi untuk mengonversi skema, tabel, dan nama kolom menjadi huruf kecil, dan tambahkan aturan lain yang diperlukan untuk migrasi spesifik ini.	DBA
Mulai tugas.	Mulai tugas replikasi. Pastikan beban penuh berjalan. Jalankan ALTER SYSTEM SWITCH LOGFILE pada database Oracle utama untuk memulai tugas.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip mid-migration dari AWS SCT.	Di Amazon RDS for PostgreSQL, jalankan skrip ini: create_index.sql dan create_constraint.sql.	DBA
Mulai ulang tugas untuk melanjutkan perubahan pengambilan data (CDC).	Di instans Amazon RDS for PostgreSQL DB, jalankan VACUUM, dan mulai ulang tugas AWS DMS untuk menerapkan perubahan CDC yang di-cache.	DBA

Potong ke database PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa log AWS DMS dan tabel metadata.	Validasi kesalahan apa pun dan perbaiki jika diperlukan.	DBA
Hentikan semua dependensi Oracle.	Matikan pendengar pada database Oracle dan jalankan ALTER SYSTEM SWITCH LOGFILE. Hentikan tugas AWS DMS jika tidak menunjukkan aktivitas.	DBA
Jalankan skrip pasca-migrasi dari AWS SCT.	Di Amazon RDS for PostgreSQL, jalankan skrip ini: create_foreign_key_constrai nt.sql dan create_triggers.sql.	DBA
Selesaikan langkah-langkah Amazon RDS for PostgreSQL tambahan.	Tingkatkan urutan untuk mencocokkan Oracle jika diperlukan, jalankan VACUUM	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan ANALYZE, dan ambil snapshot untuk kepatuhan.	
Buka koneksi ke Amazon RDS for PostgreSQL.	Hapus grup keamanan AWS DMS dari Amazon RDS for PostgreSQL, tambahkan grup keamanan produksi, dan arahkan aplikasi Anda ke database baru.	DBA
Bersihkan sumber daya AWS DMS.	Hapus titik akhir, tugas replikasi, instance replikasi, dan instance. EC2	SysAdmin, DBA

Sumber daya terkait

- Dokumentasi AWS DMS
- Dokumentasi AWS SCT
- Harga Amazon RDS for PostgreSQL
- Menggunakan database Oracle sebagai sumber AWS DMS
- Menggunakan database PostgreSQL sebagai target AWS DMS
- SharePlex Dokumentasi pencarian

Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk PostgreSQL menggunakan tampilan terwujud dan AWS DMS

Kumar Babu P G dan Pragnesh Patel, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Oracle 8i atau 9i warisan lokal ke Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL atau Amazon Aurora PostgreSQL Compatible Edition.

AWS Database Migration Service (AWS DMS) tidak mendukung Oracle 8i atau 9i sebagai sumber, jadi pola ini menggunakan instance database Oracle perantara yang kompatibel dengan AWS DMS, seperti Oracle 10g atau 11g. Ini juga menggunakan fitur tampilan terwujud untuk memigrasikan data dari sumber Oracle 8i/9i instance ke instance Oracle 10g/11g menengah.

AWS Schema Conversion Tool (AWS SCT) mengonversi skema database, dan AWS DMS memigrasikan data ke database PostgreSQL target.

Pola ini membantu pengguna yang ingin bermigrasi dari database Oracle lama dengan downtime database minimum. Dalam implementasi ini, waktu henti akan dibatasi pada lamanya waktu yang diperlukan untuk membuat atau memvalidasi semua kunci asing, pemicu, dan urutan pada database target.

Pola ini menggunakan instans Amazon Elastic Compute Cloud (Amazon EC2) dengan database Oracle 10g/11g yang diinstal untuk membantu AWS DMS mengalirkan data. Anda dapat menghentikan sementara replikasi streaming dari database Oracle lokal ke instans Oracle perantara untuk mengaktifkan AWS DMS mengejar validasi data atau menggunakan alat validasi data lain. Instans PostgreSQL DB dan database Oracle perantara akan memiliki data yang sama ketika AWS DMS telah selesai memigrasikan perubahan saat ini.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sumber database Oracle 8i atau 9i di pusat data lokal
- AWS Direct Connect dikonfigurasi antara pusat data lokal dan AWS
- Driver Java Database Connectivity (JDBC) untuk konektor AWS SCT diinstal baik pada mesin lokal atau pada instance di EC2 mana AWS SCT diinstal

- Keakraban dengan menggunakan database Oracle sebagai sumber AWS DMS
- Keakraban dengan menggunakan database PostgreSQL sebagai target AWS DMS

Batasan

- Batas ukuran database adalah 64 TB

Versi produk

- Oracle 8i atau 9i untuk database sumber
- Oracle 10g atau 11g untuk database perantara
- PostgreSQL 10.17 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database Oracle 8i atau 9i

Tumpukan teknologi target

- Amazon RDS untuk PostgreSQL atau Aurora PostgreSQL kompatibel

Arsitektur target

Alat

- AWS DMS membantu memigrasikan database dengan cepat dan aman. Database sumber tetap beroperasi penuh selama migrasi, meminimalkan waktu henti ke aplikasi yang bergantung pada database. AWS DMS dapat memigrasikan data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan.
- AWS SCT secara otomatis mengonversi skema basis data sumber dan sebagian besar objek kode database, termasuk tampilan, prosedur tersimpan, dan fungsi, ke format yang kompatibel dengan basis data target. Objek yang tidak dapat dikonversi secara otomatis ditandai dengan jelas sehingga dapat dikonversi secara manual untuk menyelesaikan migrasi. AWS SCT juga dapat memindai kode sumber aplikasi Anda untuk pernyataan SQL yang disematkan dan

mengonversinya sebagai bagian dari proyek konversi skema database. Selama proses ini, AWS SCT melakukan pengoptimalan kode cloud-native dengan mengonversi fungsi Oracle dan SQL Server lama menjadi setara AWS mereka, untuk membantu Anda memodernisasi aplikasi saat memigrasikan database Anda. Saat konversi skema selesai, AWS SCT dapat membantu memigrasikan data dari berbagai gudang data ke Amazon Redshift dengan menggunakan agen migrasi data bawaan.

Praktik terbaik

Untuk praktik terbaik untuk menyegarkan tampilan terwujud, lihat dokumentasi Oracle berikut:

- Menyegarkan tampilan terwujud
- Penyegaran cepat untuk tampilan terwujud

Epik

Instal Oracle pada sebuah EC2 instance dan buat tampilan terwujud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan jaringan untuk EC2 contoh.	Buat virtual private cloud (VPC), subnet, gateway internet, tabel rute, dan grup keamanan.	AWS SysAdmin
Buat EC2 instance.	Pilih Amazon Machine Image (AMI) untuk EC2 instance. Pilih ukuran instans dan konfigurasikan detail instans: jumlah instance (1), VPC dan subnet dari langkah sebelumnya, tetapkan IP publik secara otomatis, dan opsi lainnya. Tambahkan penyimpanan, konfigura sikan grup keamanan, dan luncurkan instance. Saat	AWS SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	diminta, buat dan simpan key pair untuk langkah selanjutn ya.	
Instal Oracle pada EC2 instance.	Dapatkan lisensi dan binari Oracle yang diperlukan, dan instal Oracle 10g atau 11g pada instans. EC2	DBA
Konfigurasikan jaringan Oracle.	Memodifikasi atau menambahkan entri `listener.ora` untuk terhubung ke sumber lokal Oracle 8i/9i database, dan kemudian membuat link database.	DBA
Buat tampilan yang terwujud.	Identifikasi objek database untuk direplikasi dalam database sumber Oracle 8i/9i, dan kemudian membuat tampilan terwujud untuk semua objek dengan menggunakan link database.	DBA
Terapkan skrip untuk menyegarkan tampilan terwujud pada interval yang diperlukan.	Kembangkan dan terapkan skrip untuk menyegarkan tampilan terwujud pada interval yang diperlukan pada instans Amazon EC2 Oracle 10g/11g. Gunakan opsi penyegaran tambahan untuk menyegarkan tampilan yang terwujud.	DBA

Mengkonversi skema database Oracle ke PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS SCT.	Buat laporan baru, lalu sambungkan ke Oracle sebagai sumber dan PostgreSQL sebagai target. Dalam pengaturan proyek, buka tab SQL Scripting. Ubah skrip SQL target ke Multiple Files. (AWS SCT tidak mendukung database Oracle 8i/9i, jadi Anda harus memulihkan dump khusus skema pada instans Oracle 10g/11g perantara dan menggunakannya sebagai sumber untuk AWS SCT.)	DBA
Mengkonversi skema database Oracle.	Pada tab Action, pilih Generate Report, Convert Schema, dan kemudian Save as SQL.	DBA
Memodifikasi skrip SQL.	Lakukan modifikasi berdasarkan praktik terbaik. Misalnya, beralih ke tipe data yang sesuai dan kembangkan ekuivalen PostgreSQL untuk fungsi khusus Oracle.	DBA, DevDBA

Membuat dan mengonfigurasi instans Amazon RDS DB untuk meng-host database yang dikonversi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon RDS DB.	Di konsol Amazon RDS, buat instance PostgreSQL DB baru.	AWS SysAdmin, DBA
Konfigurasikan instans DB.	Tentukan versi mesin DB, kelas instans DB, penyebaran multi-AZ, jenis penyimpanan, dan penyimpanan yang dialokasikan. Masukkan pengidentifikasi instans DB, nama pengguna master, dan kata sandi utama.	AWS SysAdmin, DBA
Konfigurasikan jaringan dan keamanan.	Tentukan VPC, grup subnet, aksesibilitas publik, preferensi Availability Zone, dan grup keamanan.	DBA, SysAdmin
Konfigurasikan opsi basis data.	Tentukan nama database, port, grup parameter, enkripsi, dan kunci master.	DBA, AWS SysAdmin
Konfigurasikan cadangan.	Tentukan periode retensi cadangan, jendela cadangan, waktu mulai, durasi, dan apakah akan menyalin tag ke snapshot.	AWS SysAdmin, DBA
Konfigurasikan opsi pemantauan.	Mengaktifkan atau menonaktifkan pemantauan dan wawasan kinerja yang disempurnakan.	AWS SysAdmin, DBA
Konfigurasikan opsi pemeliharaan.	Tentukan peningkatan versi minor otomatis, jendela	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pemeliharaan, dan hari mulai, waktu, dan durasi.	
Jalankan skrip pra-migrasi dari AWS SCT.	Pada instans Amazon RDS for PostgreSQL target, buat skema database dengan menggunakan skrip SQL dari AWS SCT dengan modifikasi lainnya. Ini mungkin termasuk menjalankan beberapa skrip dan termasuk pembuatan pengguna, pembuatan database, pembuatan skema, tabel, tampilan, fungsi, dan objek kode lainnya.	AWS SysAdmin, DBA

Migrasi data dengan menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance replikasi di AWS DMS.	Lengkapi bidang untuk nama, kelas instance, VPC (sama seperti untuk EC2 instance) , Multi-AZ, dan aksesibilitas publik. Di bagian konfigurasi lanjutan, tentukan penyimpan an yang dialokasikan, grup subnet, Availability Zone, grup keamanan VPC, dan kunci AWS Key Management Service (AWS KMS).	AWS SysAdmin, DBA
Buat titik akhir basis data sumber.	Tentukan nama titik akhir, jenis, mesin sumber (Oracle), nama server (nama DNS	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pribadi EC2 instance), port, mode SSL, nama pengguna, kata sandi, SID, VPC (tentukan VPC yang memiliki instance replikasi), dan contoh replikasi. Untuk menguji koneksi, pilih Run Test, lalu buat endpoint. Anda juga dapat mengonfigurasi pengaturan lanjutan berikut: maxFileSizedan numberDataTypeSkala.	
Hubungkan AWS DMS ke Amazon RDS untuk PostgreSQL.	Buat grup keamanan migrasi untuk koneksi di seluruh VPCs, jika database PostgreSQL Anda ada di VPC lain.	AWS SysAdmin, DBA
Buat titik akhir basis data target.	Tentukan nama titik akhir, jenis, mesin sumber (PostgreSQL), nama server (titik akhir Amazon RDS), port, mode SSL, nama pengguna, kata sandi, nama database, VPC (tentukan VPC yang memiliki instance replikasi), dan contoh replikasi. Untuk menguji koneksi, pilih Run Test, lalu buat endpoint. Anda juga dapat mengonfigurasi pengaturan lanjutan berikut: maxFileSizedan numberDataTypeSkala.	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas replikasi AWS DMS.	Tentukan nama tugas, contoh replikasi, titik akhir sumber dan target, dan contoh replikasi. Untuk jenis migrasi, pilih Migrasi data yang ada dan replikasi perubahan yang sedang berlangsung. Kosongkan kotak centang Start task on create.	AWS SysAdmin, DBA
Konfigurasikan pengaturan tugas replikasi AWS DMS.	Untuk mode persiapan tabel target, pilih Jangan lakukan apa-apa. Hentikan tugas setelah beban penuh selesai (untuk membuat kunci utama). Tentukan mode LOB terbatas atau penuh, dan aktifkan tabel kontrol. Secara opsional, Anda dapat mengonfigurasi pengaturan CommitRat elanjutan.	DBA
Konfigurasikan pemetaan tabel.	Di bagian Pemetaan tabel, buat aturan Sertakan untuk semua tabel di semua skema yang disertakan dalam migrasi, lalu buat aturan Kecualikan. Tambahkan tiga aturan transformasi untuk mengonversi skema, tabel, dan nama kolom menjadi huruf kecil, dan tambahkan aturan lain yang Anda perlukan untuk migrasi khusus ini.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai tugas.	Mulai tugas replikasi. Pastikan beban penuh berjalan. Jalankan `ALTER SYSTEM SWITCH LOGFILE` pada database Oracle utama untuk memulai tugas.	DBA
Jalankan skrip mid-migration dari AWS SCT.	Di Amazon RDS for PostgreSQL, jalankan `create_constraint.sql` skrip `create_index.sql` berikut: dan (jika skema lengkap awalnya tidak dibuat).	DBA
Lanjutkan tugas untuk melanjutkan perubahan pengambilan data (CDC).	Jalankan `VACUUM` di instans Amazon RDS for PostgreSQL DB, dan mulai ulang tugas AWS DMS untuk menerapkan perubahan CDC yang di-cache.	DBA

Potong ke database PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa log AWS DMS dan tabel validasi.	Periksa dan perbaiki kesalahan replikasi atau validasi apa pun.	DBA
Berhenti menggunakan database Oracle lokal dan dependensinya.	Hentikan semua dependensi Oracle, matikan pendengar di database Oracle, dan jalankan. `ALTER SYSTEM SWITCH LOGFILE` Hentikan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tugas AWS DMS jika tidak menunjukkan aktivitas.	
Jalankan skrip pasca-migrasi dari AWS SCT.	Di Amazon RDS for PostgreSQL, jalankan skrip ini:. `create_foreign_key_constraint.sql` and `create_triggers.sql` Pastikan urutannya mutakhir.	DBA
Lengkapi tambahan Amazon RDS untuk langkah-langkah PostgreSQL.	Tingkatkan urutan untuk mencocokkan Oracle jika diperlukan, jalankan `VACUUM` dan`ANALYZE`, dan ambil snapshot untuk kepatuhan.	DBA
Buka koneksi ke Amazon RDS untuk PostgreSQL.	Hapus grup keamanan AWS DMS dari Amazon RDS for PostgreSQL, tambahkan grup keamanan produksi, dan arahkan aplikasi Anda ke database baru.	DBA
Bersihkan objek AWS DMS.	Hapus titik akhir, tugas replikasi, instance replikasi, dan instance. EC2	SysAdmin, DBA

Sumber daya terkait

- Dokumentasi AWS DMS
- Dokumentasi AWS SCT
- Harga Amazon RDS for PostgreSQL
- Menggunakan database Oracle sebagai sumber AWS DMS
- Menggunakan database PostgreSQL sebagai target AWS DMS

Bermigrasi dari Oracle di Amazon EC2 ke Amazon RDS untuk MySQL menggunakan AWS DMS dan AWS SCT

Anil Kunapareddy, Amazon Web Services

Harshad Gohil, Tidak Ada

Ringkasan

Mengelola database Oracle di instans Amazon Elastic Compute Cloud (Amazon EC2) memerlukan sumber daya dan bisa mahal. Memindahkan database ini ke Amazon Relational Database Service (Amazon RDS) untuk instans MySQL DB akan memudahkan pekerjaan Anda dengan mengoptimalkan anggaran TI secara keseluruhan. Amazon RDS for MySQL juga menyediakan fitur seperti multi-AZ, skalabilitas, dan pencadangan otomatis.

Pola ini memandu Anda melalui migrasi database Oracle sumber di Amazon EC2 ke target Amazon RDS for MySQL DB instans. Ini menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data, dan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema database sumber dan objek ke format yang kompatibel dengan Amazon RDS for MySQL.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber dengan layanan instance dan listener berjalan, dalam mode ARCHIVELOG
- Target Amazon RDS for MySQL database, dengan penyimpanan yang cukup untuk migrasi data

Batasan

- AWS DMS tidak membuat skema pada database target; Anda harus melakukannya. Nama skema harus sudah ada untuk target. Tabel dari skema sumber diimpor ke pengguna/skema, yang digunakan AWS DMS untuk terhubung ke instance target. Anda harus membuat beberapa tugas replikasi jika Anda harus memigrasikan beberapa skema.

Versi produk

- Semua edisi database Oracle untuk versi 10.2 dan yang lebih baru, 11g dan hingga 12.2, dan 18c. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Oracle sebagai Sumber

untuk AWS DMS dan Menggunakan Database yang kompatibel dengan MySQL sebagai Target untuk AWS DMS. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif. Untuk informasi tentang versi database Oracle yang didukung oleh AWS SCT, lihat dokumentasi AWS SCT.

- AWS DMS mendukung MySQL versi 5.5, 5.6, dan 5.7.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle pada sebuah instance EC2

Tumpukan teknologi target

- Amazon RDS for MySQL DB instans

Arsitektur migrasi data

Arsitektur sumber dan target

Alat

- AWS DMS - AWS Database Migration Service (AWS DMS) adalah layanan web yang dapat Anda gunakan untuk memigrasikan data dari database Anda yang ada di tempat, pada instans Amazon RDS DB, atau dalam database pada EC2 instans, ke database pada layanan AWS seperti Amazon RDS for MySQL atau instans. EC2 Anda juga dapat memigrasikan database dari layanan AWS ke database lokal. Anda dapat memigrasikan data antara mesin database heterogen atau homogen.
- AWS SCT - AWS Schema Conversion Tool (AWS SCT) membuat migrasi database heterogen dapat diprediksi dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar objek kode database, termasuk tampilan, prosedur tersimpan, dan fungsi, ke format yang kompatibel dengan basis data target. Setelah mengonversi skema database dan objek kode menggunakan AWS SCT, Anda dapat menggunakan AWS DMS untuk memigrasikan data dari database sumber ke database target untuk menyelesaikan proyek migrasi Anda.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi versi dan mesin basis data sumber dan target.		DBA/Pengembang
Identifikasi contoh replikasi DMS.		DBA/Pengembang
Identifikasi persyaratan penyimpanan seperti jenis dan kapasitas penyimpanan.		DBA/Pengembang
Identifikasi persyaratan jaringan seperti latensi dan bandwidth.		DBA/Pengembang
Identifikasi persyaratan perangkat keras untuk instance server sumber dan target (berdasarkan daftar kompatibilitas Oracle dan persyaratan kapasitas).		DBA/Pengembang
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA/Pengembang
Instal driver AWS SCT dan Oracle.		DBA/Pengembang
Tentukan strategi cadangan.		DBA/Pengembang
Tentukan persyaratan ketersediaan.		DBA/Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi migrasi aplikasi dan strategi peralihan.		DBA/Pengembang
Pilih jenis instans DB yang tepat berdasarkan kapasitas , penyimpanan, dan fitur jaringan.		DBA/Pengembang

Konfigurasikan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC). Sumber, target, dan contoh replikasi harus dalam VPC yang sama. Ini juga bagus untuk memiliki ini di Availability Zone yang sama.		Developer
Buat grup keamanan yang diperlukan untuk akses database.		Developer
Hasilkan dan konfigurasikan key pair.		Developer
Konfigurasikan subnet, Availability Zone, dan blok CIDR.		Developer

Konfigurasikan sumbernya: Oracle database pada contoh EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Oracle Database di Amazon EC2 dengan pengguna dan peran yang diperlukan.		DBA
Lakukan tiga langkah di kolom berikutnya untuk mengakses Oracle dari luar EC2 instance.	1. Ubah host lokal `tnsnames` ke DNS EC2 publik Amazon. 2. Ubah host lokal `listener` ke DNS EC2 publik Amazon. 3. Berhenti dan restart pendengar.	DBA
Saat Amazon EC2 dimulai ulang, DNS publik berubah. Pastikan untuk memperbarui DNS EC2 publik Amazon di 'tnsnames' dan 'listener' atau gunakan alamat IP Elastis.		DBA/Pengembang
Konfigurasikan grup keamanan EC2 instance sehingga instance replikasi dan klien yang diperlukan dapat mengakses database sumber.		DBA/Pengembang

Konfigurasikan target: Amazon RDS for MySQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan mulai Amazon RDS for MySQL DB instans.		Developer
Buat tablespace yang diperlukan di Amazon RDS for MySQL DB instance.		DBA
Konfigurasikan grup keamanan sehingga instance replikasi dan klien yang diperlukan dapat mengakses database target.		Developer

Konfigurasikan AWS SCT dan buat skema di database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal driver AWS SCT dan Oracle.		Developer
Masukkan parameter yang sesuai dan sambungkan ke sumber dan target.		Developer
Hasilkan laporan konversi skema.		Developer
Perbaiki kode dan skema seperlunya, terutama tablespace dan tanda kutip, dan jalankan pada database target.		Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi skema pada sumber vs. target sebelum memigrasi kan data.		Developer

Migrasi data menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Untuk full-load dan change data capture (CDC) atau hanya CDC, Anda harus menetapkan atribut koneksi tambahan.		Developer
Pengguna yang ditentuka n dalam definisi database Oracle sumber AWS DMS harus diberikan semua hak istimewa yang diperlukan. Untuk daftar lengkapnya, lihat https://docs.aws.amazon.com /dms/ latest/userguide/C HAP _source.oracle.html #CHAP_Source .Oracle.Self-Managed.		DBA/Pengembang
Aktifkan logging tambahan di database sumber.		DBA/Pengembang
Untuk pengambilan data penuh dan perubahan (CDC) atau hanya CDC, aktifkan mode ARCHIVELOG di database sumber.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir sumber dan target, dan uji koneksi.		Developer
Ketika titik akhir berhasil terhubung, buat tugas replikasi .		Developer
Pilih CDC saja (atau) beban penuh plus CDC dalam tugas untuk menangkap perubahan hanya untuk replikasi berkelanjutan (atau) beban penuh ditambah perubahan yang sedang berlangsung, masing-masing.		Developer
Jalankan tugas replikasi dan pantau CloudWatch log Amazon.		Developer
Validasi data dalam basis data sumber dan target.		Developer

Migrasikan aplikasi Anda dan potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti langkah-langkah untuk strategi migrasi aplikasi Anda.		DBA, Pengembang, Pemilik aplikasi
Ikuti langkah-langkah untuk strategi cutover/switch-over aplikasi Anda.		DBA, Pengembang, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi skema dan data dalam basis data sumber vs. target.		DBA/Pengembang
Kumpulkan metrik di sekitar waktu untuk bermigras i, persen manual vs alat, penghematan biaya, dll.		DBA/Developer/AppOwner
Tinjau dokumen proyek dan artefak.		DBA/Developer/AppOwner
Matikan sumber daya AWS sementara.		DBA/Pengembang
Tutup proyek dan berikan umpan balik.		DBA/Developer/AppOwner

Sumber daya terkait

- Dokumentasi AWS DMS
- Situs AWS DMS
- Posting blog AWS DMS
- Strategi untuk Migrasi Database Oracle ke AWS
- Amazon RDS for Oracle FAQs
- FAQ Oracle
- Amazon EC2
- Amazon EC2 FAQs
- Melisensikan Perangkat Lunak Oracle di Lingkungan Cloud Computing

Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS untuk MariaDB menggunakan AWS DMS dan AWS SCT

Veeranjaneyulu Grandhi dan Vinod Kumar, Amazon Web Services

Ringkasan

Pola ini memandu Anda melalui langkah-langkah untuk memigrasikan database Oracle pada instans Amazon Elastic Compute Cloud (Amazon EC2) ke Amazon Relational Database Service (Amazon RDS) untuk instance MariaDB DB. Pola ini menggunakan AWS Data Migration Service (AWS DMS) untuk migrasi data dan AWS Schema Conversion Tool (AWS SCT) untuk konversi skema.

Mengelola database Oracle pada EC2 instans membutuhkan lebih banyak sumber daya dan lebih mahal daripada menggunakan database di Amazon RDS. Amazon RDS memudahkan untuk mengatur, mengoperasikan, dan menskalakan database relasional di cloud. Amazon RDS menyediakan kapasitas yang hemat biaya dan dapat diubah ukurannya sambil mengotomatiskan tugas administrasi yang memakan waktu seperti penyediaan perangkat keras, penyiapan basis data, penambalan, dan pencadangan.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Database Oracle sumber dengan layanan instance dan listener aktif dan berjalan. Database ini harus dalam mode ARCHIVELOG.
- Keakraban dengan Menggunakan Database Oracle sebagai Sumber AWS DMS.
- Keakraban dengan Menggunakan Oracle sebagai Sumber AWS SCT.

Batasan

- Batas ukuran database: 64 TB

Versi produk

- Semua edisi database Oracle untuk versi 10.2 dan yang lebih baru, 11g dan hingga 12.2, dan 18c. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS dan tabel versi AWS SCT dalam dokumentasi AWS.

- Amazon RDS mendukung MariaDB Server Community Server versi 10.3, 10.4, 10.5, dan 10.6. Untuk daftar terbaru versi yang didukung, lihat dokumentasi Amazon RDS.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle pada sebuah instance EC2

Tumpukan teknologi target

- Amazon RDS for MariaDB

Arsitektur migrasi data

Arsitektur target

Alat

- AWS Schema Conversion Tool (AWS SCT) membuat migrasi database heterogen dapat diprediksi dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar objek kode database—termasuk tampilan, prosedur tersimpan, dan fungsi—ke format yang kompatibel dengan basis data target. Setelah mengonversi skema database dan objek kode menggunakan AWS SCT, Anda dapat menggunakan AWS DMS untuk memigrasikan data dari database sumber ke database target untuk menyelesaikan proyek migrasi Anda. Untuk informasi selengkapnya, lihat Menggunakan Oracle sebagai Sumber AWS SCT dalam dokumentasi AWS SCT.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan database ke AWS dengan cepat dan aman. Database sumber tetap beroperasi penuh selama migrasi, meminimalkan waktu henti ke aplikasi yang bergantung pada database. AWS DMS dapat memigrasikan data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan. AWS DMS mendukung migrasi homogen seperti Oracle ke Oracle, serta migrasi heterogen antara platform database yang berbeda, seperti Oracle atau Microsoft SQL Server ke Amazon Aurora. Untuk mempelajari lebih lanjut tentang memigrasi database Oracle, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS dalam dokumentasi AWS DMS.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi versi dan mesin database.	Identifikasi versi dan mesin basis data sumber dan target.	DBA, Pengembang
Identifikasi contoh replikasi.	Identifikasi instans replikasi AWS DMS.	DBA, Pengembang
Identifikasi persyaratan penyimpanan.	Identifikasi jenis dan kapasitas penyimpanan.	DBA, Pengembang
Identifikasi persyaratan jaringan.	Identifikasi latensi jaringan dan bandwidth.	DBA, Pengembang
Identifikasi persyaratan perangkat keras.	Identifikasi persyaratan perangkat keras untuk instance server sumber dan target (berdasarkan daftar kompatibilitas Oracle dan persyaratan kapasitas).	DBA, Pengembang
Identifikasi persyaratan keamanan.	Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.	DBA, Pengembang
Instal driver.	Instal driver AWS SCT dan Oracle terbaru.	DBA, Pengembang
Tentukan strategi cadangan.		DBA, Pengembang
Tentukan persyaratan ketersediaan.		DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih migration/switchover strategi aplikasi.		DBA, Pengembang
Pilih jenis instans.	Pilih jenis instans yang tepat berdasarkan kapasitas , penyimpanan, dan fitur jaringan.	DBA, Pengembang

Konfigurasikan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).	Instance sumber, target, dan replikasi harus berada di VPC yang sama dan di Availability Zone yang sama (disarankan).	Developer
Buat grup keamanan.	Buat grup keamanan yang diperlukan untuk akses database.	Developer
Hasilkan key pair.	Hasilkan dan konfigurasikan key pair.	Developer
Konfigurasikan sumber daya lainnya.	Konfigurasikan subnet, Availability Zone, dan blok CIDR.	Developer

Konfigurasikan sumbernya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan EC2 instance.	Untuk petunjuk, lihat EC2 dokumentasi Amazon.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal database Oracle.	Instal database Oracle pada EC2 instance, dengan pengguna dan peran yang diperlukan.	DBA
Ikuti langkah-langkah dalam deskripsi tugas untuk mengakses Oracle dari luar EC2 instance.	1. Ubah host lokal `tnsnames` ke DNS EC2 publik Amazon. 2. Ubah host lokal `listener` ke DNS EC2 publik Amazon. 3. Berhenti dan restart pendengar.	DBA
Perbarui DNS EC2 publik Amazon.	Setelah EC2 instance dimulai ulang, DNS publik berubah. Pastikan untuk memperbar ui DNS EC2 publik Amazon di `tnsnames` dan`listener`, atau gunakan alamat IP Elastis.	DBA, Pengembang
Konfigurasikan grup keamanan EC2 instance.	Konfigurasikan grup keamanan EC2 instance sehingga instance replikasi dan klien yang diperlukan dapat mengakses database sumber.	DBA, Pengembang

Konfigurasikan target Amazon RDS untuk lingkungan MariaDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai instance RDS DB.	Konfigurasikan dan mulai Amazon RDS for MariaDB instans DB.	Developer
Buat ruang meja.	Buat ruang tabel yang diperlukan di database Amazon RDS MariaDB.	DBA
Konfigurasikan grup keamanan.	Konfigurasikan grup keamanan sehingga instance replikasi dan klien yang diperlukan dapat mengakses database target.	Developer

Konfigurasikan AWS SCT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal driver.	Instal driver AWS SCT dan Oracle terbaru.	Developer
Connect.	Masukkan parameter yang sesuai dan kemudian sambungkan ke sumber dan target.	Developer
Menghasilkan laporan konversi skema.	Buat laporan konversi skema AWS SCT.	Developer
Perbaiki kode dan skema seperlunya.	Lakukan koreksi yang diperlukan pada kode dan skema (terutama ruang tabel dan tanda kutip).	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi skema.	Validasi skema pada sumber versus target sebelum memuat data.	Developer

Migrasi data menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan atribut koneksi.	Untuk pengambilan data beban penuh dan perubahan (CDC) atau hanya untuk CDC, tetapkan atribut koneksi tambahan. Untuk informasi selengkapnya, lihat dokumentasi Amazon RDS.	Developer
Aktifkan pencatatan tambahan.	Aktifkan logging tambahan pada database sumber.	DBA, Pengembang
Aktifkan mode log arsip.	Untuk beban penuh dan CDC (atau hanya untuk CDC), aktifkan mode log arsip pada database sumber.	DBA
Buat dan uji titik akhir.	Buat titik akhir sumber dan target dan uji koneksi. Untuk informasi selengkapnya, lihat dokumentasi Amazon DMS.	Developer
Buat tugas replikasi.	Ketika titik akhir berhasil terhubung, buat tugas replikasi . Untuk informasi selengkap nya, lihat dokumentasi Amazon DMS.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih jenis replikasi.	Pilih CDC saja atau Full load plus CDC dalam tugas untuk menangkap perubahan hanya untuk replikasi berkelanjutan, atau untuk beban penuh dan perubahan yang sedang berlangsung, masing-masing.	Developer
Mulai dan pantau tugas.	Mulai tugas replikasi dan pantau CloudWatch log Amazon. Untuk informasi selengkapnya, lihat dokumentasi Amazon DMS.	Developer
Validasi data.	Validasi data dalam basis data sumber dan target.	Developer

Migrasikan aplikasi dan potong ke database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi yang dipilih.		DBA, Pemilik aplikasi, Pengembang
Ikuti cutover/switchover strategi aplikasi yang dipilih.		DBA, Pemilik aplikasi, Pengembang

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi skema dan data.	Pastikan skema dan data berhasil divalidasi di sumber	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	versus target sebelum penutupan proyek.	
Kumpulkan metrik.	Kumpulkan metrik untuk waktu migrasi, persentase tugas manual versus alat, penghematan biaya, dan kriteria serupa.	DBA, Pemilik aplikasi, Pengembang
Tinjau dokumentasi.	Tinjau dokumen proyek dan artefak.	DBA, Pemilik aplikasi, Pengembang
Matikan sumber daya.	Matikan sumber daya AWS sementara.	DBA, Pengembang
Tutup proyek.	Tutup proyek migrasi dan berikan umpan balik apa pun.	DBA, Pemilik aplikasi, Pengembang

Sumber daya terkait

- Ikhtisar MariaDB Amazon RDS
- Detail produk Amazon RDS for MariaDB
- Menggunakan Database Oracle sebagai Sumber AWS DMS
- Strategi untuk Migrasi Database Oracle ke AWS
- Melisensikan Perangkat Lunak Oracle di Lingkungan Cloud Computing
- Amazon RDS for Oracle FAQs
- Ikhtisar AWS DMS
- Posting blog AWS DMS
- EC2 Ikhtisar Amazon
- Amazon EC2 FAQs
- Dokumentasi AWS SCT

Memigrasikan database Oracle lokal ke Amazon RDS for MySQL menggunakan AWS DMS dan AWS SCT

Sergey Dmitriev dan Naresh Damera, Amazon Web Services

Ringkasan

Pola ini memandu Anda melalui migrasi database Oracle lokal ke Amazon Relational Database Service (Amazon RDS) untuk instans MySQL DB. Ini menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data, dan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema database sumber dan objek ke format yang kompatibel dengan Amazon RDS for MySQL.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Oracle sumber di pusat data lokal

Batasan

- Batas ukuran database: 64 TB

Versi produk

- Semua edisi database Oracle untuk versi 11g (versi 11.2.0.3.v1 dan yang lebih baru) dan hingga 12.2, dan 18c. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif. Untuk informasi tentang versi database Oracle yang didukung oleh AWS SCT, lihat dokumentasi AWS SCT.
- AWS DMS saat ini mendukung MySQL versi 5.5, 5.6, dan 5.7. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database yang kompatibel dengan MySQL sebagai Target untuk AWS DMS dalam dokumentasi AWS.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal

Tumpukan teknologi target

- Amazon RDS for MySQL DB instans

Arsitektur migrasi data

Alat

- AWS DMS - AWS Database Migration Services (AWS DMS) membantu Anda memigrasikan database relasional, gudang data, database NoSQL, dan jenis penyimpanan data lainnya. Anda dapat menggunakan AWS DMS untuk memigrasikan data Anda ke AWS Cloud, antar instans lokal (melalui penyiapan AWS Cloud), atau antara kombinasi pengaturan cloud dan lokal.
- AWS SCT - AWS Schema Conversion Tool (AWS SCT) digunakan untuk mengonversi skema database Anda dari satu mesin database ke mesin database lainnya. Kode khusus yang dikonversi alat mencakup tampilan, prosedur tersimpan, dan fungsi. Kode apa pun yang tidak dapat dikonversi oleh alat secara otomatis ditandai dengan jelas sehingga Anda dapat mengonversinya sendiri.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target dan mesin.		DBA
Identifikasi persyaratan perangkat keras untuk instance server target.		DBA, SysAdmin
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih jenis instans yang tepat berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.		DBA, SysAdmin
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA, SysAdmin
Identifikasi strategi migrasi aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat virtual private cloud (VPC) dan subnet.		SysAdmin
Buat grup keamanan dan daftar kontrol akses jaringan (ACLs).		SysAdmin
Konfigurasikan dan mulai instans Amazon RDS DB.		DBA, SysAdmin

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan skema database dengan menggunakan AWS SCT.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan menggunakan AWS DMS.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan AWS SCT untuk menganalisis dan mengonversi kode SQL di dalam kode aplikasi.	Untuk informasi selengkapnya, lihat https://docs.aws.amazon.com/SchemaConversionTool/ latest/userguide/ CHAP _Converting.app.html.	Pemilik aplikasi
Ikuti strategi migrasi aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.		DBA, SysAdmin, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, SysAdmin
Meninjau dan memvalidasi dokumen proyek.		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan metrik di sekitar waktu untuk bermigrasi,% manual vs alat, penghematan biaya, dll.		DBA, SysAdmin
Tutup proyek dan berikan umpan balik.		

Sumber daya terkait

Referensi

- Dokumentasi AWS DMS
- Dokumentasi AWS SCT
- Harga Amazon RDS

Tutorial dan video

- Memulai dengan AWS DMS
- Memulai dengan Amazon RDS
- AWS DMS (video)
- Amazon RDS (video)

Memigrasikan database Oracle lokal ke Amazon RDS for PostgreSQL menggunakan Oracle bystander dan AWS DMS

Cady Motyka, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Oracle lokal ke salah satu layanan database AWS yang kompatibel dengan PostgreSQL berikut dengan waktu henti minimal:

- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL
- Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL

Solusinya menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data, AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema database, dan database pengamat Oracle untuk membantu mengelola migrasi. Dalam implementasi ini, downtime dibatasi untuk berapa lama waktu yang dibutuhkan untuk membuat atau memvalidasi semua kunci asing pada database.

Solusinya juga menggunakan instans Amazon Elastic Compute Cloud (Amazon EC2) dengan database pengamat Oracle untuk membantu mengontrol aliran data melalui AWS DMS. Anda dapat menghentikan sementara replikasi streaming dari database Oracle lokal ke pengamat Oracle untuk mengaktifkan AWS DMS untuk mengejar validasi data, atau menggunakan alat validasi data lain. Instans Amazon RDS for PostgreSQL DB atau instans DB yang kompatibel dengan Aurora PostgreSQL dan database pengamat akan memiliki data yang sama saat AWS DMS menyelesaikan migrasi perubahan saat ini.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Oracle sumber di pusat data lokal dengan database siaga Active Data Guard yang dikonfigurasi
- AWS Direct Connect dikonfigurasi antara pusat data lokal dan AWS Secrets Manager untuk menyimpan rahasia database
- Driver Java Database Connectivity (JDBC) untuk konektor AWS SCT, diinstal baik pada mesin lokal atau pada instance di EC2 mana AWS SCT diinstal

- Keakraban dengan menggunakan database Oracle sebagai sumber AWS DMS
- Keakraban menggunakan database PostgreSQL sebagai target AWS DMS

Batasan

- Batas ukuran database: 64 TB

Versi produk

- AWS DMS mendukung semua edisi database Oracle untuk versi 10.2 dan yang lebih baru (untuk versi 10.x), 11g dan hingga 12.2, 18c, dan 19c. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif. Untuk informasi tentang versi database Oracle yang didukung oleh AWS SCT, lihat dokumentasi AWS SCT.
- AWS DMS mendukung PostgreSQL versi 9.4 dan yang lebih baru (untuk versi 9.x), 10.x, 11.x, 12.x, dan 13.x. Untuk informasi terbaru, lihat Menggunakan Database PostgreSQL sebagai Target untuk AWS DMS dalam dokumentasi AWS.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal
- Sebuah EC2 contoh yang menyimpan pengamat untuk database Oracle

Tumpukan teknologi target

- Amazon RDS untuk PostgreSQL atau Aurora PostgreSQL misalnya, PostgreSQL, PostgreSQL 9.3 dan yang lebih baru

Arsitektur target

Diagram berikut menunjukkan contoh alur kerja untuk memigrasikan database Oracle ke database AWS yang kompatibel dengan PostgreSQL dengan menggunakan AWS DMS dan pengamat Oracle:

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.

Epik

Mengkonversi skema database Oracle ke PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS SCT.	Buat laporan baru, dan sambungkan ke Oracle sebagai sumber dan PostgreSQL sebagai target. Di Pengaturan Proyek, buka tab SQL Scripting. Ubah Target SQL Script ke Multiple Files. File-file ini akan digunakan nanti dan diberi nama sebagai berikut: • create_database.sql • create_sequence.sql • create_table.sql • create_view.sql • create_function.sql	DBA
Mengkonversi skema database Oracle.	Di tab Tindakan, pilih Hasilkan Laporan. Kemudian, pilih	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Konversi Skema dan pilih Simpan sebagai SQL.	
Ubah skrip.	Misalnya, Anda mungkin ingin mengubah skrip jika nomor dalam skema sumber telah dikonversi ke format numerik di PostgreSQL, tetapi Anda ingin menggunakan BIGINT sebagai gantinya untuk kinerja yang lebih baik.	DBA

Membuat dan mengonfigurasi instans Amazon RDS DB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon RDS DB.	Di Wilayah AWS yang benar, buat instans PostgreSQL DB baru. Untuk informasi selengkapnya, lihat Membuat instans PostgreSQL DB dan menghubungkan ke database pada instans PostgreSQL DB dalam dokumentasi Amazon RDS.	AWS SysAdmin, DBA
Konfigurasikan spesifikasi instans DB.	Tentukan versi mesin DB, kelas instans DB, penyebaran multi-AZ, jenis penyimpanan, dan penyimpanan yang dialokasikan. Masukkan pengidentifikasi instans DB, nama pengguna utama, dan kata sandi utama.	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan jaringan dan keamanan.	Tentukan virtual private cloud (VPC), grup subnet, aksesibil itas publik, preferensi Availabil ity Zone, dan grup keamanan.	DBA, SysAdmin
Konfigurasikan opsi basis data.	Tentukan nama database, port, grup parameter, enkripsi, dan kunci KMS.	AWS SysAdmin, DBA
Konfigurasikan cadangan.	Tentukan periode retensi cadangan, jendela cadangan, waktu mulai, durasi, dan apakah akan menyalin tag ke snapshot.	AWS SysAdmin, DBA
Konfigurasikan opsi pemantauan.	Aktifkan atau nonaktifkan pemantauan dan wawasan kinerja yang disempurnakan.	AWS SysAdmin, DBA
Konfigurasikan opsi pemelihar aan.	Tentukan peningkatan versi minor otomatis, jendela pemeliharaan, dan hari mulai, waktu, dan durasi.	AWS SysAdmin, DBA
Jalankan skrip pra-migrasi dari AWS SCT.	Pada instans Amazon RDS, jalankan skrip berikut yang dihasilkan oleh AWS SCT: • create_database.sql • create_sequence.sql • create_table.sql • create_view.sql • create_function.sql	AWS SysAdmin, DBA

Konfigurasikan pengamat Oracle di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan jaringan untuk Amazon EC2.	Buat VPC baru, subnet, gateway internet, tabel rute, dan grup keamanan.	AWS SysAdmin
Buat EC2 instance.	Di Wilayah AWS yang sesuai, buat EC2 instance baru. Pilih Amazon Machine Image (AMI), pilih ukuran instans, dan konfigurasikan detail instans: jumlah instance (1), VPC dan subnet yang Anda buat di tugas sebelumnya, tetapkan IP publik secara otomatis, dan opsi lainnya. Tambahkan penyimpan an, konfigurasikan grup keamanan, dan luncurkan . Saat diminta, buat dan simpan key pair untuk langkah selanjutnya.	AWS SysAdmin
Connect database sumber Oracle ke EC2 instance.	Salin alamat IP IPv4 publik dan DNS ke file teks dan sambungkan dengan menggunakan SSH sebagai berikut: ssh -i “your_file.pem” EC2-user@<your-IP- -DNS>. address-or-public	AWS SysAdmin
Siapkan host awal untuk pengamat di Amazon. EC2	Siapkan kunci SSH, profil bash, ORATAB, dan tautan simbolis. Buat direktori Oracle.	AWS SysAdmin, Linux Admin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan salinan database untuk pengamat di Amazon EC2	Gunakan RMAN untuk membuat salinan database, mengaktifkan logging tambahan, dan membuat file kontrol siaga. Setelah penyalinan selesai, tempatkan database dalam mode pemulihan.	AWS SysAdmin, DBA
Siapkan Oracle Data Guard.	Ubah file listener.ora Anda dan mulai pendengar. Siapkan tujuan arsip baru. Tempatkan pengamat dalam mode pemulihan, ganti file sementara untuk menghinda ri korupsi di masa depan, instal crontab jika perlu untuk mencegah direktori arsip kehabisan ruang, dan edit manage-trclog-files-oraclef ile.cfg untuk sumber dan siaga.	AWS SysAdmin, DBA
Siapkan database Oracle untuk menyinkronkan pengiriman.	Tambahkan file log siaga dan ubah mode pemulihan. Ubah pengiriman log ke SYNC AFFIRM pada sumber utama dan sumber siaga. Alihkan log pada primer, konfirmas ikan melalui log peringata n EC2 pengamat Amazon bahwa Anda menggunakan file log siaga, dan konfirmas ikan bahwa aliran pengulang an mengalir di SYNC.	AWS SysAdmin, DBA

Migrasi data dengan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance replikasi di AWS DMS.	Lengkapi bidang untuk nama, kelas instance, VPC (sama seperti instans Amazon EC2), Multi-AZ, dan aksesibilitas publik. Di bagian Advance, tentukan penyimpanan yang dialokasikan, grup subnet, Availability Zone, grup keamanan VPC, dan kunci AWS Key Management Service (AWS KMS).	AWS SysAdmin, DBA
Buat titik akhir basis data sumber.	Tentukan nama titik akhir, jenis, mesin sumber (Oracle), nama server (nama DNS EC2 pribadi Amazon), port, mode SSL, nama pengguna, kata sandi, SID, VPC (tentukan VPC yang memiliki instance replikasi), dan contoh replikasi . Untuk menguji koneksi, pilih Run Test, lalu buat endpoint. Anda juga dapat mengonfig urasi pengaturan lanjutan berikut: maxFileSizedan numberDataTypeSkala.	AWS SysAdmin, DBA
Hubungkan AWS DMS ke Amazon RDS untuk PostgreSQL.	Buat grup keamanan migrasi untuk koneksi lintas VPCs.	AWS SysAdmin, DBA
Buat titik akhir basis data target.	Tentukan nama titik akhir, jenis, mesin sumber (PostgreSQL), nama server (titik akhir	AWS SysAdmin, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon RDS), port, mode SSL, nama pengguna, kata sandi, nama database, VPC (tentukan VPC yang memiliki instance replikasi), dan contoh replikasi. Untuk menguji koneksi, pilih Run Test, lalu buat endpoint. Anda juga dapat mengonfigurasi pengaturan lanjutan berikut: maxFileSize dan numberDat aTypeSkala.	
Buat tugas replikasi AWS DMS.	Tentukan nama tugas, contoh replikasi, titik akhir sumber dan target, dan contoh replikasi. Untuk jenis migrasi, pilih Migrasi data yang ada dan replikasi perubahan yang sedang berlangsung. Kosongkan kotak centang Start task on create.	AWS SysAdmin, DBA
Konfigurasikan pengaturan tugas replikasi AWS DMS.	Untuk mode persiapan tabel target, pilih Jangan lakukan apa-apa. Hentikan tugas setelah beban penuh selesai (untuk membuat kunci utama). Tentukan mode LOB terbatas atau penuh, dan aktifkan tabel kontrol. Secara opsional, Anda dapat mengonfigurasi pengaturan CommitRat elanjutan.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pemetaan tabel.	Di bagian Pemetaan tabel, buat aturan Sertakan untuk semua tabel di semua skema yang disertakan dalam migrasi, lalu buat aturan Kecualikan. Tambahkan tiga aturan transformasi untuk mengonversi skema, tabel, dan nama kolom menjadi huruf kecil, dan tambahkan aturan lain yang diperlukan untuk migrasi spesifik ini.	DBA
Mulai tugas.	Mulai tugas replikasi. Pastikan beban penuh berjalan. Jalankan ALTER SYSTEM SWITCH LOGFILE pada database Oracle utama untuk memulai tugas.	DBA
Jalankan skrip mid-migration dari AWS SCT.	Di Amazon RDS for PostgreSQL, jalankan skrip berikut yang dihasilkan oleh AWS SCT: • create_index.sql • create_constraint.sql	DBA
Mulai ulang tugas untuk melanjutkan perubahan pengambilan data (CDC).	Jalankan VACUUM di Amazon RDS untuk instans DB Amazon RDS for PostgreSQL, dan mulai ulang tugas AWS DMS untuk menerapkan perubahan CDC yang di-cache.	DBA

Potong ke database PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau log AWS DMS dan tabel validasi untuk mengetahui kesalahan apa pun.	Periksa dan perbaiki kesalahan replikasi atau validasi apa pun.	DBA
Hentikan semua dependensi Oracle.	Hentikan semua dependens i Oracle, matikan pendengar pada database Oracle, dan jalankan ALTER SYSTEM SWITCH LOGFILE. Hentikan tugas AWS DMS jika tidak menunjukkan aktivitas.	DBA
Jalankan skrip pasca-migrasi dari AWS SCT.	Di Amazon RDS for PostgreSQL, jalankan skrip berikut yang dihasilkan oleh AWS SCT: • create_foreign_key_constrai nt.sql • create_triggers.sql	DBA
Lengkapi tambahan Amazon RDS untuk langkah-langkah PostgreSQL.	Tingkatkan urutan untuk mencocokkan Oracle jika diperlukan, jalankan VACUUM dan ANALYZE, dan ambil snapshot untuk kepatuhan.	DBA
Buka koneksi ke Amazon RDS for PostgreSQL.	Hapus grup keamanan AWS DMS dari Amazon RDS for PostgreSQL, tambahkan grup keamanan produksi, dan arahkan aplikasi Anda ke database baru.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan objek AWS DMS.	Hapus titik akhir, tugas replikasi, instance replikasi, dan instance. EC2	SysAdmin, DBA

Sumber daya terkait

- Dokumentasi AWS DMS
- Dokumentasi AWS SCT
- Harga Amazon RDS for PostgreSQL

Migrasikan Database Oracle ke Amazon Redshift menggunakan AWS DMS dan AWS SCT

Piyush Goyal dan Brian motzer, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan database Oracle ke gudang data cloud Amazon Redshift di Amazon Web Services (AWS) Cloud dengan menggunakan AWS Database Migration Service (AWS DMS) dan AWS Schema Conversion Tool (AWS SCT). Pola ini mencakup basis data Oracle sumber yang ada di tempat atau diinstal pada instans Amazon Elastic Compute Cloud (Amazon EC2). Ini juga mencakup Amazon Relational Database Service (Amazon RDS) untuk database Oracle.

Prasyarat dan batasan

Prasyarat

- Database Oracle yang berjalan di pusat data lokal atau di AWS Cloud
- Akun AWS yang aktif
- Keakraban dengan menggunakan database Oracle sebagai sumber AWS DMS
- Keakraban dengan menggunakan database Amazon Redshift sebagai target AWS DMS
- Pengetahuan tentang Amazon RDS, Amazon Redshift, teknologi database yang berlaku, dan SQL
- Driver Java Database Connectivity (JDBC) untuk konektor AWS SCT, tempat AWS SCT diinstal

Versi produk

- Untuk database Oracle yang dikelola sendiri, AWS DMS mendukung semua edisi database Oracle untuk versi 10.2 dan yang lebih baru (untuk versi 10. x), 11g dan hingga 12,2, 18c, dan 19c. Untuk database Amazon RDS for Oracle yang dikelola AWS, AWS DMS mendukung semua edisi database Oracle untuk versi 11g (versi 11.2.0.4 dan yang lebih baru) dan hingga 12.2, 18c, dan 19c. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif.

Arsitektur

Tumpukan teknologi sumber

Salah satu dari yang berikut:

- Database Oracle lokal
- Database Oracle pada sebuah instance EC2
- Instans Amazon RDS for Oracle DB

Tumpukan teknologi target

- Amazon Redshift

Arsitektur target

Dari database Oracle yang berjalan di AWS Cloud hingga Amazon Redshift:

Dari database Oracle yang berjalan di pusat data lokal hingga Amazon Redshift:

Alat

- AWS DMS - AWS Data Migration Service (AWS DMS) membantu Anda memigrasikan database ke AWS dengan cepat dan aman. Database sumber tetap beroperasi penuh selama migrasi, meminimalkan waktu henti ke aplikasi yang bergantung pada database. AWS DMS dapat memigrasikan data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan.
- AWS SCT - AWS Schema Conversion Tool (AWS SCT) dapat digunakan untuk mengonversi skema database yang ada dari satu mesin database ke mesin database lainnya. Ini mendukung berbagai mesin database, termasuk Oracle, SQL Server, dan PostgressQL, sebagai sumber.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi database.	Validasi versi basis data sumber dan target dan pastikan mereka didukung	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	oleh AWS DMS. Untuk informasi tentang versi Oracle Database yang didukung, lihat Menggunakan database Oracle sebagai sumber AWS DMS. Untuk informasi tentang menggunakan Amazon Redshift sebagai target, lihat Menggunakan database Amazon Redshift sebagai target AWS DMS.	
Buat VPC dan grup keamanan.	Di akun AWS Anda, buat virtual private cloud (VPC), jika tidak ada. Buat grup keamanan untuk lalu lintas keluar ke basis data sumber dan target. Untuk informasi selengkapnya, lihat dokumentasi Amazon Virtual Private Cloud (Amazon VPC).	Administrator sistem
Instal AWS SCT.	Unduh dan instal AWS SCT versi terbaru dan driver yang sesuai. Untuk informasi selengkapnya, lihat Menginstal, memverifikasi, dan memperbarui AWS SCT.	DBA
Buat pengguna untuk tugas AWS DMS.	Buat pengguna AWS DMS di database sumber dan berikan hak READ. Pengguna ini akan digunakan oleh AWS SCT dan AWS DMS.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji konektivitas DB.	Uji konektivitas ke instans Oracle DB.	DBA
Buat proyek baru di AWS SCT.	Buka alat AWS SCT dan buat proyek baru.	DBA
Analisis skema Oracle yang akan dimigrasikan.	Gunakan AWS SCT untuk menganalisis skema yang akan dimigrasikan, dan buat laporan penilaian migrasi database. Untuk informasi selengkapnya, lihat Membuat laporan penilaian migrasi database di dokumentasi AWS SCT.	DBA
Tinjau laporan penilaian.	Tinjau laporan untuk kelayakan migrasi. Beberapa objek DB mungkin memerlukan konversi manual. Untuk informasi selengkapnya tentang laporan, lihat Melihat laporan penilaian di dokumentasi AWS SCT.	DBA

Siapkan database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster Amazon Redshift.	Buat klaster Amazon Redshift di dalam VPC yang Anda buat sebelumnya. Untuk informasi selengkapnya, lihat klaster Amazon Redshift	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di dokumentasi Amazon Redshift.	
Buat pengguna basis data.	Ekstrak daftar pengguna, peran, dan hibah dari database sumber Oracle. Buat pengguna di database Amazon Redshift target dan terapkan peran dari langkah sebelumnya.	DBA
Mengevaluasi parameter database.	Tinjau opsi database, parameter, file jaringan, dan tautan basis data dari database sumber Oracle, dan evaluasi penerapannya ke target.	DBA
Terapkan pengaturan yang relevan ke target.	Untuk informasi selengkap nya tentang langkah ini, lihat Referensi konfigura si di dokumentasi Amazon Redshift.	DBA

Buat objek dalam database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna AWS DMS di database target.	Buat pengguna AWS DMS di database target dan berikan hak baca dan tulis. Validasi konektivitas dari AWS SCT.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konversi skema, tinjau laporan SQL, dan simpan kesalahan atau peringatan apa pun.	Untuk informasi selengkap nya, lihat Mengonversi skema database menggunakan AWS SCT dalam dokumentasi AWS SCT.	DBA
Terapkan perubahan skema ke database target atau simpan sebagai file.sql.	Untuk petunjuk, lihat Menyimpan dan menerapkan skema konversi Anda di AWS SCT dalam dokumentasi AWS SCT.	DBA
Validasi objek dalam database target.	Validasi objek yang dibuat pada langkah sebelumny a dalam database target. Menulis ulang atau mendesain ulang objek apa pun yang tidak berhasil dikonversi.	DBA
Nonaktifkan kunci dan pemicu asing.	Nonaktifkan kunci dan pemicu asing apa pun. Ini dapat menyebabkan masalah pemuatan data selama proses pemuatan penuh saat menjalankan AWS DMS.	DBA

Migrasi data menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans replikasi AWS DMS.	Masuk ke AWS Managemen t Console, dan buka konsol AWS DMS. Di panel navigasi, pilih Instance replikasi, Buat instance replikasi. Untuk	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	petunjuk terperinci, lihat langkah 1 dalam Memulai AWS DMS dalam dokumentasi AWS DMS.	
Buat titik akhir sumber dan target.	Buat titik akhir sumber dan target, Uji koneksi dari instance replikasi ke titik akhir sumber dan target. Untuk petunjuk terperinci, lihat langkah 2 dalam Memulai AWS DMS dalam dokumentasi AWS DMS.	DBA
Buat tugas replikasi.	Buat tugas replikasi dan pilih metode migrasi yang sesuai. Untuk petunjuk terperinci, lihat langkah 3 dalam Memulai AWS DMS dalam dokumentasi AWS DMS.	DBA
Mulai replikasi data.	Mulai tugas replikasi dan pantau log untuk kesalahan apa pun.	DBA

Migrasi aplikasi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server aplikasi.	Buat server aplikasi baru di AWS.	Pemilik aplikasi
Migrasikan kode aplikasi.	Migrasikan kode aplikasi ke server baru.	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan server aplikasi.	Konfigurasikan server aplikasi untuk database target dan driver.	Pemilik aplikasi
Optimalkan kode aplikasi.	Optimalkan kode aplikasi untuk mesin target.	Pemilik aplikasi

Potong ke database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi pengguna.	Dalam database Amazon Redshift target, validasi pengguna dan berikan mereka peran dan hak istimewa.	DBA
Validasi bahwa aplikasi terkunci.	Pastikan aplikasi terkunci, untuk mencegah perubahan lebih lanjut.	Pemilik aplikasi
Validasi data.	Validasi data dalam database Amazon Redshift target.	DBA
Aktifkan kunci dan pemicu asing.	Aktifkan kunci asing dan pemicu di database Amazon Redshift target.	DBA
Connect ke database baru.	Konfigurasikan aplikasi untuk terhubung ke database Amazon Redshift baru.	Pemilik aplikasi
Lakukan pemeriksaan akhir.	Lakukan pemeriksaan sistem yang final dan komprehensif sebelum ditayangkan.	DBA, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pergi hidup.	Tayangkan langsung dengan basis data Amazon Redshift target.	DBA

Tutup proyek migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.	Matikan sumber daya AWS sementara seperti instans replikasi AWS DMS dan instans yang EC2 digunakan untuk AWS SCT.	DBA, Administrator sistem
Tinjau dokumen.	Meninjau dan memvalidasi dokumen proyek migrasi.	DBA, Administrator sistem
Kumpulkan metrik.	Kumpulkan informasi tentang proyek migrasi, seperti waktu untuk bermigrasi, persentase tugas manual versus alat, dan penghematan biaya total.	DBA, Administrator sistem
Tutup proyek.	Tutup proyek dan berikan umpan balik.	DBA, Administrator sistem

Sumber daya terkait

Referensi

- Panduan pengguna AWS DMS
- Panduan pengguna AWS SCT
- Panduan memulai Amazon Redshift

Tutorial dan video

- Selami AWS SCT dan AWS DMS (presentasi dari AWS re: Invent 2019)
- Memulai AWS Database Migration Service

Migrasikan database Oracle ke Aurora PostgreSQL menggunakan AWS DMS dan AWS SCT

Senthil Ramasamy, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Oracle ke Amazon Aurora PostgreSQL Compatible Edition dengan menggunakan AWS Data Migration Service (AWS DMS) dan AWS Schema Conversion Tool (AWS SCT).

Pola ini mencakup database Oracle sumber yang ada di lokasi, database Oracle yang diinstal pada instans Amazon Elastic Compute Cloud (Amazon), dan EC2 Amazon Relational Database Service (Amazon RDS) untuk database Oracle. Pola mengubah database ini menjadi Aurora PostgreSQL kompatibel.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Database Oracle di pusat data lokal atau di AWS Cloud.
- Klien SQL diinstal baik pada mesin lokal atau pada sebuah EC2 instance.
- Driver Java Database Connectivity (JDBC) untuk konektor AWS SCT, diinstal pada mesin lokal atau instance EC2 tempat AWS SCT diinstal.

Batasan

- Batas ukuran database: 128 TB
- Jika database sumber mendukung aplikasi komersial off-the-shelf (COTS) atau khusus vendor, Anda mungkin tidak dapat mengonversinya ke mesin basis data lain. Sebelum menggunakan pola ini, konfirmasikan bahwa aplikasi mendukung Aurora PostgreSQL kompatibel.

Versi produk

- Untuk database Oracle yang dikelola sendiri, AWS DMS mendukung semua edisi database Oracle untuk versi 10.2 dan yang lebih baru (untuk versi 10.x), 11g, dan hingga 12.2, 18c, dan 19c. Untuk daftar terbaru versi database Oracle yang didukung (baik yang dikelola sendiri maupun Amazon

RDS for Oracle), lihat Menggunakan database Oracle sebagai sumber untuk AWS DMS dan Menggunakan database PostgreSQL sebagai target untuk AWS DMS.

- Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif. Untuk informasi tentang versi database Oracle yang didukung oleh AWS SCT, lihat dokumentasi AWS SCT.
- Aurora mendukung versi PostgreSQL yang tercantum dalam rilis Amazon Aurora PostgreSQL dan versi mesin.

Arsitektur

Tumpukan teknologi sumber

Salah satu dari yang berikut:

- Database Oracle lokal
- Database Oracle pada sebuah instance EC2
- Instans Amazon RDS for Oracle DB

Tumpukan teknologi target

- Kompatibel dengan Aurora PostgreSQL

Arsitektur target

Arsitektur migrasi data

- Dari database Oracle yang berjalan di AWS Cloud

- Dari database Oracle yang berjalan di pusat data lokal

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.

- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan database sumber.	Untuk menyiapkan database sumber, lihat Menggunakan Oracle Database sebagai sumber AWS SCT dalam dokumentasi AWS SCT.	DBA
Buat EC2 instance untuk AWS SCT.	Buat dan konfigurasikan EC2 instance untuk AWS SCT, jika diperlukan.	DBA
Unduh AWS SCT.	Unduh versi terbaru AWS SCT dan driver terkait. Untuk informasi selengkapnya, lihat Menginstal, memverifikasi, dan memperbarui AWS SCT di dokumentasi AWS SCT.	DBA
Tambahkan pengguna dan izin.	Menambahkan dan memvalidasi pengguna prasyarat dan izin dalam database sumber.	DBA
Buat proyek AWS SCT.	Buat proyek AWS SCT untuk beban kerja, dan sambungkan ke database sumber. Untuk petunjuknya, lihat Membuat proyek AWS SCT dan Menambahkan server	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	database dalam dokumentasi AWS SCT.	
Mengevaluasi kelayakan.	Buat laporan penilaian, yang merangkum item tindakan untuk skema yang tidak dapat dikonversi secara otomatis dan memberikan perkiraan untuk upaya konversi manual. Untuk informasi selengkapnya, lihat Membuat dan meninjau laporan penilaian migrasi database dalam dokumentasi AWS SCT.	DBA

Siapkan database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans DB Amazon RDS target.	Buat instans DB Amazon RDS target, menggunakan Amazon Aurora sebagai mesin database. Untuk petunjuknya, lihat Membuat instans Amazon RDS DB dalam dokumentasi Amazon RDS.	DBA
Ekstrak pengguna, peran, dan izin.	Ekstrak daftar pengguna, peran, dan izin dari database sumber.	DBA
Pengguna peta.	Memetakan pengguna database yang ada untuk pengguna database baru.	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna.	Buat pengguna di database target.	DBA, Pemilik aplikasi
Terapkan peran.	Terapkan peran dari langkah sebelumnya ke database target.	DBA
Periksa opsi, parameter, file jaringan, dan tautan basis data.	Tinjau database sumber untuk opsi, parameter, file jaringan, dan tautan basis data, dan kemudian evaluasi penerapan nya ke basis data target.	DBA
Terapkan pengaturan.	Terapkan pengaturan yang relevan ke database target.	DBA

Transfer objek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan konektivitas AWS SCT.	Konfigurasikan konektivitas AWS SCT ke database target.	DBA
Konversi skema menggunak an AWS SCT.	AWS SCT secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan basis data target. Kode apa pun yang tidak dapat dikonversi oleh alat secara otomatis ditandai dengan jelas sehingga Anda dapat mengonversinya secara manual.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau laporannya.	Tinjau laporan SQL yang dihasilkan dan simpan kesalahan dan peringatan apa pun.	DBA
Terapkan perubahan skema otomatis.	Terapkan perubahan skema otomatis ke database target atau simpan sebagai file.sql.	DBA
Validasi objek.	Validasi bahwa AWS SCT membuat objek pada target.	DBA
Menangani item yang tidak dikonversi.	Menulis ulang, menolak, atau mendesain ulang item apa pun secara manual yang gagal dikonversi secara otomatis.	DBA, Pemilik aplikasi
Terapkan peran dan izin pengguna.	Terapkan peran yang dihasilkan dan izin pengguna dan tinjau pengecualian apa pun.	DBA

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan metodenya.	Tentukan metode untuk memigrasi data.	DBA
Buat contoh replikasi.	Buat instance replikasi dari konsol AWS DMS. Untuk informasi selengkapnya, lihat Bekerja dengan instans replikasi AWS DMS dalam dokumentasi AWS DMS.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir sumber dan target.	Untuk membuat titik akhir, ikuti petunjuk dalam Membuat titik akhir sumber dan target dalam dokumentasi AWS DMS.	DBA
Buat tugas replikasi.	Untuk membuat tugas, lihat Bekerja dengan tugas AWS DMS dalam dokumentasi AWS DMS.	DBA
Mulai tugas replikasi dan pantau log.	Untuk informasi selengkap nya tentang langkah ini, lihat Memantau tugas AWS DMS dalam dokumentasi AWS DMS.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menganalisis dan mengkonve rsi item SQL dalam kode aplikasi.	Gunakan AWS SCT untuk menganalisis dan mengonver si item SQL dalam kode aplikasi. Ketika Anda mengonversi skema database Anda dari satu mesin ke mesin lain, Anda juga perlu memperbarui kode SQL dalam aplikasi Anda untuk berintera ksi dengan mesin database baru, bukan yang lama. Anda dapat melihat, menganalisis, mengedit, dan menyimpan kode SQL yang dikonversi.	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server aplikasi.	Buat server aplikasi baru di AWS.	Pemilik aplikasi
Migrasikan kode aplikasi.	Migrasikan kode aplikasi ke server baru.	Pemilik aplikasi
Konfigurasikan server aplikasi.	Konfigurasikan server aplikasi untuk database target dan driver.	Pemilik aplikasi
Perbaiki kode.	Perbaiki kode apa pun yang khusus untuk mesin basis data sumber di aplikasi Anda.	Pemilik aplikasi
Optimalkan kode.	Optimalkan kode aplikasi Anda untuk mesin basis data target.	Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Potong ke database target.	Lakukan cutover ke database baru.	DBA
Kunci aplikasi.	Kunci aplikasi dari perubahan lebih lanjut.	Pemilik aplikasi
Validasi perubahan.	Validasi bahwa semua perubahan disebarkan ke database target.	DBA
Redirect ke database target.	Arahkan server aplikasi baru ke database target.	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa semuanya.	Lakukan pemeriksaan sistem yang final dan komprehensif.	Pemilik aplikasi
Pergilah hidup.	Selesaikan tugas cutover akhir.	Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya sementara.	Matikan sumber daya AWS sementara seperti instans replikasi AWS DMS dan instans yang EC2 digunakan untuk AWS SCT.	DBA, Pemilik aplikasi
Perbarui umpan balik.	Perbarui umpan balik tentang proses AWS DMS untuk tim internal.	DBA, Pemilik aplikasi
Merevisi proses dan template.	Merevisi proses AWS DMS dan meningkatkan template jika perlu.	DBA, Pemilik aplikasi
Validasi dokumen.	Meninjau dan memvalidasi dokumen proyek.	DBA, Pemilik aplikasi
Kumpulkan metrik.	Kumpulkan metrik untuk mengevaluasi waktu migrasi, persentase penghematan biaya manual versus alat, dan sebagainya.	DBA, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tutup proyek.	Tutup proyek migrasi dan berikan umpan balik kepada pemangku kepentingan.	DBA, Pemilik aplikasi

Sumber daya terkait

Referensi

- Menggunakan Database Oracle sebagai Sumber AWS DMS
- Menggunakan Database PostgreSQL sebagai Target untuk AWS Database Migration Service
- Oracle Database 11g/12c ke Amazon Aurora dengan PostgreSQL Kompatibilitas (9.6.x) Playbook Migrasi
- Oracle Database 19c ke Amazon Aurora dengan PostgreSQL Kompatibilitas (12.4) Playbook Migrasi
- Migrasi database Amazon RDS for Oracle ke Amazon Aurora PostgreSQL Edisi yang kompatibel
- Layanan Migrasi Data AWS
- Alat Konversi Skema AWS
- Bermigrasi dari Oracle ke Amazon Aurora
- Harga Amazon RDS

Tutorial dan video

- Step-by-StepPanduan Migrasi Database
- Memulai dengan AWS DMS
- Memulai dengan Amazon RDS
- Layanan Migrasi Data AWS (video)
- Migrasi database Oracle ke PostgreSQL (video)

Informasi tambahan

.

Memigrasi data dari database Oracle lokal ke Aurora PostgreSQL

Michelle Deng dan Shunan Xiang, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk migrasi data dari database Oracle lokal ke Amazon Aurora PostgreSQL Edisi yang kompatibel dengan Amazon Aurora. Ini menargetkan strategi migrasi data online dengan jumlah downtime minimal untuk database Oracle multi-terabyte yang berisi tabel besar dengan aktivitas bahasa manipulasi data tinggi (DML/bahasa manipulasi data tinggi). Database siaga Oracle Active Data Guard digunakan sebagai sumber untuk membongkar migrasi data dari database utama. Replikasi dari database utama Oracle ke standby dapat ditangguhkan selama beban penuh untuk menghindari kesalahan ORA-01555.

Kolom tabel di kunci utama (PKs) atau kunci asing (FKs), dengan tipe data NUMBER, biasanya digunakan untuk menyimpan bilangan bulat di Oracle. Kami menyarankan Anda mengonversinya menjadi INT atau BIGINT di PostgreSQL untuk kinerja yang lebih baik. Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengubah pemetaan tipe data default untuk kolom PK dan FK. (Untuk informasi lebih lanjut, lihat posting AWS blog Mengonversi tipe data NUMBER dari Oracle ke PostgreSQL.) Migrasi data dalam pola ini menggunakan AWS Database Migration Service (AWS DMS) untuk pemuatan penuh dan pengambilan data perubahan (CDC).

Anda juga dapat menggunakan pola ini untuk memigrasikan database Oracle lokal ke Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL, atau database Oracle yang di-host di Amazon Elastic Compute Cloud (Amazon) ke EC2 Amazon RDS for PostgreSQL atau Aurora PostgreSQL yang kompatibel dengan Aurora.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber Oracle di pusat data lokal dengan siaga Active Data Guard yang dikonfigurasi
- AWS Direct Connect dikonfigurasi antara pusat data lokal dan AWS Cloud
- Keakraban dengan menggunakan database Oracle sebagai sumber AWS DMS
- Keakraban dengan menggunakan database PostgreSQL sebagai target AWS DMS

Batasan

- Cluster database Amazon Aurora dapat dibuat dengan penyimpanan hingga 128 TiB. Instans database Amazon RDS for PostgreSQL dapat dibuat dengan penyimpanan hingga 64 TiB. Untuk informasi penyimpanan terbaru, lihat penyimpanan dan keandalan Amazon Aurora serta penyimpanan instans Amazon RDS DB dalam dokumentasi AWS.

Versi produk

- AWS DMS mendukung semua edisi database Oracle untuk versi 10.2 dan yang lebih baru (untuk versi 10.x), 11g dan hingga 12.2, 18c, dan 19c. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS dalam dokumentasi AWS.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal dengan siaga Oracle Active Data Guard yang dikonfigurasi

Tumpukan teknologi target

- Kompatibel dengan Aurora PostgreSQL

Arsitektur migrasi data

Alat

- AWS DMS - AWS Database Migration Service (AWS DMS) mendukung beberapa basis data sumber dan target. Lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS dalam dokumentasi AWS DMS untuk daftar versi dan edisi database target dan sumber Oracle yang didukung. Jika database sumber tidak didukung oleh AWS DMS, Anda harus memilih metode lain untuk memigrasikan data di Fase 6 (di bagian Epics). Catatan penting: Karena ini adalah migrasi heterogen, Anda harus terlebih dahulu memeriksa untuk melihat apakah database mendukung aplikasi komersial off-the-shelf (COTS). Jika aplikasinya COTS, konsultasikan dengan vendor untuk mengonfirmasi bahwa Aurora PostgreSQL kompatibel didukung sebelum melanjutkan. Untuk informasi selengkapnya, lihat Panduan Step-by-Step Migrasi AWS DMS di dokumentasi AWS.
- AWS SCT - AWS Schema Conversion Tool (AWS SCT) memfasilitasi migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom

ke format yang kompatibel dengan database target. Kode khusus yang dikonversi alat mencakup tampilan, prosedur tersimpan, dan fungsi. Kode apa pun yang tidak dapat dikonversi oleh alat secara otomatis ditandai dengan jelas sehingga Anda dapat mengonversinya sendiri.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target.		DBA
Instal AWS SCT dan driver.		DBA
Tambahkan dan validasi pengguna prasyarat AWS SCT dan database sumber hibah.		DBA
Buat proyek AWS SCT untuk beban kerja, dan sambungkan ke database sumber.		DBA
Menghasilkan laporan penilaian dan mengevaluasi kelayakan.		DBA, Pemilik aplikasi

Siapkan database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database target yang kompatibel dengan Aurora PostgreSQL.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak daftar pengguna, peran, dan hibah dari database sumber.		DBA
Memetakan pengguna database yang ada untuk pengguna database baru.		Pemilik aplikasi
Buat pengguna di database target.		DBA
Terapkan peran dari langkah sebelumnya ke database target yang kompatibel dengan Aurora PostgreSQL.		DBA
Tinjau opsi basis data, parameter, file jaringan, dan tautan basis data dari basis data sumber, dan evaluasi penerapannya ke basis data target.		DBA, Pemilik aplikasi
Terapkan pengaturan yang relevan ke database target.		DBA

Mempersiapkan konversi kode objek database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan konektivitas AWS SCT ke database target.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konversikan skema di AWS SCT, dan simpan kode yang dikonversi sebagai file.sql.		DBA, Pemilik aplikasi
Secara manual mengonversi objek database apa pun yang gagal dikonversi secara otomatis.		DBA, Pemilik aplikasi
Optimalkan konversi kode database.		DBA, Pemilik aplikasi
Pisahkan file.sql menjadi beberapa file.sql berdasarkan jenis objek.		DBA, Pemilik aplikasi
Validasi skrip SQL dalam database target.		DBA, Pemilik aplikasi

Mempersiapkan migrasi data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans replikasi AWS DMS.		DBA
Buat titik akhir sumber dan target.	Jika tipe data PKs dan FKs dikonversi dari NUMBER di Oracle ke BIGINT di PostgreSQL, pertimbangkan untuk menentukan atribut koneksi saat Anda membuat titik akhir sumber. `numberDataTypeScale=-2`	DBA

Migrasikan data — beban penuh

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skema dan tabel dalam database target.		DBA
Buat tugas beban penuh AWS DMS dengan mengelomp okkan tabel atau membagi tabel besar berdasarkan ukuran tabel.		DBA
Hentikan aplikasi pada database sumber Oracle untuk waktu yang singkat.		Pemilik aplikasi
Verifikasi bahwa database siaga Oracle sinkron dengan database utama, dan hentikan replikasi dari database utama ke database siaga.		DBA, Pemilik aplikasi
Mulai aplikasi pada database sumber Oracle.		Pemilik aplikasi
Mulai tugas beban penuh AWS DMS secara paralel dari database siaga Oracle ke database yang kompatibel dengan Aurora PostgreSQL.		DBA
Buat PKs dan indeks sekunder setelah beban penuh selesai.		DBA
Validasi data.		DBA

Migrasi data — CDC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas replikasi berkelanjutan AWS DMS dengan menentukan waktu mulai CDC kustom atau nomor perubahan sistem (SCN) saat siaga Oracle disinkronkan dengan database utama, dan sebelum aplikasi dimulai ulang di tugas sebelumnya.		DBA
Mulai tugas AWS DMS secara paralel untuk mereplikasi perubahan yang sedang berlangsung dari database siaga Oracle ke database yang kompatibel dengan Aurora PostgreSQL.		DBA
Membangun kembali replikasi dari database utama Oracle ke database siaga.		DBA
Memantau log dan menghentikan aplikasi pada database Oracle ketika target database Aurora PostgreSQL kompatibel hampir sinkron dengan database sumber Oracle.		DBA, Pemilik aplikasi
Hentikan tugas AWS DMS ketika target sepenuhnya disinkronkan dengan database Oracle sumber.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat FKs dan memvalidasi data dalam database target.		DBA
Buat fungsi, tampilan, pemicu, urutan, dan jenis objek lainnya dalam database target.		DBA
Terapkan hibah peran dalam database target.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan AWS SCT untuk menganalisis dan mengonversi pernyataan SQL di dalam kode aplikasi.		Pemilik aplikasi
Buat server aplikasi baru di AWS.		Pemilik aplikasi
Migrasikan kode aplikasi ke server baru.		Pemilik aplikasi
Konfigurasikan server aplikasi untuk database target dan driver.		Pemilik aplikasi
Perbaiki kode apa pun yang khusus untuk mesin basis data sumber dalam aplikasi.		Pemilik aplikasi
Optimalkan kode aplikasi untuk database target.		Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Arahkan server aplikasi baru ke database target.		DBA, Pemilik aplikasi
Lakukan pemeriksaan kewarasan.		DBA, Pemilik aplikasi
Pergilah hidup.		DBA, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, Administrator sistem
Meninjau dan memvalidasi dokumen proyek.		DBA, Pemilik aplikasi
Kumpulkan metrik untuk waktu migrasi, persentase penggunaan manual versus alat, penghematan biaya, dan data serupa.		DBA, Pemilik aplikasi
Tutup proyek dan berikan umpan balik.		DBA, Pemilik aplikasi

Sumber daya terkait

Referensi

- Database Oracle ke Aurora PostgreSQL kompatibel: Migrasi Playbook
- Migrasi Amazon RDS for Oracle Database ke Amazon Aurora MySQL

- Situs AWS DMS
- Dokumentasi AWS DMS
- Situs AWS SCT
- Dokumentasi AWS SCT
- Bermigrasi dari Oracle ke Amazon Aurora

Tutorial

- Memulai dengan AWS DMS
- Memulai dengan Amazon RDS
- Step-by-StepPanduan Layanan Migrasi Database AWS

Bermigrasi dari SAP ASE ke Amazon RDS for SQL Server menggunakan AWS DMS

Amit Kumar, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan database SAP Adaptive Server Enterprise (ASE) ke instans DB Layanan Relational Database Amazon (Amazon RDS) yang menjalankan Microsoft SQL Server. Basis data sumber dapat ditemukan di pusat data lokal atau di instans Amazon Elastic Compute Cloud EC2 (Amazon). Pola ini menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data dan (opsional) alat rekayasa perangkat lunak berbantuan komputer (CASE) untuk mengonversi skema database.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database SAP ASE di pusat data lokal atau pada instans EC2
- Target Amazon RDS for SQL Server database yang aktif dan berjalan

Batasan

- Batas ukuran database: 64 TB

Versi produk

- SAP ASE versi 15.7 atau 16.x saja. Untuk informasi terbaru, lihat Menggunakan Database SAP sebagai Sumber untuk AWS DMS.
- Untuk database target Amazon RDS, AWS DMS mendukung versi Microsoft SQL Server di Amazon RDS untuk edisi Enterprise, Standard, Web, dan Express. Untuk informasi terbaru tentang versi yang didukung, lihat dokumentasi AWS DMS. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif.

Arsitektur

Tumpukan teknologi sumber

- Database SAP ASE yang ada di tempat atau di instans Amazon EC2

Tumpukan teknologi target

- Instans Amazon RDS for SQL Server DB

Arsitektur sumber dan target

Dari database SAP ASE di Amazon EC2 ke Amazon RDS untuk instans DB Amazon RDS for SQL Server:

Dari database SAP ASE lokal ke instans Amazon RDS for SQL Server DB:

Alat

- AWS Database Migration Service (AWS DMS) adalah layanan web yang dapat Anda gunakan untuk memigrasikan data dari database Anda yang ada di tempat, pada instans Amazon RDS DB, atau dalam database pada EC2 instans, ke database pada layanan AWS seperti Amazon RDS for SQL Server atau instans. EC2 Anda juga dapat memigrasikan database dari layanan AWS ke database lokal. Anda dapat memigrasikan data antara mesin database heterogen atau homogen.
- Untuk konversi skema, Anda dapat menggunakan erwin Data Modeler atau SAP secara opsional. PowerDesigner

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target.		DBA
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, SysAdmin
Pilih jenis instans yang tepat berdasarkan kapasitas,		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
fitur penyimpanan, dan fitur jaringan.		
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA, SysAdmin
Identifikasi strategi migrasi aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat virtual private cloud (VPC) dan subnet.		SysAdmin
Buat grup keamanan dan daftar kontrol akses jaringan (ACLs).		SysAdmin
Konfigurasikan dan mulai instans Amazon RDS DB.		SysAdmin

Migrasikan data - opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan skema database secara manual atau gunakan alat CASE seperti erwin Data Modeler atau SAP. PowerDesigner		DBA

Migrasikan data - opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data menggunakan AWS DMS.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.		DBA, SysAdmin, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, SysAdmin
Tinjau dan validasi dokumen proyek.		DBA, SysAdmin, Pemilik aplikasi
Kumpulkan metrik seperti waktu untuk bermigrasi, persentase tugas manual		DBA, SysAdmin, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
versus otomatis, dan penghematan biaya.		
Tutup proyek dan berikan umpan balik.		DBA, SysAdmin, Pemilik aplikasi

Sumber daya terkait

Referensi

- Situs AWS DMS
- Harga Amazon RDS
- Menggunakan Database SAP ASE sebagai Sumber AWS DMS
- Batasan untuk RDS Kustom untuk SQL Server

Tutorial dan video

- Memulai dengan AWS DMS
- Memulai dengan Amazon RDS
- AWS DMS (video)
- Amazon RDS (video)

Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan AWS DMS

Marcelo Fernandes, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift dengan menggunakan AWS Data Migration Service (AWS DMS).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Microsoft SQL Server sumber di pusat data lokal
- Prasyarat lengkap untuk menggunakan database Amazon Redshift sebagai target AWS DMS, seperti yang dibahas dalam dokumentasi AWS DMS

Versi produk

- SQL Server 2005-2019, Enterprise, Standard, Workgroup, Developer, dan edisi Web. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Microsoft SQL Server sebagai Sumber untuk AWS DMS dalam dokumentasi AWS.

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server lokal

Tumpukan teknologi target

- Amazon Redshift

Arsitektur migrasi data

Alat

- AWS DMS adalah layanan migrasi data yang mendukung beberapa jenis basis data sumber dan target. Untuk informasi tentang versi dan edisi database Microsoft SQL Server yang didukung untuk digunakan dengan AWS DMS, lihat Menggunakan Database Microsoft SQL Server sebagai Sumber untuk AWS DMS dalam dokumentasi AWS DMS. Jika AWS DMS tidak mendukung basis data sumber Anda, Anda harus memilih metode alternatif untuk migrasi data.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target dan mesin.		DBA
Identifikasi persyaratan perangkat keras untuk instance server target.		DBA, Administrator sistem
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, Administrator sistem
Pilih jenis instans yang tepat berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.		DBA, Administrator sistem
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA, Administrator sistem
Identifikasi strategi migrasi aplikasi.		DBA, Pemilik aplikasi, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).	Untuk informasi selengkapnya, lihat Bekerja dengan instans DB di VPC dalam dokumentasi AWS.	Administrator sistem
Buat grup keamanan.		Administrator sistem
Konfigurasikan dan mulai cluster Amazon Redshift.	Untuk informasi selengkapnya, lihat Membuat contoh klaster Amazon Redshift di dokumentasi Amazon Redshift.	DBA, Administrator sistem

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dari database Microsoft SQL Server dengan menggunakan AWS DMS.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.		DBA, Pemilik aplikasi, Administrator sistem

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.		DBA, Pemilik aplikasi, Administrator sistem

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya sementara.		DBA, Administrator sistem
Meninjau dan memvalidasi dokumen proyek.		DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik seperti waktu untuk bermigrasi, persentase tugas manual versus otomatis, dan penghematan biaya.		DBA, Pemilik aplikasi, Administrator sistem
Tutup proyek dan berikan umpan balik.		DBA, Pemilik aplikasi, Administrator sistem

Sumber daya terkait

Referensi

- Dokumentasi AWS DMS
- Dokumentasi Amazon Redshift
- Harga Amazon Redshift

Tutorial dan video

- Memulai dengan AWS DMS

- Memulai dengan Amazon Redshift
- Menggunakan database Amazon Redshift sebagai target untuk AWS Database Migration Service
- AWS DMS (video)

Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT

Neha Thakur, Amazon Web Services

Ringkasan

Pola ini menguraikan langkah-langkah untuk memigrasikan database sumber Microsoft SQL Server lokal ke database target Amazon Redshift dengan menggunakan agen ekstraksi data AWS Schema Conversion Tool (AWS SCT). Agen adalah program eksternal yang terintegrasi dengan AWS SCT tetapi melakukan transformasi data di tempat lain dan berinteraksi dengan layanan AWS lainnya atas nama Anda.

Prasyarat dan batasan

Prasyarat

- Database sumber Microsoft SQL Server yang digunakan untuk beban kerja gudang data di pusat data lokal
- Akun AWS yang aktif

Versi produk

- Microsoft SQL Server versi 2008 atau yang lebih baru. Untuk daftar terbaru versi yang didukung, lihat dokumentasi AWS SCT.

Arsitektur

sumber tumpukan teknologi

- Database Microsoft SQL Server lokal

Target tumpukan teknologi

- Amazon Redshift

Arsitektur migrasi data

Alat

- AWS Schema Conversion Tool (AWS SCT) menangani migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target. Ketika basis data sumber dan target sangat berbeda, Anda dapat menggunakan agen AWS SCT untuk melakukan transformasi data tambahan. Untuk informasi selengkapnya, lihat Memigrasi Data dari Gudang Data Lokal ke Amazon Redshift di dokumentasi AWS.

Praktik terbaik

- Praktik terbaik untuk AWS SCT
- Praktik terbaik untuk Amazon Redshift

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi dan mesin basis data sumber dan target.		DBA
Identifikasi persyaratan perangkat keras untuk instance server target.		DBA, SysAdmin
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, SysAdmin
Pilih jenis instans yang tepat (kapasitas, fitur penyimpanan, fitur jaringan).		DBA, SysAdmin
Identifikasi persyaratan keamanan akses jaringan		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
untuk basis data sumber dan target.		
Pilih strategi migrasi aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat virtual private cloud (VPC) dan subnet.		SysAdmin
Buat grup keamanan.		SysAdmin
Konfigurasikan dan mulai cluster Amazon Redshift.		SysAdmin

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data menggunakan agen ekstraksi data AWS SCT.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi yang dipilih.		DBA, SysAdmin, Pemilik aplikasi

Potong ke database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.		DBA, SysAdmin, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, SysAdmin
Tinjau dan validasi dokumen proyek.		DBA, SysAdmin, Pemilik aplikasi
Kumpulkan metrik seperti waktu untuk bermigrasi, persentase tugas manual versus otomatis, dan penghematan biaya.		DBA, SysAdmin, Pemilik aplikasi
Tutup proyek dan berikan umpan balik apa pun.		DBA, SysAdmin, Pemilik aplikasi

Sumber daya terkait

Referensi

- Panduan Pengguna AWS SCT
- Menggunakan Agen Ekstraksi Data
- Harga Amazon Redshift

Tutorial dan video

- Memulai dengan AWS Schema Conversion Tool

- Memulai dengan Amazon Redshift

Migrasi aplikasi lama dari Oracle Pro*C ke ECPG

Sai Parthasaradhi dan Mahesh Balumuri, Amazon Web Services

Ringkasan

Sebagian besar aplikasi lama yang telah tertanam kode SQL menggunakan precompiler Oracle Pro*C untuk mengakses database. Ketika Anda memigrasikan database Oracle ini ke Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL atau Amazon Aurora PostgreSQL Compatible Edition, Anda harus mengonversi kode aplikasi Anda ke format yang kompatibel dengan precompiler di PostgreSQL, yang disebut ECPG. Pola ini menjelaskan cara mengonversi kode Oracle Pro*C ke padanannya di PostgreSQL ECPG.

Untuk informasi selengkapnya tentang Pro*C, lihat dokumentasi Oracle. Untuk pengantar singkat tentang ECPG, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database yang kompatibel dengan Amazon RDS for PostgreSQL atau Aurora PostgreSQL
- Database Oracle berjalan di tempat

Alat

- Paket PostgreSQL tercantum di bagian berikutnya.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat sumber terbuka untuk berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda. Dengan konfigurasi minimal, Anda dapat menjalankan perintah AWS CLI yang menerapkan fungsionalitas yang setara dengan yang disediakan oleh AWS Management Console berbasis browser dari prompt perintah.

Epik

Mengatur lingkungan build pada CentOS atau RHEL

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal paket PostgreSQL.	Instal paket PostgreSQL yang diperlukan dengan menggunakan perintah berikut.
	`yum update -y`
	`yum install -y yum-utils`
	`rpm -ivh https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm`
	`dnf -qy module disable postgresql`
		Pengembang aplikasi, DevOps insinyur

Instal file header dan pustaka.	Instal `postgresql12-devel` paket, yang berisi file header dan pustaka, dengan menggunakan perintah berikut. Instal paket di lingkungan pengembangan dan runtime untuk menghindari kesalahan di lingkungan runtime.
	`dnf -y install postgresql12-devel`
	`yum install ncompress zip ghostscript jq unzip wget git -y`
		Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Untuk lingkungan pengembangan saja, jalankan juga perintah berikut.
	`yum install zlib-devel make -y`
	`ln -s /usr/pgsql-12/bin/ecpg /usr/bin/`	

Konfigurasikan variabel jalur lingkungan.	Mengatur jalur lingkungan untuk pustaka klien PostgreSQL. `export PATH=$PATH:/usr/pgsql-12/bin`	Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal perangkat lunak tambahan seperlunya.	Jika diperlukan, instal pgLoader sebagai pengganti SQL* loader di Oracle.
	`wget -O /etc/yum.repos.d/pgloader-ccl.repo https://dl.packager.io/srv/opf/pgloader-ccl/master/installer/el/7.repo`
	`yum install pgloader-ccl -y`
	`ln -s /opt/pgloader-ccl/bin/pgloader /usr/bin/`
	Jika Anda memanggil aplikasi Java apa pun dari modul Pro* C, instal Java.
	`yum install java -y`
	Instal ant untuk mengkompilasi kode Java.
	`yum install ant -y`
		Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal AWS CLI.	Instal AWS CLI untuk menjalankan perintah agar berinteraksi dengan layanan AWS seperti AWS Secrets Manager dan Amazon Simple Storage Service (Amazon S3) dari aplikasi Anda.
	`cd /tmp/`
	`curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"`
	`unzip awscliv2.zip`
	`./aws/install -i /usr/local/aws-cli -b /usr/local/bin --update`
		Pengembang aplikasi, DevOps insinyur

Identifikasi program yang akan dikonversi.	Identifikasi aplikasi yang ingin Anda konversi dari Pro* C ke ECPG.	Pengembang aplikasi, Pemilik aplikasi

Konversi kode Pro* C ke ECPG

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus header yang tidak diinginkan.	Hapus `include` header yang tidak diperlukan di PostgreSQL, seperti,, dan. `oci.h` `oratypes` `sqlda`	Pemilik aplikasi, Pengembang aplikasi
Perbarui deklarasi variabel.	Tambahkan `EXEC SQL` pernyataan untuk semua deklarasi variabel yang	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	digunakan sebagai variabel host. Hapus `EXEC SQL VAR` deklarasi seperti berikut ini dari aplikasi Anda. `EXEC SQL VAR query IS STRING(2048);`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui fungsionalitas ROWNUM.	`ROWNUM`Fungsi ini tidak tersedia di PostgreSQL. Ganti ini dengan fungsi `ROW_NUMBER` jendela dalam kueri SQL.
	
	Kode Pro* C:
	
	`SELECT SUBSTR(RTRIM(FILE_NAME,'.txt'),12) INTO :gcpclFileseq FROM (SELECT FILE_NAME FROM DEMO_FILES_TABLE WHERE FILE_NAME LIKE '%POC%' ORDER BY FILE_NAME DESC) FL2 WHERE ROWNUM <=1 ORDER BY ROWNUM;`
	
	Kode ECPG:
	
	`SELECT SUBSTR(RTRIM(FILE_NAME,'.txt'),12) INTO :gcpclFileseq FROM (SELECT FILE_NAME , ROW_NUMBER() OVER (ORDER BY FILE_NAME DESC) AS ROWNUM FROM demo_schema.DEMO_FILES_TABLE WHERE FILE_NAME LIKE '%POC%' ORDER BY FILE_NAME DESC) FL2`
		Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`WHERE ROWNUM <=1 ORDER BY ROWNUM;`	

Perbarui parameter fungsi untuk menggunakan variabel alias.	Di PostgreSQL, parameter fungsi tidak dapat digunakan sebagai variabel host. Menimpa mereka dengan menggunakan variabel alias.
	
	Kode Pro* C:
	
	`int processData(int referenceId){ EXEC SQL char col_val[100]; EXEC SQL select column_name INTO :col_val from table_name where col=:referenceId; }`
	
	Kode ECPG:
	
	`int processData(int referenceIdParam){ EXEC SQL int referenceId = referenceIdParam; EXEC SQL char col_val[100]; EXEC SQL select column_name INTO :col_val from table_name where col=:referenceId; }`
		Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui jenis struct.	Tentukan `struct` tipe `EXEC SQL BEGIN` dan `END` blok dengan `typedef` jika variabel `struct` tipe digunakan sebagai variabel host. Jika `struct` jenis didefinisikan dalam file header (`.h`), sertakan file dengan pernyataan `EXEC SQL include`. Kode Pro* C: Berkas header (`demo.h`) `struct s_partiti on_ranges { char sc_table_ group[31]; char sc_table_ name[31]; char sc_range_ value[10]; }; struct s_partiti on_ranges_ind { short ss_table_ group; short ss_table_ name; short ss_range_ value; };` Kode ECPG: Berkas header (`demo.h`)	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

```
EXEC SQL BEGIN DECLARE
 SECTION;
typedef struct
{
  char   sc_table_
group[31];
  char   sc_table_
name[31];
  char   sc_range_
value[10];
} s_partition_ranges;
typedef struct
{
  short    ss_table_
group;
  short    ss_table_
name;
  short    ss_range_
value;
} s_partition_ranges
_ind;
EXEC SQL END DECLARE
 SECTION;
```

Berkas Pro*C () `demo.pc`

```
#include "demo.h"
struct s_partiti
on_ranges gc_partit
ion_data[MAX_PART_
TABLE] ;
struct s_partiti
on_ranges_ind
 gc_partition_data_
ind[MAX_PART_TABLE] ;
```

File ECPG () `demo.pc`

```
exec sql include
 "demo.h"
```

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`EXEC SQL BEGIN DECLARE SECTION; s_partition_ranges gc_partition_data[ MAX_PART_TABLE] ; s_partition_ranges_ind gc_partition_data_ ind[MAX_PART_TABLE] ; EXEC SQL END DECLARE SECTION;`	

Ubah logika untuk mengambil dari kursor.	Untuk mengambil beberapa baris dari kursor dengan menggunakan variabel array, ubah kode yang akan digunakan. `FETCH FORWARD`
	
	Kode Pro* C:
	
	`EXEC SQL char aPoeFiles [MAX_FILES][FILENA ME_LENGTH]; EXEC SQL FETCH filename_ cursor into :aPoeFile s;`
	
	Kode ECPG:
	
	`EXEC SQL char aPoeFiles [MAX_FILES][FILENA ME_LENGTH]; EXEC SQL int fetchSize = MAX_FILES; EXEC SQL FETCH FORWARD :fetchSiz e filename_cursor into :aPoeFiles;`
		Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ubah panggilan paket yang tidak memiliki nilai pengembalian.	Fungsi paket Oracle yang tidak memiliki nilai kembali harus dipanggil dengan variabel indikator. Jika aplikasi Anda menyertakan beberapa fungsi yang memiliki nama yang sama atau jika fungsi tipe yang tidak dikenal menghasilkan kesalahan runtime, ketikkan nilai ke tipe data.
	
	Kode Pro* C:
	
	`void ProcessData (char *data , int id) { EXEC SQL EXECUTE BEGIN pkg_demo.process_data (:data, :id); END; END-EXEC; }`
	
	Kode ECPG:
	
	`void ProcessData (char *dataParam, int idParam ) { EXEC SQL char *data = dataParam;`
		Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>EXEC SQL int id
	= idParam;
	EXEC SQL short
	rowInd;
	EXEC SQL short
	rowInd = 0;
	EXEC SQL SELECT
	pkg_demo.process_data
	(
	
	inp_data => :data::te
	xt,
	
	inp_id => :id
	)
	INTO :rowInd;
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tulis ulang variabel SQL_CURSOR.	Tulis ulang `SQL_CURSOR` variabel dan implementasinya.
	
	Kode Pro* C:
	
	`/* SQL Cursor */`
	`SQL_CURSOR`
	`demo_cursor;`
	`EXEC SQL`
	`ALLOCATE :demo_cursor;`
	`EXEC SQL EXECUTE`
	`BEGIN`
	`pkg_demo.`
	`get_cursor(`
	`demo_cur=`
	`>:demo_cursor`
	`);`
	`END;`
	`END-EXEC;`
	
	Kode ECPG:
	
	`EXEC SQL DECLARE`
	`demo_cursor CURSOR FOR`
	`SELECT`
	`* from`
	`pkg_demo.open_file`
	`name_rc(`
	`demo_cur=`
	`>refcursor`
	`) ;`
	`EXEC SQL char open_file`
	`name_rcInd[100];`
	`# As the below function`
	`returns cursor_name`
	`as`
	`# return we need to`
	`use char[] type as`
	`indicator.`
		Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`EXEC SQL SELECT pkg_demo.get_cursor ( demo_cur= >'demo_cursor' ) INTO :open_fil ename_rcInd;`	
Terapkan pola migrasi umum.	• Ubah kueri SQL sehingga kompatibel dengan PostgreSQL. • Pindahkan blok anonim, jika tidak didukung di ECPG, ke database. • Hapus `dbms_appl ication_info` logika, yang tidak didukung oleh PostgreSQL. • Pindahkan `EXEC SQL COMMIT` pernyataan setelah kursor ditutup. Jika Anda melakukan kueri saat berada di loop untuk mengambil catatan dari kursor, kursor ditutup dan kesalahan kursor tidak ada ditampilkan. • Untuk informasi tentang penanganan pengecual ian di ECPG dan kode kesalahan, lihat Penangana n Kesalahan dalam dokumentasi PostgreSQL.	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan debugging, jika diperlukan.	Untuk menjalankan program ECPG dalam mode debug, tambahkan perintah berikut di dalam blok fungsi utama. `ECPGdebug(1, stderr);`	Pengembang aplikasi, Pemilik aplikasi

Kompilasi program ECPG

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file yang dapat dieksekusi untuk ECPG.	Jika Anda memiliki file sumber SQL C tertanam bernama`prog1.pgc` , Anda dapat membuat program yang dapat dieksekusi dengan menggunakan urutan perintah berikut.
	`ecpg prog1.pgc`
	`cc -I/usr/local/pgsql/include -c prog1.c`
	`cc -o prog1 prog1.o -L/usr/local/pgsql/lib -lecpg`
		Pengembang aplikasi, Pemilik aplikasi

Buat file make untuk kompilasi.	Buat file make untuk mengkompilasi program ECPG, seperti yang ditunjukkan pada file contoh berikut. `CFLAGS ::= $(CFLAGS) -I/usr/pgsql-12/include -g -Wall`	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`LDFLAGS ::= $(LDFLAGS ) -L/usr/pgsql-12/li b -Wl,-rpath,/usr/pg sql-12/lib`
	`LDLIBS ::= $(LDLIBS) - lecpg`
	`PROGRAMS = test`
	`.PHONY: all clean`
	`%.c: %.pgc`
	`ecpg $<`
	`all: $(PROGRAMS)`
	`clean:`
	`rm -f $(PROGRAM S) $(PROGRAMS:%=%.c) $(PROGRAMS:%=%.o)`	

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji kode.	Uji kode aplikasi yang dikonversi untuk memastika n bahwa itu berfungsi dengan benar.	Pengembang aplikasi, Pemilik aplikasi, Insinyur uji

Sumber daya terkait

- ECPG - SQL Tertanam di C (dokumentasi PostgreSQL)
- Penanganan Kesalahan (dokumentasi PostgreSQL)
- Mengapa Menggunakan Oracle Pro*C/C++ Precompiler (dokumentasi Oracle)

Informasi tambahan

PostgreSQL memiliki precompiler SQL tertanam, ECPG, yang setara dengan precompiler Oracle Pro*C. ECPG mengkonversi program C yang telah tertanam pernyataan SQL untuk kode C standar

dengan mengganti panggilan SQL dengan panggilan fungsi khusus. File output kemudian dapat diproses dengan rantai alat kompiler C apa pun.

File input dan output

ECPG mengonversi setiap file input yang Anda tentukan pada baris perintah ke file output C yang sesuai. Jika nama file input tidak memiliki ekstensi file, .pgc diasumsikan. Ekstensi file diganti dengan `.c` untuk membangun nama file output. Namun, Anda dapat mengganti nama file output default dengan menggunakan `-o` opsi.

Jika Anda menggunakan tanda hubung (`-`) sebagai nama file input, ECPG membaca program dari input standar dan menulis ke output standar, kecuali jika Anda menggantinya dengan menggunakan opsi. `-o`

File header

Ketika kompiler PostgreSQL mengkompilasi file kode C yang telah diproses sebelumnya, ia mencari file header ECPG di direktori PostgreSQL. `include` Oleh karena itu, Anda mungkin harus menggunakan `-I` opsi untuk mengarahkan kompiler ke direktori yang benar (misalnya,`-I/usr/local/pgsql/include`).

Perpustakaan

Program yang menggunakan kode C dengan SQL tertanam harus ditautkan dengan `libecpg` perpustakaan. Misalnya, Anda dapat menggunakan opsi `-L/usr/local/pgsql/lib -lecpg` tautan.

Aplikasi ECPG yang dikonversi memanggil fungsi di `libpq` perpustakaan melalui perpustakaan SQL tertanam (`ecpglib`), dan berkomunikasi dengan server PostgreSQL dengan menggunakan protokol standar. frontend/backend

Migrasikan kolom virtual yang dihasilkan dari Oracle ke PostgreSQL

Veeranjaneyulu Grandhi, Rajesh Madiwale, dan Ramesh Pathuri, Amazon Web Services

Ringkasan

Di versi 11 dan sebelumnya, PostgreSQL tidak menyediakan fitur yang secara langsung setara dengan kolom virtual Oracle. Menangani kolom virtual yang dihasilkan saat bermigrasi dari Oracle Database ke PostgreSQL versi 11 atau sebelumnya sulit karena dua alasan:

- Kolom virtual tidak terlihat selama migrasi.
- PostgreSQL tidak mendukung `generate` ekspresi sebelum versi 12.

Namun, ada solusi untuk meniru fungsionalitas serupa. Bila Anda menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data dari Oracle Database ke PostgreSQL versi 11 dan yang lebih lama, Anda dapat menggunakan fungsi pemicu untuk mengisi nilai dalam kolom virtual yang dihasilkan. Pola ini memberikan contoh Oracle Database dan kode PostgreSQL yang dapat Anda gunakan untuk tujuan ini. Di AWS, Anda dapat menggunakan Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL atau Amazon Aurora PostgreSQL Edisi yang kompatibel dengan PostgreSQL untuk database PostgreSQL Anda.

Dimulai dengan PostgreSQL versi 12, kolom yang dihasilkan didukung. Kolom yang dihasilkan dapat dihitung dari nilai kolom lain dengan cepat, atau dihitung dan disimpan. Kolom yang dihasilkan PostgreSQL mirip dengan kolom virtual Oracle.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Oracle sumber
- Target database PostgreSQL (di Amazon RDS untuk PostgreSQL atau Aurora PostgreSQL yang kompatibel dengan Aurora)
- Keahlian pengkodean PL/PGSQL

Batasan

- Berlaku hanya untuk versi PostgreSQL sebelum versi 12.

- Berlaku untuk Oracle Database versi 11g atau yang lebih baru.
- Kolom virtual tidak didukung dalam alat migrasi data.
- Berlaku hanya untuk kolom yang didefinisikan dalam tabel yang sama.
- Jika kolom yang dihasilkan virtual mengacu pada fungsi deterministik yang ditentukan pengguna, itu tidak dapat digunakan sebagai kolom kunci partisi.
- Output dari ekspresi harus berupa nilai skalar. Itu tidak dapat mengembalikan tipe data yang disediakan Oracle, tipe yang ditentukan pengguna, atau. `LOB LONG RAW`
- Indeks yang didefinisikan terhadap kolom virtual setara dengan indeks berbasis fungsi di PostgreSQL.
- Statistik tabel harus dikumpulkan.

Alat

- pgAdmin 4 adalah alat manajemen open source untuk PostgreSQL. Alat ini menyediakan antarmuka grafis yang menyederhanakan pembuatan, pemeliharaan, dan penggunaan objek database.
- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi gratis untuk bekerja dengan SQL di database Oracle baik dalam penerapan tradisional maupun cloud.

Epik

Buat tabel basis data sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel Database Oracle sumber.	Di Oracle Database, buat tabel dengan kolom virtual yang dihasilkan dengan menggunakan pernyataan berikut. `CREATE TABLE test.generated_column ( CODE NUMBER, STATUS VARCHAR2(12) DEFAULT 'PreOpen', FLAG CHAR(1) GENERATED ALWAYS AS (CASE`	DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>UPPER(STATUS) WHEN 'OPEN' THEN 'N' ELSE 'Y' END) VIRTUAL VISIBLE);</pre> Dalam tabel sumber ini, data di `STATUS` kolom dimigrasikan melalui AWS DMS ke database target. `FLAG`Kolom, bagaimanapun, diisi dengan menggunakan `generate by` fungsionalitas, sehingga kolom ini tidak terlihat oleh AWS DMS selama migrasi. Untuk mengimplementasikan fungsionalitas`generated by`, Anda harus menggunakan pemicu dan fungsi dalam database target untuk mengisi nilai di `FLAG` kolom, seperti yang ditunjukkan dalam epik berikutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel PostgreSQL target di AWS.	Buat tabel PostgreSQL di AWS dengan menggunakan pernyataan berikut. `CREATE TABLE test.generated_column ( code integer not null, status character varying(12) not null , flag character(1) );` Dalam tabel ini, `status` kolom adalah kolom standar. `flag`Kolom akan menjadi kolom yang dihasilkan berdasarkan data di `status` kolom.	DBA, Pengembang aplikasi

Buat fungsi pemicu untuk menangani kolom virtual di PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pemicu PostgreSQL.	Di PostgreSQL, buat pemicu. `CREATE TRIGGER tgr_gen_column AFTER INSERT OR UPDATE OF status ON test.generated_column FOR EACH ROW EXECUTE FUNCTION test.tgf_gen_column();`	DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi pemicu PostgreSQL.	Di PostgreSQL, buat fungsi untuk pemicu. Fungsi ini mengisi kolom virtual yang disisipkan atau diperbarui oleh aplikasi atau AWS DMS, dan memvalidasi data. `CREATE OR REPLACE FUNCTION test.tgf_gen_column() RETURNS trigger AS $VIRTUAL_COL$ BEGIN IF (TG_OP = 'INSERT') THEN IF (NEW.flag IS NOT NULL) THEN RAISE EXCEPTION 'ERROR: cannot insert into column "flag"' USING DETAIL = 'Column "flag" is a generated column.'; END IF; END IF; IF (TG_OP = 'UPDATE') THEN IF (NEW.flag::VARCHAR != OLD.flag::varchar) THEN RAISE EXCEPTION 'ERROR: cannot update column "flag"' USING DETAIL = 'Column "flag" is a generated column.'; END IF; END IF; IF TG_OP IN ('INSERT','UPDATE') THEN`	DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`IF (old.flag is NULL) OR (coalesce(old.status,'') != coalesce(new.status,'')) THEN UPDATE test.generated_column SET flag = (CASE UPPER(status) WHEN 'OPEN' THEN 'N' ELSE 'Y' END) WHERE code = new.code; END IF; END IF; RETURN NEW; END $VIRTUAL_COL$ LANGUAGE plpgsql;`	

Uji migrasi data dengan menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh replikasi.	Untuk membuat instance replikasi, ikuti petunjuk dalam dokumentasi AWS DMS. Instans replikasi harus berada di cloud pribadi virtual (VPC) yang sama dengan basis data sumber dan target Anda.	DBA, Pengembang aplikasi
Buat titik akhir sumber dan target.	Untuk membuat titik akhir, ikuti petunjuk dalam dokumentasi AWS DMS.	DBA, Pengembang aplikasi
Uji koneksi titik akhir.	Anda dapat menguji koneksi titik akhir dengan menentuka	DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n VPC dan instance replikasi dan memilih Run test.	
Buat dan mulai tugas beban penuh.	Untuk petunjuknya, lihat Membuat Tugas dan setelan tugas beban penuh dalam dokumentasi AWS DMS.	DBA, Pengembang aplikasi
Validasi data untuk kolom virtual.	Bandingkan data di kolom virtual di database sumber dan target. Anda dapat memvalidasi data secara manual atau menulis skrip untuk langkah ini.	DBA, Pengembang aplikasi

Sumber daya terkait

- Memulai AWS Database Migration Service (dokumentasi AWS DMS)
- Menggunakan database Oracle sebagai sumber AWS DMS (dokumentasi AWS DMS)
- Menggunakan database PostgreSQL sebagai target AWS DMS (dokumentasi AWS DMS)
- Kolom yang dihasilkan di PostgreSQL (dokumentasi PostgreSQL)
- Fungsi pemicu (dokumentasi PostgreSQL)
- Kolom virtual di Oracle Database (dokumentasi Oracle)

Siapkan fungsionalitas Oracle UTL_FILE pada Aurora PostgreSQL yang kompatibel

Rakesh Raghav dan anuradha chintha, Amazon Web Services

Ringkasan

Sebagai bagian dari perjalanan migrasi Anda dari Oracle ke Amazon Aurora PostgreSQL Compatible Edition di Amazon Web Services (AWS) Cloud, Anda mungkin menghadapi beberapa tantangan. Misalnya, memigrasikan kode yang bergantung pada `UTL_FILE` utilitas Oracle selalu menjadi tantangan. Di Oracle PL/SQL, `UTL_FILE` paket ini digunakan untuk operasi file, seperti baca dan tulis, bersama dengan sistem operasi yang mendasarinya. `UTL_FILE`Utilitas ini berfungsi untuk sistem server dan mesin klien.

Amazon Aurora PostgreSQL kompatibel adalah penawaran database terkelola. Karena itu, tidak mungkin mengakses file di server database. Pola ini memandu Anda melalui integrasi Amazon Simple Storage Service (Amazon S3) dan Amazon Aurora PostgreSQL yang kompatibel untuk mencapai subset fungsionalitas. `UTL_FILE` Dengan menggunakan integrasi ini, kita dapat membuat dan menggunakan file tanpa menggunakan alat atau layanan ekstrak, transformasi, dan muat (ETL) pihak ketiga.

Secara opsional, Anda dapat mengatur CloudWatch pemantauan Amazon dan notifikasi Amazon SNS.

Kami merekomendasikan pengujian solusi ini secara menyeluruh sebelum menerapkannya di lingkungan produksi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Keahlian AWS Database Migration Service (AWS DMS)
- Keahlian dalam PL/pgSQL pengkodean
- Cluster yang kompatibel dengan Amazon Aurora PostgreSQL
- Ember S3

Batasan

Pola ini tidak menyediakan fungsionalitas untuk bertindak sebagai pengganti `UTL_FILE` utilitas Oracle. Namun, langkah-langkah dan kode sampel dapat ditingkatkan lebih lanjut untuk mencapai tujuan modernisasi database Anda.

Versi produk

- Amazon Aurora PostgreSQL Edisi 11.9 yang kompatibel

Arsitektur

Tumpukan teknologi target

- Kompatibel dengan Amazon Aurora PostgreSQL
- Amazon CloudWatch
- Amazon Simple Notification Service (Amazon SNS)
- Amazon S3

Arsitektur target

Diagram berikut menunjukkan representasi tingkat tinggi dari solusi.

1. File diunggah dari aplikasi ke dalam ember S3.
2. `aws_s3`Ekstensi mengakses data, menggunakan PL/PGSQL, dan mengunggah data ke Aurora PostgreSQL kompatibel.

Alat

- Kompatibel dengan Amazon Aurora PostgreSQL — Amazon Aurora PostgreSQL Compatible Edition adalah mesin basis data relasional yang dikelola sepenuhnya, kompatibel dengan PostgreSQL, dan sesuai dengan Asam. Ini menggabungkan kecepatan dan keandalan database komersial kelas atas dengan efektivitas biaya database sumber terbuka.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat terpadu untuk mengelola layanan AWS Anda. Dengan hanya satu alat untuk mengunduh dan mengonfigurasi, Anda dapat mengontrol beberapa layanan AWS dari baris perintah dan mengotomatiskannya melalui skrip.

- Amazon CloudWatch — Amazon CloudWatch memonitor sumber daya dan penggunaan Amazon S3.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet. Dalam pola ini, Amazon S3 menyediakan lapisan penyimpanan untuk menerima dan menyimpan file untuk konsumsi dan transmisi ke dan dari cluster Aurora PostgreSQL yang kompatibel.
- aws_s3 - Ekstensi `aws_s3` mengintegrasikan Amazon S3 dan Aurora PostgreSQL kompatibel.
- Amazon SNS - Amazon Simple Notification Service (Amazon SNS) mengoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien. Dalam pola ini, Amazon SNS digunakan untuk mengirim notifikasi.
- pgAdmin - pgAdmin adalah alat manajemen sumber terbuka untuk Postgres. pgAdmin 4 menyediakan antarmuka grafis untuk membuat, memelihara, dan menggunakan objek database.

Kode

Untuk mencapai fungsionalitas yang diperlukan, pola membuat beberapa fungsi dengan penamaan yang mirip dengan`UTL_FILE`. Bagian Informasi tambahan berisi basis kode untuk fungsi-fungsi ini.

Dalam kode, ganti `testaurorabucket` dengan nama bucket S3 pengujian Anda. Ganti `us-east-1` dengan Wilayah AWS tempat bucket S3 pengujian Anda berada.

Epik

Integrasikan Amazon S3 dan Aurora PostgreSQL yang kompatibel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kebijakan IAM.	Buat kebijakan AWS Identity and Access Management (IAM) yang memberikan akses ke bucket S3 dan objek di dalamnya. Untuk kode, lihat bagian Informasi tambahan.	Administrator AWS, DBA
Tambahkan peran akses Amazon S3 ke Aurora PostgreSQL.	Buat dua peran IAM: satu peran untuk dibaca dan satu peran untuk akses tulis ke Amazon S3. Lampirkan	Administrator AWS, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dua peran ke cluster yang kompatibel dengan Aurora PostgreSQL: • Satu peran untuk fitur S3Export • Satu peran untuk fitur S3Import Untuk informasi selengkapnya, lihat dokumentasi yang kompatibel dengan Aurora PostgreSQL tentang mengimpor dan mengekspor data ke Amazon S3.	

Siapkan ekstensi di Aurora PostgreSQL yang kompatibel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ekstensi aws_commons.	`aws_commons` Ekstensi adalah ketergantungan `aws_s3` ekstensi.	DBA, Pengembang
Buat ekstensi aws_s3.	`aws_s3`Ekstensi berinteraksi dengan Amazon S3.	DBA, Pengembang

Validasi integrasi yang kompatibel dengan Amazon S3 dan Aurora PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji mengimpor file dari Amazon S3 ke Aurora PostgreSQL.	Untuk menguji mengimpor file ke Aurora PostgreSQL yang kompatibel, buat	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	contoh file CSV dan unggah ke bucket S3. Buat definisi tabel berdasarkan file CSV, dan muat file ke dalam tabel dengan menggunakan `aws_s3.table_import_from_s3` fungsi.	
Uji ekspor file dari Aurora PostgreSQL ke Amazon S3.	Untuk menguji mengekspor file dari Aurora PostgreSQL yang kompatibel, buat tabel uji, isi dengan data, lalu ekspor data dengan menggunakan fungsi. `aws_s3.query_export_to_s3`	DBA, Pengembang

Untuk meniru utilitas UTL_FILE, buat fungsi pembungkus

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skema utl_file_utility.	Skema menjaga fungsi pembungkus bersama-sama. Untuk membuat skema, jalankan perintah berikut. `CREATE SCHEMA utl_file_utility;`	DBA, Pengembang
Buat tipe file_type.	Untuk membuat `file_type` tipe, gunakan kode berikut. `CREATE TYPE utl_file_utility.file_type AS (`	DBA/Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>p_path character varying(30), p_file_name character varying);</pre>	
Buat fungsi init.	`init`Fungsi menginisi alisasi variabel umum seperti `bucket` atau`region`. Untuk kode, lihat bagian Informasi tambahan.	DBA/Pengembang
Buat fungsi pembungkus.	Buat fungsi pembungku `sfopen`,`put_line`, dan`fclose`. Untuk kode, lihat bagian Informasi tambahan.	DBA, Pengembang

Uji fungsi pembungkus

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji fungsi pembungkus dalam mode tulis.	Untuk menguji fungsi pembungkus dalam mode tulis, gunakan kode yang disediakan di bagian Informasi tambahan.	DBA, Pengembang
Uji fungsi pembungkus dalam mode append.	Untuk menguji fungsi pembungkus dalam mode append, gunakan kode yang disediakan di bagian Informasi tambahan.	DBA, Pengembang

Sumber daya terkait

- Integrasi Amazon S3
- Amazon S3
- Aurora
- Amazon CloudWatch
- Amazon SNS

Informasi tambahan

Menyiapkan kebijakan IAM

Buat kebijakan berikut.

Nama kebijakan	JSON
S3 IntRead	`{ "Version": "2012-10-17", "Statement": [ { "Sid": "S3integrationtest", "Effect": "Allow", "Action": [ "s3:GetObject", "s3:ListBucket" ], "Resource": [ "arn:aws:s3:::testaurorabucket/*", "arn:aws:s3:::testaurorabucket" ] } ] }`
S3 IntWrite	`{ "Version": "2012-10-17",`

```
    "Statement": [
        {
            "Sid": "S3integrationtest
",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",

                "s3:ListBucket"
            ],
            "Resource": [
    "arn:aws:s3:::testaurorabucket/
*",                "arn:aws:s3:::test
aurorabucket"
            ]
        }
    ]
}
```

Buat fungsi init

Untuk menginisialisasi variabel umum, seperti `bucket` atau`region`, buat `init` fungsi dengan menggunakan kode berikut.

```
CREATE OR REPLACE FUNCTION utl_file_utility.init(
    )
    RETURNS void
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
BEGIN
      perform set_config
      ( format( '%s.%s','UTL_FILE_UTILITY', 'region' )
      , 'us-east-1'::text
      , false );

      perform set_config
      ( format( '%s.%s','UTL_FILE_UTILITY', 's3bucket' )
      , 'testaurorabucket'::text
      , false );
```

```
END;
$BODY$;
```

Buat fungsi pembungkus

Buat fungsi`fopen`,`put_line`, dan `fclose` pembungkus.

fopen

```
CREATE OR REPLACE FUNCTION utl_file_utility.fopen(
    p_file_name character varying,
    p_path character varying,
    p_mode character DEFAULT 'W'::bpchar,
    OUT p_file_type utl_file_utility.file_type)
    RETURNS utl_file_utility.file_type
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
declare
    v_sql character varying;
    v_cnt_stat integer;
    v_cnt integer;
    v_tabname character varying;
    v_filewithpath character varying;
    v_region character varying;
    v_bucket character varying;

BEGIN
    /*initialize common variable */
    PERFORM utl_file_utility.init();
    v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 'region' ) );
    v_bucket :=  current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 's3bucket' ) );

    /* set tabname*/
    v_tabname := substring(p_file_name,1,case when strpos(p_file_name,'.') = 0 then
 length(p_file_name) else strpos(p_file_name,'.') - 1 end );
    v_filewithpath := case when NULLif(p_path,'') is null then p_file_name else
 concat_ws('/',p_path,p_file_name) end ;
    raise notice 'v_bucket %, v_filewithpath % , v_region %', v_bucket,v_filewithpath,
 v_region;
```

```
   /* APPEND MODE HANDLING; RETURN EXISTING FILE DETAILS IF PRESENT ELSE CREATE AN
EMPTY FILE */
   IF p_mode = 'A' THEN
       v_sql := concat_ws('','create temp table if not exists ', v_tabname,' (col1
text)');
       execute v_sql;

       begin
       PERFORM aws_s3.table_import_from_s3
           ( v_tabname,
           '',
           'DELIMITER AS ''#''',
           aws_commons.create_s3_uri
           (     v_bucket,
               v_filewithpath ,
               v_region)
           );
       exception
           when others then
            raise notice 'File load issue ,%',sqlerrm;
            raise;
       end;
       execute concat_ws('','select count(*) from ',v_tabname) into v_cnt;

       IF v_cnt > 0
       then
           p_file_type.p_path := p_path;
           p_file_type.p_file_name := p_file_name;
       else
           PERFORM aws_s3.query_export_to_s3('select ''''',
                           aws_commons.create_s3_uri(v_bucket, v_filewithpath,
v_region)
                             );

           p_file_type.p_path := p_path;
           p_file_type.p_file_name := p_file_name;
       end if;
       v_sql := concat_ws('','drop table ', v_tabname);
       execute v_sql;
   ELSEIF p_mode = 'W' THEN
           PERFORM aws_s3.query_export_to_s3('select ''''',
                           aws_commons.create_s3_uri(v_bucket, v_filewithpath,
v_region)
                             );
```

```
            p_file_type.p_path := p_path;
            p_file_type.p_file_name := p_file_name;
    END IF;

EXCEPTION
        when others then
            p_file_type.p_path := p_path;
            p_file_type.p_file_name := p_file_name;
            raise notice 'fopenerror,%',sqlerrm;
            raise;
END;
$BODY$;
```

put_line

```
CREATE OR REPLACE FUNCTION utl_file_utility.put_line(
    p_file_name character varying,
    p_path character varying,
    p_line text,
    p_flag character DEFAULT 'W'::bpchar)
    RETURNS boolean
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
/*****************************************************************************
* Write line, p_line in windows format to file, p_fp - with carriage return
* added before new line.
*****************************************************************************/
declare
    v_sql varchar;
    v_ins_sql varchar;
    v_cnt INTEGER;
    v_filewithpath character varying;
    v_tabname  character varying;
    v_bucket character varying;
    v_region character varying;

BEGIN
 PERFORM utl_file_utility.init();

/* check if temp table already exist */
```

```
 v_tabname := substring(p_file_name,1,case when strpos(p_file_name,'.') = 0 then
 length(p_file_name) else strpos(p_file_name,'.') - 1 end );

 v_sql := concat_ws('','select count(1) FROM pg_catalog.pg_class c LEFT JOIN
 pg_catalog.pg_namespace n ON n.oid = c.relnamespace where n.nspname like ''pg_temp_
%'''
                        ,' AND pg_catalog.pg_table_is_visible(c.oid) AND
 Upper(relname) = Upper( '''
                        ,  v_tabname ,''' ) ');

 execute v_sql into v_cnt;

  IF v_cnt = 0 THEN
         v_sql := concat_ws('','create temp table ',v_tabname,' (col text)');
        execute v_sql;
        /* CHECK IF APPEND MODE */
        IF upper(p_flag) = 'A' THEN
            PERFORM utl_file_utility.init();
            v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY',
 'region' ) );
            v_bucket :=  current_setting( format( '%s.%s', 'UTL_FILE_UTILITY',
 's3bucket' ) );

            /* set tabname*/
            v_filewithpath := case when NULLif(p_path,'') is null then p_file_name else
 concat_ws('/',p_path,p_file_name) end ;

            begin
               PERFORM aws_s3.table_import_from_s3
                     ( v_tabname,
                          '',
                       'DELIMITER AS ''#''',
                        aws_commons.create_s3_uri
                           ( v_bucket,
                               v_filewithpath,
                               v_region    )
                    );
            exception
                when others then
                    raise notice  'Error Message : %',sqlerrm;
                    raise;
            end;
        END IF;
```

```
    END IF;
    /* INSERT INTO TEMP TABLE */
    v_ins_sql := concat_ws('','insert into ',v_tabname,' values(''',p_line,''')');
    execute v_ins_sql;
    RETURN TRUE;
    exception
            when others then
                raise notice  'Error Message : %',sqlerrm;
                raise;
END;
$BODY$;
```

fclose

```
CREATE OR REPLACE FUNCTION utl_file_utility.fclose(
    p_file_name character varying,
    p_path character varying)
    RETURNS boolean
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
DECLARE
    v_filewithpath character varying;
    v_bucket character varying;
    v_region character varying;
    v_tabname character varying;
    v_sql character varying;
BEGIN
      PERFORM utl_file_utility.init();

    v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 'region' ) );
    v_bucket :=  current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 's3bucket' ) );

    v_tabname := substring(p_file_name,1,case when strpos(p_file_name,'.') = 0 then
 length(p_file_name) else strpos(p_file_name,'.') - 1 end );
    v_filewithpath := case when NULLif(p_path,'') is null then p_file_name else
 concat_ws('/',p_path,p_file_name) end ;

    raise notice 'v_bucket %, v_filewithpath % , v_region %', v_bucket,v_filewithpath,
 v_region ;
```

```
    /* exporting to s3 */
    perform aws_s3.query_export_to_s3
        (concat_ws('','select * from ',v_tabname,'  order by ctid asc'),
            aws_commons.create_s3_uri(v_bucket, v_filewithpath, v_region)
        );
    v_sql := concat_ws('','drop table ', v_tabname);
    execute v_sql;
    RETURN TRUE;
EXCEPTION
       when others then
     raise notice 'error fclose %',sqlerrm;
     RAISE;
END;
$BODY$;
```

Uji fungsi pengaturan dan pembungkus Anda

Gunakan blok kode anonim berikut untuk menguji pengaturan Anda.

Uji mode tulis

Kode berikut menulis file bernama `s3inttest` dalam bucket S3.

```
do $$
declare
l_file_name varchar := 's3inttest' ;
l_path varchar := 'integration_test' ;
l_mode char(1) := 'W';
l_fs utl_file_utility.file_type ;
l_status boolean;

begin
select * from
utl_file_utility.fopen( l_file_name, l_path , l_mode ) into l_fs ;
raise notice 'fopen : l_fs : %', l_fs;

select * from
utl_file_utility.put_line( l_file_name, l_path ,'this is test file:in s3bucket: for
 test purpose', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from utl_file_utility.fclose( l_file_name , l_path ) into l_status ;
raise notice 'fclose : l_status %', l_status;
```

```
end;
$$
```

Uji mode append

Kode berikut menambahkan baris ke `s3inttest` file yang dibuat dalam tes sebelumnya.

```
do $$
declare
l_file_name varchar := 's3inttest' ;
l_path varchar := 'integration_test' ;
l_mode char(1) := 'A';
l_fs utl_file_utility.file_type ;
l_status boolean;

begin
select * from
utl_file_utility.fopen( l_file_name, l_path , l_mode ) into l_fs ;
raise notice 'fopen : l_fs : %', l_fs;


select * from
utl_file_utility.put_line( l_file_name, l_path ,'this is test file:in s3bucket: for
 test purpose : append 1', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from
utl_file_utility.put_line( l_file_name, l_path ,'this is test file:in s3bucket : for
 test purpose : append 2', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from utl_file_utility.fclose( l_file_name , l_path ) into l_status ;
raise notice 'fclose : l_status %', l_status;

end;
$$
```

Pemberitahuan Amazon SNS

Secara opsional, Anda dapat mengatur CloudWatch pemantauan Amazon dan notifikasi Amazon SNS di bucket S3. Untuk informasi selengkapnya, lihat Memantau Amazon S3 dan Menyiapkan Pemberitahuan Amazon SNS.

Validasi objek database setelah bermigrasi dari Oracle ke Amazon Aurora PostgreSQL

Venkatramana Chintha dan Eduardo Valentim, Amazon Web Services

Ringkasan

Pola ini menjelaskan step-by-step pendekatan untuk memvalidasi objek setelah memigrasikan database Oracle ke Amazon Aurora PostgreSQL Edisi yang kompatibel.

Pola ini menguraikan skenario penggunaan dan langkah-langkah untuk validasi objek database; untuk informasi lebih rinci, lihat Memvalidasi objek database setelah migrasi menggunakan AWS SCT dan AWS DMS di blog AWS Database.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Database Oracle lokal yang dimigrasikan ke database yang kompatibel dengan Aurora PostgreSQL.
- Kredensi login yang menerapkan RDSData FullAccess kebijakan Amazon, untuk database yang kompatibel dengan Aurora PostgreSQL.
- Pola ini menggunakan editor kueri untuk cluster DB Aurora Serverless, yang tersedia di konsol Amazon Relational Database Service (Amazon RDS). Namun, Anda dapat menggunakan pola ini dengan editor kueri lainnya.

Batasan

- Objek Oracle SYNONYM tidak tersedia di PostgreSQL tetapi sebagian dapat divalidasi melalui tampilan atau SET search_path query.
- Editor kueri Amazon RDS hanya tersedia di Wilayah AWS tertentu dan untuk versi MySQL dan PostgreSQL tertentu.

Arsitektur

Alat

Alat

- Amazon Aurora PostgreSQL Compatible Edition — Aurora PostgreSQL kompatibel adalah mesin database relasional relasional yang dikelola sepenuhnya, kompatibel dengan PostgreSQL, dan sesuai dengan Asam yang menggabungkan kecepatan dan keandalan database komersial kelas atas dengan kesederhanaan dan efektivitas biaya database sumber terbuka.
- Amazon RDS — Amazon Relational Database Service (Amazon RDS) memudahkan penyiapan, pengoperasian, dan skala database relasional di AWS Cloud. Layanan ini menyediakan kapasitas yang hemat biaya dan dapat diubah ukurannya untuk basis data relasional standar industri serta mengelola tugas administrasi basis data umum.
- Editor Kueri untuk Aurora Severless - Editor kueri membantu Anda menjalankan kueri SQL di konsol Amazon RDS. Anda dapat menjalankan pernyataan SQL yang valid pada cluster DB Aurora Serverless, termasuk manipulasi data dan pernyataan definisi data.

Untuk memvalidasi objek, gunakan skrip lengkap di file “Skrip validasi objek” di bagian “Lampiran”. Gunakan tabel berikut untuk referensi.

Objek Oracle	Script untuk digunakan
Paket	Permintaan 1
Tabel	Permintaan 3
Tampilan	Permintaan 5
Urutan	Permintaan 7
Pemicu	Permintaan 9
Kunci primer	Permintaan 11
Indeks	Permintaan 13
Batasan pemeriksaan	Permintaan 15
Kunci asing	Permintaan 17

PostgreSQL objek	Script untuk digunakan
Paket	Permintaan 2
Tabel	Permintaan 4
Tampilan	Permintaan 6
Urutan	Permintaan 8
Pemicu	Permintaan 10
Kunci primer	Permintaan 12
Indeks	Permintaan 14
Batasan pemeriksaan	Permintaan 16
Kunci asing	Permintaan 18

Epik

Validasi objek dalam database sumber Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kueri validasi “paket” di database sumber Oracle.	Unduh dan buka file “Skrip validasi objek” dari bagian “Lampiran”. Connect ke database Oracle sumber melalui program klien Anda. Jalankan skrip validasi “Query 1" dari file “Object validation scripts”. Penting: Masukkan nama pengguna Oracle Anda alih-alih “your_schema” dalam kueri. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kueri validasi “tabel”.	Jalankan skrip “Query 3" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “tampilan”.	Jalankan skrip “Query 5" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan validasi hitungan “urutan”.	Jalankan skrip “Query 7" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “pemicu”.	Jalankan skrip “Query 9" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “kunci utama”.	Jalankan skrip “Query 11" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “indeks”.	Jalankan skrip validasi “Query 13" dari file “Object validatio n scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “periksa kendala”.	Jalankan skrip “Query 15" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kueri validasi “kunci asing”.	Jalankan skrip validasi “Query 17" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA

Validasi objek dalam database target yang kompatibel dengan Aurora PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke database target yang kompatibel dengan Aurora PostgreSQL dengan menggunakan editor kueri.	Masuk ke AWS Management Console dan buka konsol Amazon RDS. Di sudut kanan atas, pilih Wilayah AWS tempat Anda membuat database yang kompatibel dengan Aurora PostgreSQL. Di panel navigasi, pilih “Databases,” dan pilih target Aurora PostgreSQL database yang kompatibel. Di “Tindakan,” pilih “Kueri.” Penting: Jika Anda belum terhubung ke database sebelumnya, halaman “Connect to database” akan terbuka. Anda kemudian perlu memasukkan informasi database Anda, seperti nama pengguna dan kata sandi.	Pengembang, DBA
Jalankan kueri validasi “paket”.	Jalankan skrip “Kueri 2" dari file “Skrip validasi objek” di bagian “Lampiran”. Pastikan	Pengembang, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda mencatat hasil kueri Anda.	
Jalankan kueri validasi “tabel”.	Kembali ke editor kueri untuk database yang kompatibel dengan Aurora PostgreSQL, dan jalankan skrip “Query 4" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “tampilan”.	Kembali ke editor kueri untuk database yang kompatibel dengan Aurora PostgreSQL, dan jalankan skrip “Query 6" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan validasi hitungan “urutan”.	Kembali ke editor kueri untuk database yang kompatibel dengan Aurora PostgreSQL, dan jalankan skrip “Query 8" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “pemicu”.	Kembali ke editor kueri untuk database yang kompatibel dengan Aurora PostgreSQL, dan jalankan skrip “Query 10" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kueri validasi “kunci utama”.	Kembali ke editor kueri untuk database yang kompatibel dengan Aurora PostgreSQL, dan jalankan skrip “Query 12" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “indeks”.	Kembali ke editor kueri untuk database yang kompatibel dengan Aurora PostgreSQL, dan jalankan skrip “Query 14" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “periksa kendala”.	Jalankan skrip “Query 16" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA
Jalankan kueri validasi “kunci asing”.	Jalankan skrip validasi “Query 18" dari file “Object validation scripts”. Pastikan Anda mencatat hasil kueri Anda.	Pengembang, DBA

Bandingkan catatan validasi basis data sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bandingkan dan validasi kedua hasil kueri.	Bandingkan hasil kueri database yang kompatibel dengan Oracle dan Aurora PostgreSQL untuk memvalidasi semua objek.	Pengembang, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jika semuanya cocok, maka semua objek telah berhasil divalidasi.	

Sumber daya terkait

- Memvalidasi objek database setelah migrasi menggunakan AWS SCT dan AWS DMS
- Fitur Amazon Aurora: Edisi yang kompatibel dengan PostgreSQL

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Rehost

Topik

- Mempercepat penemuan dan migrasi beban kerja Microsoft ke AWS
- Membuat proses persetujuan untuk permintaan firewall selama migrasi rehost ke AWS
- Menyerap dan memigrasikan instans EC2 Windows ke akun AWS Managed Services
- Memigrasikan database Couchbase Server ke Amazon EC2
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan menggunakan pengiriman log untuk mengurangi waktu pemadaman
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan pemulihan bencana ketersediaan tinggi
- Migrasi aplikasi yang dihosting IIS ke Amazon dengan menggunakan appcmd.exe EC2
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2 menggunakan Layanan Migrasi Aplikasi
- Migrasikan beban kerja F5 BIG-IP ke F5 BIG-IP VE di AWS Cloud
- Memigrasikan aplikasi web Go lokal ke AWS Elastic Beanstalk menggunakan metode biner
- Memigrasi server SFTP lokal untuk menggunakan AWS AWS Transfer for SFTP
- Memigrasikan VM lokal ke Amazon dengan EC2 menggunakan AWS Application Migration Service
- Migrasikan kumpulan data kecil dari lokasi ke Amazon S3 menggunakan AWS SFTP
- Memigrasi database Oracle lokal ke Oracle di Amazon EC2
- Memigrasikan database Oracle lokal ke Amazon menggunakan Oracle Data EC2 Pump
- Migrasikan sistem RHEL BYOL ke instans AWS License-Included dengan menggunakan AWS MGN
- Memigrasikan database SAP ASE lokal ke Amazon EC2
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2
- Menghosting ulang beban kerja lokal di AWS Cloud: daftar periksa migrasi
- Menyiapkan infrastruktur Multi-AZ untuk SQL Server Always On FCI dengan menggunakan Amazon FSx
- Gunakan kueri BMC Discovery untuk mengekstrak data migrasi untuk perencanaan migrasi

Mempercepat penemuan dan migrasi beban kerja Microsoft ke AWS

Ali Alzand, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menggunakan PowerShell modul Migration Validator Toolkit untuk menemukan dan memigrasikan beban kerja Microsoft Anda ke AWS. Modul ini bekerja dengan melakukan beberapa pemeriksaan dan validasi untuk tugas umum yang terkait dengan beban kerja Microsoft apa pun. Misalnya, modul memeriksa instance yang mungkin memiliki beberapa disk yang terpasang padanya atau instance yang menggunakan banyak alamat IP. Untuk daftar lengkap pemeriksaan yang dapat dilakukan modul, lihat bagian Cek di GitHub halaman modul.

PowerShell Modul Migration Validator Toolkit dapat membantu organisasi Anda mengurangi waktu dan upaya yang terlibat dalam menemukan aplikasi dan layanan apa yang berjalan pada beban kerja Microsoft Anda. Modul ini juga dapat membantu Anda mengidentifikasi konfigurasi beban kerja Anda sehingga Anda dapat mengetahui apakah konfigurasi Anda didukung di AWS. Modul ini juga memberikan rekomendasi untuk langkah selanjutnya dan tindakan mitigasi, sehingga Anda dapat menghindari kesalahan konfigurasi sebelum, selama, atau setelah migrasi.

Prasyarat dan batasan

Prasyarat

- Akun administrator lokal
- PowerShell 4.0

Batasan

- Hanya berfungsi di Microsoft Windows Server 2012 R2 atau yang lebih baru

Alat

Alat

- PowerShell 4.0

Repositori kode

PowerShell [Modul Migration Validator Toolkit untuk pola ini tersedia di repositori -microsoft-workloads. GitHub migration-validator-toolkit-for]

Epik

Jalankan PowerShell modul Migration Validator Toolkit pada satu target

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Unduh, ekstrak, impor, dan panggil modul.	Pilih salah satu metode berikut untuk mengunduh dan menyebarkan modul:
	• Jalankan PowerShell skrip
	• Unduh dan ekstrak file.zip
	• Kloning repositori GitHub
	
	Jalankan PowerShell skrip
	
	Dalam PowerShell, jalankan kode contoh berikut:
	
	`#MigrationValidatorToolkit`
	`$uri = 'https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads/archive/refs/heads/main.zip'`
	`$destination = (Get-Location).Path`
	`if ((Test-Path -Path "$destination\MigrationValidatorToolkit.zip" -PathType Leaf) -or (Test-Path -Path "$destination\MigrationValidatorToolkit")) {`
		Administrator Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> write-host "File
	 $destination\Migra
	tionValidatorToolk
	it.zip or folder
	 $destination\Migra
	tionValidatorToolkit
	 found, exiting"
	}else {
	    Write-host "Enable
	 TLS 1.2 for this
	 PowerShell session
	 only."
	    [Net.ServicePointM
	anager]::SecurityP
	rotocol = [Net.Secu
	rityProtocolType]:
	:Tls12
	    $webClient =
	 New-Object System.Ne
	t.WebClient
	    Write-host
	 "Downloading Migration
	ValidatorToolkit.zip"
	    $webClient.Downloa
	dFile($uri, "$destina
	tion\MigrationVali
	datorToolkit.zip")
	    Write-host
	 "MigrationValidato
	rToolkit.zip download
	 successfully"
	    Add-Type -Assembly
	 "system.io.compres
	sion.filesystem"
	    [System.IO.Compres
	sion.ZipFile]::Ext
	ractToDirectory("$
	destination\Migrat
	ionValidatorToolki
	t.zip","$destinati
	on\MigrationValida
	torToolkit")</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>Write-host
	"Extracting Migration
	ValidatorToolkit.zip
	complete successfully"
	Import-Module
	"$destination\Migr
	ationValidatorToolkit
	\migration-validator-
	toolkit-for-microsoft
	-workloads-main\Mi
	grationValidatorTo
	olkit.psm1"; Invoke-
	MigrationValidatorTo
	olkit
	}</pre>
	Kode mengunduh modul dari file.zip. Kemudian, kode mengekstrak, mengimpor, dan memanggil modul.
	Unduh dan ekstrak file.zip
	1. Unduh file.zip (unduh).
	2. Ekstrak file.zip.
	3. Ikuti langkah-langkah dalam Invoke the module manual story dari panduan ini.
	Kloning repositori GitHub
	1. Untuk mengkloning repositori GitHub migration-validator-toolkit-for-microsoft-workloads, jalankan perintah Git berikut di jendela terminal:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads.git` 2. Ikuti langkah-langkah dalam Invoke the module manual story dari panduan ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Panggil modul secara manual.	1. Buka direktori tempat modul yang diunduh disimpan.
	2. Untuk menghasilkan output pilihan Anda, jalankan salah satu perintah berikut sebagai administrator di PowerShell:
	
	Format Format-Tabel:
	`Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit`
	
	Format Format-Daftar:
	`Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit -List`
	
	GridViewFormat keluar:
	`Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit -GridView`
	
	ConvertTo-Csv format:
	`Import-Module .\MigrationValidatorToolkit.psm1;Invoke-Migra`
		Administrator Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`tionValidatorToolkit -csv`	

Jalankan PowerShell modul Migration Validator Toolkit pada beberapa target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh file.zip atau kloning repositori. GitHub	Pilih salah satu opsi berikut: • Unduh file zip. (unduh). • Untuk mengkloning repositori GitHub migration-validator-toolkit-for-microsoft-workloads, jalankan perintah Git berikut di jendela terminal: `git clone https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads.git`	Administrator Sistem
Perbarui daftar server.csv.	Jika Anda mengunduh file.zip, ikuti langkah-langkah berikut: 1. Ekstrak file.zip. 2. Pergi ke `Migration ValidatorToolkit\Inputs\` direktori. 3. Perbarui `serverlist.csv` dengan nama host komputer target Anda.	Administrator Sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memanggil modul.	Anda dapat menggunakan komputer apa pun dalam domain yang menggunakan pengguna domain yang memiliki akses administrator ke komputer target. 1. Unduh kode sumber sebagai file.zip dan ekstrak file. 2. Sebagai administrator di PowerShell, jalankan perintah berikut: `Import-Module .\MigrationValidatorToolkit.psm1;Invoke-DomainComputers` File output.csv disimpan `MigrationValidatorToolkit\Outputs\folder` dengan nama awalan. `DomainComputers_MigrationAutomations_YYYY-MM-DDTHH-MM-SS`	Administrator Sistem

Pemecahan masalah

Isu	Solusi
`MigrationValidatorToolkit` menulis informasi tentang eksekusi, perintah, dan	Anda dapat melihat file log secara manual di lokasi berikut:

Isu	Solusi
kesalahan untuk mencatat file pada host yang sedang berjalan.	1. Pergi ke `MigrationValidator Toolkit\logs\` direktori. 2. Temukan file log. Format nama file log adalah: `ComputerName_Migra tionValidatorToolkit_YYYY-MM-SSTHH-MM-SS.log`

Sumber daya terkait

- Opsi, alat, dan praktik terbaik untuk memigrasikan beban kerja Microsoft ke AWS (AWS Prescriptive Guidance)
- Pola migrasi Microsoft (AWS Prescriptive Guidance)
- Layanan Migrasi Cloud Gratis di AWS (dokumentasi AWS)
- Tindakan pasca-peluncuran yang telah ditentukan sebelumnya (Dokumentasi pemasaran aplikasi)

Informasi tambahan

Pertanyaan yang sering diajukan

Di mana saya dapat menjalankan modul Migration Validator Toolkit? PowerShell

Anda dapat menjalankan modul di Microsoft Windows Server 2012 R2 atau yang lebih baru.

Kapan saya menjalankan modul ini?

Kami menyarankan Anda menjalankan modul selama fase penilaian perjalanan migrasi.

Apakah modul memodifikasi server saya yang ada?

Tidak. Semua tindakan dalam modul ini hanya-baca.

Berapa lama waktu yang dibutuhkan untuk menjalankan modul?

Biasanya dibutuhkan 1-5 menit untuk menjalankan modul, tetapi itu tergantung pada alokasi sumber daya server Anda.

Izin apa yang dibutuhkan modul untuk dijalankan?

Anda harus menjalankan modul dari akun administrator lokal.

Bisakah saya menjalankan modul di server fisik?

Ya, selama sistem operasinya adalah Microsoft Windows Server 2012 R2 atau yang lebih baru.

Bagaimana cara menjalankan modul dalam skala besar untuk beberapa server?

Untuk menjalankan modul pada beberapa komputer yang bergabung dengan domain dalam skala besar, ikuti langkah-langkah dalam PowerShell modul Jalankan Migration Validator Toolkit pada beberapa target epik panduan ini. Untuk komputer yang tidak bergabung dengan domain, gunakan pemanggilan jarak jauh atau jalankan modul secara lokal dengan mengikuti langkah-langkah dalam modul Run the Migration Validator Toolkit PowerShell pada satu target epik panduan ini.

Membuat proses persetujuan untuk permintaan firewall selama migrasi rehost ke AWS

Srikanth Rangavajhala, Amazon Web Services

Ringkasan

Jika Anda ingin menggunakan AWS Application Migration Serviceatau Cloud Migration Factory AWS aktif untuk migrasi rehost ke AWS Cloud, salah satu prasyaratnya adalah Anda harus membiarkan port TCP 443 dan 1500 tetap terbuka. Biasanya, membuka port firewall ini memerlukan persetujuan dari tim keamanan informasi (InfoSec) Anda.

Pola ini menguraikan proses untuk mendapatkan persetujuan permintaan firewall dari InfoSec tim selama migrasi rehost ke. AWS Cloud Anda dapat menggunakan proses ini untuk menghindari penolakan permintaan firewall Anda oleh InfoSec tim, yang dapat menjadi mahal dan memakan waktu. Proses permintaan firewall memiliki dua langkah peninjauan dan persetujuan antara konsultan AWS migrasi dan prospek yang bekerja dengan tim Anda InfoSec dan aplikasi untuk membuka port firewall.

Pola ini mengasumsikan bahwa Anda merencanakan migrasi rehost dengan AWS konsultan atau spesialis migrasi dari organisasi Anda. Anda dapat menggunakan pola ini jika organisasi Anda tidak memiliki proses persetujuan firewall atau formulir persetujuan menyeluruh permintaan firewall. Untuk informasi lebih lanjut tentang ini, lihat bagian Batasan dari pola ini. Untuk informasi selengkapnya tentang persyaratan jaringan untuk Layanan Migrasi Aplikasi, lihat Persyaratan jaringan dalam dokumentasi Layanan Migrasi Aplikasi.

Prasyarat dan batasan

Prasyarat

- Migrasi rehost yang direncanakan dengan AWS konsultan atau spesialis migrasi dari organisasi Anda
- Port dan informasi IP yang diperlukan untuk memigrasikan tumpukan
- Diagram arsitektur negara yang ada dan masa depan
- Informasi firewall tentang infrastruktur lokal dan tujuan, port, dan arus zone-to-zone lalu lintas
- Daftar periksa tinjauan permintaan firewall (terlampir)
- Dokumen permintaan firewall, dikonfigurasi sesuai dengan kebutuhan organisasi Anda
- Daftar kontak untuk pengulas dan pemberi persetujuan firewall, termasuk peran berikut:
  - Pengirim permintaan firewall — spesialis AWS migrasi atau konsultan. Pengirim permintaan firewall juga dapat menjadi spesialis migrasi dari organisasi Anda.

- Peninjau permintaan firewall — Biasanya, ini adalah titik kontak tunggal (SPOC) dari. AWS
- Firewall request approver — Anggota InfoSec tim.

Batasan

- Pola ini menjelaskan proses persetujuan permintaan firewall generik. Persyaratan dapat bervariasi untuk organisasi individu.
- Pastikan Anda melacak perubahan pada dokumen permintaan firewall Anda.

Tabel berikut menunjukkan kasus penggunaan untuk pola ini.

Apakah organisasi Anda memiliki proses persetujuan firewall yang sudah ada?	Apakah organisasi Anda memiliki formulir permintaan firewall yang ada?	Tindakan yang disarankan
Ya	Ya	Berkolaborasi dengan AWS konsultan atau spesialis migrasi Anda untuk mengimple mentasikan proses organisasi Anda.
Tidak	Ya	Gunakan proses persetuju an firewall pola ini. Gunakan AWS konsultan atau spesialis migrasi dari organisasi Anda untuk mengirimkan formulir persetujuan menyeluruh permintaan firewall.
Tidak	Tidak	Gunakan proses persetuju an firewall pola ini. Gunakan AWS konsultan atau spesialis migrasi dari organisasi Anda untuk mengirimkan formulir persetujuan menyeluruh permintaan firewall.

Arsitektur

Diagram berikut menunjukkan langkah-langkah untuk proses persetujuan permintaan firewall.

Alat

Anda dapat menggunakan alat pemindai seperti Palo Alto Networks atau SolarWindsuntuk menganalisis dan memvalidasi firewall dan alamat IP.

Epik

Analisis permintaan firewall

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Analisis port dan alamat IP.	Pengirim permintaan firewall menyelesaikan analisis awal untuk memahami port firewall dan alamat IP yang diperluka n. Setelah ini selesai, mereka meminta InfoSec tim Anda membuka port yang diperluka n dan memetakan alamat IP.	Insinyur AWS Cloud, spesialis migrasi

Validasi permintaan firewall

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi informasi firewall.	AWS Cloud Insinyur menjadwalkan pertemuan dengan InfoSec tim Anda. Selama pertemuan ini, insinyur memeriksa dan memvalidasi informasi permintaan firewall. Biasanya, pengirim permintaa n firewall adalah orang yang	Insinyur AWS Cloud, spesialis migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sama dengan pemohon firewall. Fase validasi ini dapat menjadi berulang berdasark an umpan balik yang diberikan oleh pemberi persetujuan jika ada yang diamati atau direkomendasikan.	
Perbarui dokumen permintaan firewall.	Setelah InfoSec tim membagikan umpan balik mereka, dokumen permintaa n firewall diedit, disimpan, dan diunggah ulang. Dokumen ini diperbarui setelah setiap iterasi. Kami menyarankan Anda menyimpan dokumen ini dalam folder penyimpanan yang dikontrol versi. Ini berarti bahwa semua perubahan dilacak dan diterapkan dengan benar.	Insinyur AWS Cloud, spesialis migrasi

Kirim permintaan firewall

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim permintaan firewall.	Setelah pemberi persetujuan permintaan firewall menyetuju i permintaan persetuju an selimut firewall, AWS Cloud insinyur mengirimkan permintaan firewall. Permintaa n menentukan port yang harus	Insinyur AWS Cloud, spesialis migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	terbuka dan alamat IP yang diperlukan untuk memetakan dan memperbarui. Akun AWS Anda dapat memberikan saran atau memberikan umpan balik setelah permintaan firewall dikirimkan. Kami menyarankan Anda mengotomatiskan proses umpan balik ini dan mengirim pengeditan apa pun melalui mekanisme alur kerja yang ditentukan.	

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menyerap dan memigrasikan instans EC2 Windows ke akun AWS Managed Services

Anil Kunapareddy dan Venkatramana Chintha, Amazon Web Services

Ringkasan

Pola ini menjelaskan step-by-step proses migrasi dan pengambilan instans Amazon Elastic Compute Cloud (Amazon EC2) Windows ke dalam akun Managed Services (AMS) Amazon Web Services (AWS). AMS dapat membantu Anda mengelola instans dengan lebih efisien dan aman. AMS memberikan fleksibilitas operasional, meningkatkan keamanan dan kepatuhan, serta membantu Anda mengoptimalkan kapasitas dan mengurangi biaya.

Pola ini dimulai dengan instance EC2 Windows yang telah Anda migrasi ke subnet pementasan di akun AMS Anda. Berbagai layanan dan alat migrasi tersedia untuk melakukan tugas ini, seperti AWS Application Migration Service.

Untuk membuat perubahan pada lingkungan yang dikelola AMS, Anda membuat dan mengirimkan permintaan perubahan (RFC) untuk operasi atau tindakan tertentu. Menggunakan AMS workload ingest (WIGS) RFC, Anda memasukkan instance ke dalam akun AMS dan membuat Amazon Machine Image (AMI) kustom. Anda kemudian membuat EC2 instance yang dikelola AMS dengan mengirimkan RFC lain untuk membuat tumpukan. EC2 Untuk informasi selengkapnya, lihat AMS Workload Ingest dalam dokumentasi AMS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif dan dikelola AMS
- Landing zone yang sudah ada
- Izin untuk membuat perubahan di VPC yang dikelola AMS
- Instans Amazon EC2 Windows di subnet pementasan di akun AMS Anda
- Penyelesaian prasyarat umum untuk memigrasikan beban kerja menggunakan AMS WIGS
- Penyelesaian prasyarat Windows untuk memigrasikan beban kerja menggunakan AMS WIGS

Batasan

- Pola ini untuk EC2 contoh yang mengoperasikan Windows Server. Pola ini tidak berlaku untuk instance yang menjalankan sistem operasi lain, seperti Linux.

Arsitektur

Tumpukan teknologi sumber

Instans Amazon EC2 Windows dalam subnet pementasan di akun AMS Anda

Tumpukan teknologi target

Instans Amazon EC2 Windows yang dikelola oleh AWS Managed Services (AMS)

Arsitektur target

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, dan Anda dapat meningkatkan skala atau menskalakan.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Managed Services (AMS) membantu Anda beroperasi lebih efisien dan aman dengan menyediakan pengelolaan infrastruktur AWS Anda yang berkelanjutan, termasuk pemantauan, manajemen insiden, panduan keamanan, dukungan patch, dan pencadangan untuk beban kerja AWS.

Layanan lainnya

- PowerShelladalah program manajemen otomatisasi dan konfigurasi Microsoft yang berjalan di Windows, Linux, dan macOS.

Epik

Konfigurasikan pengaturan pada instance

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah pengaturan Klien DNS.	1. Pada EC2 contoh sumber, buka Command Prompt sebagai administrator, ketik`gpedit.msc` , lalu tekan Enter. 2. Di Editor Kebijakan Grup Lokal, arahkan ke Konfigurasi Komputer, Template Administratif, Jaringan, Klien DNS. 3. Untuk akhiran DNS Primer, pilih Tidak dikonfigurasi. 4. Untuk devolusi akhiran DNS Primer, pilih Tidak dikonfigurasi.	Insinyur migrasi
Ubah pengaturan Pembaruan Windows.	1. Di Editor Kebijakan Grup Lokal, arahkan ke Konfigurasi Komputer, Template Administratif, Komponen Windows, Pembaruan Windows. 2. Untuk Tentukan lokasi layanan pembaruan Microsoft intranet, pilih Tidak dikonfigurasi. 3. Untuk Mengkonfigurasi Pembaruan Otomatis, pilih Tidak dikonfigurasi.	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk frekuensi deteksi Pembaruan Otomatis, pilih Tidak dikonfigurasi. 5. Tutup Editor Kebijakan Grup Lokal.	
Aktifkan firewall.	1. Pada EC2 contoh sumber, buka Command Prompt sebagai administrator, ketik`services.msc` , lalu tekan Enter. 2. Di Layanan Windows, aktifkan Firewall. 3. Tutup Layanan Windows.	Insinyur migrasi

Siapkan instance untuk AMS WIGS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan dan siapkan instance.	1. Menggunakan host bastion dan kredensil lokal, buat koneksi Remote Desktop Protocol (RDP) ke EC2 instance di subnet pementasan. 2. Hapus semua perangkat lunak lama, perangkat lunak antivirus, dan solusi cadangan yang tidak diperlukan di AMS.	Insinyur migrasi
Perbaiki file sppnp.dll.	1. Kunjungi `C:\Windows\System32\sppnp.dll` .	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Ubah nama `sppnp.dll` menjadi `sppnp_old.dll` .
	3. Menggunakan PowerShell dan kredensi administrator, masukkan perintah berikut:
	`dism /online /cleanup-image /restorehealth`
	`sfc /scannnow`
	4. Mulai ulang instance EC2 Windows.	

Jalankan skrip validasi Pra-wig.	1. Unduh file zip Validasi Pra-konsumsi Windows WIGS (`windows-prewings-valication.zip`) dari Migrasi beban kerja: Validasi pra-konsumsi Windows dalam dokumentasi AMS. 2. Jalankan skrip validasi Windows Pra-wig dan verifikasi hasilnya. 3. Jika validasi gagal, perbaiki masalah, dan jalankan kembali skrip validasi hingga validasi berhasil.	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AMI failsafe.	Setelah validasi Pra-wig berlalu, buat AMI pra-konsu msi sebagai berikut: 1. Pilih Deployment, Advanced stack components, AMI, Create. 2. Selama pembuatan, tambahkan tag`Key=Name, Value=APPLICATION-ID_IngestReady` . 3. Tunggu hingga AMI dibuat sebelum melanjutkan. Untuk informasi selengkapnya, lihat AMI | Buat di dokumenta si AMS.	Insinyur migrasi

Menelan dan memvalidasi instance

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim RFC untuk membuat tumpukan konsumsi beban kerja.	Kirim permintaan perubahan (RFC) untuk memulai AMS WIGS. Untuk petunjuk, lihat Workload Ingest Stack: Membuat dalam dokumenta si AMS. Ini memulai konsumsi beban kerja dan menginsta l semua perangkat lunak yang diperlukan oleh AMS, termasuk alat cadangan, perangkat lunak EC2	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	manajemen Amazon, dan perangkat lunak antivirus.	
Validasi migrasi yang berhasil.	Setelah konsumsi beban kerja selesai, Anda dapat melihat instans yang dikelola AMS dan AMI yang dicerna AMS. 1. Masuk ke instans yang dikelola AMS dengan kredensial domain. 2. Validasi domain yang bergabung sebagai berikut: a. Di Windows Explorer, klik kanan PC ini, lalu pilih Properties. b. Di bagian Spesifikasi Perangkat, konfirmasikan bahwa domain muncul di Nama perangkat lengkap. 3. Validasi sumber dan target disk drive.	Insinyur migrasi

Luncurkan instance di akun AMS target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim RFC untuk membuat EC2 tumpukan.	1. Menggunakan AMI yang dicerna AMS dari instance Windows, siapkan RFC untuk tumpukan sesuai EC2 dengan instruksi di EC2 Buat instance tumpukan	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dalam dokumentasi AMS. Dalam EC2 stack RFC, sediakan semua parameter , termasuk nama server, tag, VPC target, subnet target, tipe instance, grup keamanan target, konsumsi AMI, dan peran. 2. Kirim RFC untuk EC2 tumpukan, dan kemudian tunggu instance berhasil dibuat.	

Sumber daya terkait

AWS Prescriptive Guidance

- Mengotomatiskan aktivitas konsumsi pra-beban kerja untuk AWS Managed Services di Windows
- Secara otomatis membuat RFC di AMS menggunakan Python

Dokumentasi AMS

- Pencerapan Beban Kerja AMS
- Bagaimana Migrasi Mengubah Sumber Daya Anda
- Migrasi Beban Kerja: Proses Standar

Sumber daya pemasaran

- AWS Managed Services
- AWS Managed Services FAQs
- Sumber Daya AWS Managed Services
- Fitur AWS Managed Services

Memigrasikan database Couchbase Server ke Amazon EC2

Subhani Shaik, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan Server Couchbase dari lingkungan lokal ke Amazon Elastic Compute Cloud (Amazon). EC2 AWS

Couchbase Server adalah database NoSQL (dokumen JSON) terdistribusi yang menyediakan kemampuan database relasional. Migrasi database Couchbase Server untuk AWS dapat memberikan peningkatan skalabilitas, peningkatan kinerja, efisiensi biaya, keamanan yang ditingkatkan, manajemen yang disederhanakan, dan jangkauan global, yang dapat menguntungkan aplikasi yang memerlukan ketersediaan tinggi dan akses data latensi rendah. Anda juga mendapatkan akses ke fitur-fitur canggih melalui layanan AWS terkelola.

Couchbase Server on AWS menyediakan fitur utama berikut:

- Arsitektur memori-pertama
- Ketersediaan tinggi, pemulihan bencana, dan penyeimbangan beban
- Multi-master, penyebaran multi-wilayah untuk kinerja optimal

Untuk informasi selengkapnya tentang manfaat utama, lihat bagian Informasi tambahan dan situs web Couchbase.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan virtual private cloud (VPC), dua Availability Zone, subnet pribadi, dan grup keamanan. Untuk petunjuknya, lihat Membuat VPC di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).
- Konektivitas diaktifkan antara sumber dan lingkungan target. Untuk informasi tentang port TCX yang digunakan oleh Couchbase Server, lihat dokumentasi Couchbase.

Arsitektur

Diagram berikut menunjukkan arsitektur tingkat tinggi untuk memigrasi Server Couchbase ke. AWS

Dari kluster Couchbase lokal, data bergerak melalui gateway pelanggan dengan menggunakan. AWS Direct Connect Data melewati router dan Direct Connect rute dan mencapai VPC melalui gateway AWS Virtual Private Network (Site-to-Site VPN). VPC berisi EC2 instance yang menjalankan Couchbase Server. AWS Infrastruktur juga mencakup AWS Identity and Access Management (IAM) untuk kontrol akses, AWS Key Management Service (AWS KMS) untuk enkripsi data, Amazon Elastic Block Store (Amazon EBS) untuk penyimpanan blok, dan Amazon Simple Storage Service (Amazon S3) untuk penyimpanan data.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Direct Connectmenghubungkan jaringan internal Anda ke Direct Connect lokasi melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke publik Layanan AWS sambil melewati penyedia layanan internet di jalur jaringan Anda.

Praktik terbaik

- Menginstal dan mengonfigurasi Couchbase pada platform operasi yang berbeda
- Praktik terbaik untuk menerapkan Server Couchbase di AWS
- Membuat cluster Couchbase
- Praktik terbaik kinerja untuk aplikasi Couchbase
- Praktik terbaik keamanan untuk Couchbase Server
- Praktik terbaik penyimpanan untuk database Couchbase Server

Epik

Menerapkan EC2 instans Amazon untuk Couchbase Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka EC2 konsol Amazon.	Masuk ke Konsol Manajemen AWSdan buka EC2 konsol Amazon.	DevOps insinyur, administrator Couchbase
Menerapkan EC2 instans Amazon.	Luncurkan EC2 instance yang cocok dengan konfigura si Server Couchbase lokal. Untuk informasi selengkapnya tentang cara menerapkan EC2 instance, lihat Meluncurkan EC2 instans Amazon di EC2 dokumentasi Amazon.	DevOps insinyur, administrator Couchbase

Instal dan konfigurasikan Server Couchbase di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal cluster Couchbase.	Tinjau pedoman penerapan Couchbase Server sebelum Anda menginstal Couchbase Server di Amazon. EC2 Untuk menginstal Couchbase Server, lihat dokumentasi Couchbase Server	Administrator Couchbase
Konfigurasikan cluster.	Untuk mengonfigurasi cluster, lihat Opsi Konfigura si Cluster dalam dokumentasi Couchbase.	Administrator Couchbase

Tambahkan node baru dan seimbangkan kembali cluster Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan node untuk EC2 contoh.	Tambahkan EC2 instance yang baru diterapkan yang menginstal Couchbase ke kluster lokal yang ada. Untuk petunjuk, lihat Menambahkan Node dan Menyeimbangkan Kembali dalam dokumentasi Server Couchbase.	Administrator Couchbase
Menyeimbangkan kembali cluster.	Proses rebalancing membuat node yang baru ditambahkan dengan EC2 instance menjadi anggota aktif dari cluster Couchbase. Untuk petunjuk, lihat Menambahkan Node dan Menyeimbangkan Kembali dalam dokumentasi Server Couchbase	Administrator Couchbase

Konfigurasi ulang koneksi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus node lokal dan menyeimbangkan kembali.	Sekarang Anda dapat menghapus node lokal dari cluster. Setelah menghapus node, ikuti proses penyeimbangan ulang untuk mendistribusikan kembali data, indeks, pemrosesan peristiwa, dan pemrosesan kueri di antara node yang tersedia	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di cluster. Untuk petunjuk, lihat Menghapus Node dan Menyeimbangkan Kembali dalam dokumentasi Server Couchbase.	
Perbarui parameter koneksi.	Perbarui parameter koneksi aplikasi Anda untuk menggunakan alamat EC2 IP Amazon yang baru, sehingga aplikasi Anda dapat terhubung ke node baru.	Pengembang aplikasi Couchbase

Sumber daya terkait

- Layanan Server Couchbase
- Menyebarkan Server Couchbase Menggunakan AWS Marketplace
- Connect ke Couchbase Server
- Kelola Bucket
- Replikasi Pusat Data Lintas (XDCR)
- Perjanjian Lisensi Couchbase Inc.

Informasi tambahan

Manfaat utama

Migrasi database Couchbase Anda untuk AWS memberikan keuntungan berikut:

Skalabilitas. Anda dapat menskalakan cluster Couchbase Anda naik atau turun berdasarkan permintaan tanpa harus mengelola perangkat keras fisik, sehingga Anda dapat dengan mudah mengakomodasi volume data yang berfluktuasi dan penggunaan aplikasi. AWS menyediakan:

- Opsi penskalaan vertikal dan horizontal
- Kemampuan penyebaran global
- Load balancing di seluruh Wilayah AWS

- Solusi penskalaan basis data
- Optimalisasi pengiriman konten

Optimalisasi kinerja. AWS menyediakan infrastruktur jaringan berkinerja tinggi dan tipe instans yang dioptimalkan untuk memastikan akses data yang cepat dan latensi rendah untuk database Couchbase Anda.

- Opsi komputasi kinerja tinggi (HPC)
- Pengiriman konten global melalui Amazon CloudFront
- Beberapa opsi penyimpanan
- Layanan database tingkat lanjut, termasuk Amazon Relational Database Service (Amazon RDS) dan Amazon DynamoDB
- Koneksi latensi rendah dengan Direct Connect

Optimalisasi biaya. Pilih jenis dan konfigurasi instans yang sesuai untuk menyeimbangkan kinerja dan biaya berdasarkan beban kerja Anda. Bayar hanya untuk sumber daya yang Anda gunakan. Ini berpotensi mengurangi biaya operasional Anda dengan menghilangkan kebutuhan untuk mengelola perangkat keras lokal dan memanfaatkan AWS Cloud skala ekonomi.

- Instans cadangan dapat membantu Anda merencanakan ke depan dan mengurangi biaya secara substansional saat Anda menggunakan Couchbase di. AWS
- Penskalaan otomatis mencegah penyediaan berlebih dan membantu Anda mengoptimalkan pemanfaatan dan efisiensi biaya.

Keamanan yang ditingkatkan. Manfaatkan fitur keamanan yang kuat AWS, seperti enkripsi data, kontrol akses, dan grup keamanan untuk membantu melindungi data sensitif yang Anda simpan di Couchbase. Manfaat tambahan:

- Model Tanggung Jawab AWS Bersama jelas membedakan antara keamanan cloud (AWS tanggung jawab) dan keamanan di cloud (tanggung jawab pelanggan).
- AWS kepatuhan mendukung standar keamanan utama.
- AWS menyediakan opsi enkripsi lanjutan.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses aman ke sumber daya Anda.

Manajemen yang disederhanakan. AWS menyediakan layanan terkelola untuk Couchbase, sehingga Anda dapat fokus pada pengembangan aplikasi alih-alih mengelola infrastruktur yang mendasarinya.

Jangkauan global. Anda dapat menerapkan cluster Couchbase Anda di beberapa Wilayah AWS untuk mencapai latensi rendah bagi pengguna di seluruh dunia. Anda dapat menyebarkan database Anda sepenuhnya di cloud atau di lingkungan hybrid. Anda dapat melindungi data Anda dengan keamanan tingkat perusahaan bawaan dan sinkronisasi data dua arah yang cepat dan efisien dari edge ke cloud. Pada saat yang sama, Anda dapat menyederhanakan pengembangan dengan model pemrograman yang konsisten untuk membangun aplikasi web dan seluler.

Kelangsungan bisnis:

- Pencadangan dan pemulihan data. Jika terjadi masalah, Anda dapat menggunakannya AWS Backupuntuk memastikan ketahanan data dan pemulihan yang mudah. Untuk opsi pemulihan bencana, lihat dokumentasi AWS Well-Architected Framework.
- Penyebaran multi-wilayah Couchbase: Untuk menerapkan database Couchbase di AWS lingkungan Multi-wilayah, Anda dapat berlangganan Server Couchbase di, menggunakan AWS CloudFormationtemplat untuk membuat kluster Couchbase terpisah di setiap Wilayah AWS Marketplace, lalu mengonfigurasi replikasi Lintas wilayah untuk menyinkronkan data di seluruh Wilayah. Konfigurasi ini memastikan ketersediaan tinggi dan redundansi geografis di beberapa Wilayah. Untuk informasi selengkapnya, lihat Menerapkan Server Couchbase Menggunakan AWS Marketplace dalam dokumentasi Couchbase.

Kelincahan infrastruktur:

- Penyediaan sumber daya yang cepat dan deprovisioning
- Jangkauan infrastruktur global
- Penskalaan otomatis berdasarkan permintaan
- Infrastructure as Code (IAc) untuk penerapan yang konsisten
- Beberapa jenis instans yang dioptimalkan untuk beban kerja yang berbeda

Pemberdayaan inovasi:

- Akses ke teknologi terbaru, termasuk AI/ML, IoT, dan analitik
- Layanan terkelola, yang mengurangi overhead operasional
- Praktik pengembangan aplikasi modern

- Opsi komputasi tanpa server

Keunggulan operasional:

- Pemantauan dan pencatatan terpusat
- Manajemen sumber daya otomatis
- Kemampuan pemeliharaan prediktif
- Peningkatan visibilitas ke dalam penggunaan sumber daya
- Proses penyebaran yang efisien

Peluang modernisasi:

- Arsitektur microservices
- DevOpspraktek implementasi
- Pengembangan aplikasi cloud-native
- Modernisasi aplikasi lama

Keunggulan kompetitif:

- Waktu yang lebih cepat untuk memasarkan
- Pengalaman pelanggan yang lebih baik
- Pengambilan keputusan berbasis data
- Kecerdasan bisnis yang ditingkatkan

Migrasi Db2 untuk LUW ke EC2 Amazon dengan menggunakan pengiriman log untuk mengurangi waktu pemadaman

Feng Cai, Ambarish Satarkar, dan Saurabh Sharma, Amazon Web Services

Ringkasan

Ketika pelanggan memigrasikan beban kerja IBM Db2 mereka untuk LUW (Linux, UNIX, dan Windows) ke Amazon Web Services (AWS), menggunakan Amazon Elastic Compute Cloud (EC2Amazon) dengan model Bring Your Own License (BYOL) adalah cara tercepat. Namun, memigrasikan sejumlah besar data dari Db2 lokal ke AWS dapat menjadi tantangan, terutama ketika jendela pemadaman pendek. Banyak pelanggan mencoba mengatur jendela pemadaman menjadi kurang dari 30 menit, yang menyisakan sedikit waktu untuk database itu sendiri.

Pola ini mencakup cara menyelesaikan migrasi Db2 dengan jendela pemadaman pendek dengan menggunakan pengiriman log transaksi. Pendekatan ini berlaku untuk Db2 pada platform Linux kecil.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Instance Db2 yang berjalan pada EC2 instance yang cocok dengan tata letak sistem file lokal
- Bucket Amazon Simple Storage Service (Amazon S3) yang dapat diakses oleh instans EC2
- Kebijakan dan peran AWS Identity and Access Management (IAM) untuk melakukan panggilan terprogram ke Amazon S3
- Zona waktu dan jam sistem yang disinkronkan di Amazon EC2 dan server lokal
- Jaringan lokal yang terhubung ke AWS melalui AWS Site-to-SiteVPN atau AWS Direct Connect

Batasan

- Instans lokal Db2 dan Amazon EC2 harus berada di keluarga platform yang sama.
- Beban kerja lokal Db2 harus dicatat. Untuk memblokir transaksi yang tidak tercatat, atur `blocknonlogged=yes` dalam konfigurasi database.

Versi produk

- Db2 untuk LUW versi 11.5.9 dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Db2 di Linux x86_64

Tumpukan teknologi target

- Amazon EBS
- Amazon EC2
- AWS Identity and Access Management (IAM)
- Amazon S3
- AWS Site-to-Site VPN atau Direct Connect

Arsitektur target

Diagram berikut menunjukkan satu instans Db2 yang berjalan di lokasi dengan koneksi jaringan pribadi virtual (VPN) ke Db2 di Amazon. EC2 Garis putus-putus mewakili terowongan VPN antara pusat data Anda dan AWS Cloud.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Direct Connect menautkan jaringan internal Anda ke lokasi Direct Connect melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke layanan AWS publik sambil melewati penyedia layanan internet di jalur jaringan Anda.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon Elastic Compute Cloud (Amazon). EC2
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Site-to-Site VPN membantu Anda meneruskan lalu lintas antar instans yang Anda luncurkan di AWS dan jaringan jarak jauh Anda sendiri.

Alat-alat lainnya

- db2cli adalah perintah CLI interaktif Db2.

Praktik terbaik

- Pada database target, gunakan titik akhir gateway untuk Amazon S3 untuk mengakses gambar cadangan database dan file log di Amazon S3.
- Pada database sumber, gunakan AWS PrivateLink untuk Amazon S3 untuk mengirim gambar cadangan database dan file log ke Amazon S3.

Epik

Tetapkan variabel lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan variabel lingkungan.	Pola ini menggunakan nama-nama berikut: • Nama instans: `db2inst1` • Nama database: `SAMPLE` Anda dapat mengubahnya agar sesuai dengan lingkungan Anda.	DBA

Konfigurasikan server Db2 lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Atur AWS CLI.	Untuk mengunduh dan menginstal AWS CLI versi terbaru, jalankan perintah berikut:
	`$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"`
	`unzip awscliv2.zip`
	`sudo ./aws/install`
		Administrator Linux

Siapkan tujuan lokal untuk log arsip Db2.	Agar database target EC2 di Amazon tetap sinkron dengan database sumber lokal, log transaksi terbaru harus diambil dari sumbernya. Dalam pengaturan ini, `/db2logs` diatur oleh `LOGARCHMETH2` pada sumber sebagai area pementasan. Log yang diarsipkan di direktori ini akan disinkronkan ke Amazon S3 dan diakses oleh Db2 di Amazon. EC2 Pola yang digunakan `LOGARCHMETH2` karena `LOGARCHMETH1` mungkin telah dikonfigurasi untuk menggunakan alat vendor pihak ketiga yang tidak dapat diakses oleh perintah	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	AWS CLI. Untuk mengambil log, jalankan perintah berikut:
	`db2 connect to sample`
	`db2 update db cfg for SAMPLE using LOGARCHMETH2 disk:/db2logs`	

Jalankan backup database online.	Jalankan backup database online, dan simpan ke sistem file backup lokal: `db2 backup db sample online to /backup`	DBA

Siapkan bucket S3 dan kebijakan IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket S3 untuk server lokal untuk mengirim image Db2 cadangan dan file log ke AWS. Bucket juga akan diakses oleh Amazon EC2: `aws s3api create-bucket --bucket logshipmig-db2 --region us-east-1`	Administrator sistem AWS
Buat kebijakan IAM.	`db2bucket.json` File berisi kebijakan IAM untuk mengakses bucket Amazon S3: `{`	Administrator AWS, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"kms:GenerateDataKey",
	"kms:Decrypt",
	"s3:PutObject",
	"s3:GetObject",
	"s3:AbortMultipartUpload",
	"s3:ListBucket",
	"s3:DeleteObject",
	"s3:GetObjectVersion",
	"s3:ListMultipartUploadParts"
	],
	"Resource": [
	"arn:aws:s3:::logshipmig-db2/*",
	"arn:aws:s3:::logshipmig-db2"
	]
	}
	]</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`}`
	
	Untuk membuat kebijakan, gunakan perintah AWS CLI berikut:
	
	`aws iam create-policy \ --policy-name db2s3policy \ --policy-document file://db2bucket.json`
	
	Output JSON menunjukkan Nama Sumber Daya Amazon (ARN) untuk kebijakan, yang mewakili ID `aws_account_id` akun Anda:
	
	`"Arn": "arn:aws:iam::aws_account_id:policy/db2s3policy"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan kebijakan IAM ke peran IAM yang digunakan oleh instance. EC2	Di sebagian besar lingkungan AWS, EC2 instans yang sedang berjalan memiliki Peran IAM yang ditetapkan oleh administrator sistem Anda. Jika peran IAM tidak disetel, buat peran dan pilih Ubah peran IAM di EC2 konsol untuk mengaitkan peran dengan EC2 instance yang menghosting database Db2. Lampirkan kebijakan IAM ke peran IAM dengan kebijakan ARN: `aws iam attach-role-policy \ --policy-arn "arn:aws:iam::aws_account_id:policy/db2s3policy" \ --role-name db2s3role` Setelah kebijakan dilampirkan, EC2 instance apa pun yang terkait dengan peran IAM dapat mengakses bucket S3.	Administrator AWS, administrator sistem AWS

Kirim gambar cadangan database sumber dan file log ke Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan AWS CLI di server Db2 lokal.	Konfigurasikan AWS CLI dengan `Access Key ID`	Administrator AWS, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dan `Secret Access Key` dihasilkan pada langkah sebelumnya:
	`$ aws configure`
	`AWS Access Key ID`
	`[None]: **********`
	`AWS Secret Access Key`
	`[None]: **********`
	`Default region name`
	`[None]: us-east-1`
	`Default output format`
	`[None]: json`	

Kirim gambar cadangan ke Amazon S3.	Sebelumnya, cadangan database online disimpan ke direktori `/backup` lokal. Untuk mengirim gambar cadangan itu ke bucket S3, jalankan perintah berikut: `aws s3 sync /backup s3://logshipmig-db2/SAMPLE_backup`	Administrator AWS, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim log arsip Db2 ke Amazon S3.	Sinkronkan log arsip Db2 lokal dengan bucket S3 yang dapat diakses oleh instans Db2 target di Amazon: EC2 `aws s3 sync /db2logs s3://logshipmig-db2/SAMPLE_LOG` Jalankan perintah ini secara berkala dengan menggunakan cron atau alat penjadwalan lainnya. Frekuensi tergantung pada seberapa sering database sumber mengarsipkan file log transaksi.	Administrator AWS, Insinyur migrasi

Connect Db2 di Amazon EC2 ke Amazon S3 dan mulai sinkronisasi database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat PKCS12 keystore.	Db2 menggunakan keystore enkripsi Public-Key Cryptography Standards (PKCS) untuk menjaga kunci akses AWS tetap aman. Buat keystore dan konfigurasikan instance sumber Db2 untuk menggunakannya: `gsk8capicmd_64 -keydb -create -db "/home/db2inst1/.keystore/db2s3.p12" -pw "<passwor`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`d>" -type pkcs12 -stash`
	
	`db2 "update dbm cfg using keystore_location /home/db2inst1/.keystore/db2s3.p12 keystore_type pkcs12"`	

Buat alias akses penyimpanan Db2.	Untuk membuat alias akses penyimpanan, gunakan sintaks skrip berikut:
	
	`db2 "catalog storage access alias <alias_name> vendor S3 server <S3 endpoint> container '<bucket_name>'"`
	
	Misalnya, skrip Anda mungkin terlihat seperti berikut:
	
	`db2 "catalog storage access alias DB2AWSS3 vendor S3 server s3.us-east-1.amazonaws.com container 'logshipmig-db2'"`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Atur area pementasan.	Secara default, Db2 digunakan `DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH` sebagai area pementasan untuk mengunggah dan mengunduh file ke dan dari Amazon S3. Jalur default berada `sqllib/tmp/RemoteStorage.xxxx` di bawah direktori home instance, dengan `xxxx` mengacu pada nomor partisi Db2. Perhatikan bahwa area pementasan harus memiliki kapasitas yang cukup untuk menyimpan gambar cadangan dan file log. Anda dapat menggunakan registri untuk mengarahkan area pementasan ke direktori yang berbeda. Kami juga merekomendasikan penggunaan `DB2_ENABLE_COS_SDK=ON` ,`DB2_OBJECT_STORAGE_SETTINGS=EnableStreamingRestore` , dan tautan ke `awssdk` perpustakaan untuk melewati area pementasan Amazon S3 untuk pencadangan dan pemulihan basis data: `#By root:`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cp -rp /home/db2inst1/sqllib/lib64/awssdk/RHEL/7.6/* /home/db2inst1/sqllib/lib64/`
	
	`#By db2 instance owner:`
	`db2set DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH=/db2stage`
	`db2set DB2_ENABLE_COS_SDK=ON`
	`Db2set DB2_OBJECT_STORAGE_SETTINGS=EnableStreamingRestore`
	`db2stop`
	`db2start`	

Kembalikan database dari gambar cadangan.	Kembalikan database target di Amazon EC2 dari gambar cadangan di bucket S3: `db2 restore db sample from DB2REMOTE://DB2AWSS3/logshipmig-db2/SAMPLE_backup replace existing`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Gulung ke depan database.	Setelah pemulihan selesai, database target akan dimasukkan ke status tertunda rollforward. Konfigurasikan `LOGARCHMETH1` dan `LOGARCHMETH2` agar Db2 tahu di mana mendapatkan file log transaksi:
	`db2 update db cfg for SAMPLE using LOGARCHMETH1 'DB2REMOTE://DB2AWSS3//SAMPLE_LOGS/'`
	`db2 update db cfg for SAMPLE using LOGARCHMETH2 OFF`
	Mulai rollforward database:
	`db2 ROLLFORWARD DATABASE sample to END OF LOGS`
	Perintah ini memproses semua file log yang telah ditransfer ke bucket S3. Jalankan secara berkala berdasarkan frekuensi `s3 sync` perintah di server Db2 lokal. Misalnya, jika `s3 sync` berjalan pada setiap jam, dan dibutuhkan 10 menit untuk menyinkronkan semua file log, atur perintah untuk berjalan
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pada 10 menit setelah setiap jam.	

Bawa Db2 di Amazon EC2 online selama jendela cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bawa database target secara online.	Selama jendela cutover, lakukan salah satu hal berikut:
	• Masukkan database lokal`ADMIN MODE`, dan jalankan `s3 sync` perintah untuk memaksa log transaksi terakhir diarsipkan.
	• Matikan database.
	
	Setelah log transaksi terakhir disinkronkan ke Amazon S3, jalankan perintah untuk `ROLLFORWARD` terakhir kalinya:
	
	`db2 rollforward DB sample to END OF LOGS`
	`db2 rollforward DB sample complete`
	
	`Rollforward Status`
	`....`
	`Rollforward status = not pending`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`....`
	`DB20000I  The ROLLFORWA`
	`RD command completed`
	` successfully.`
	
	`db2 activate db sample`
	`DB20000I  The ACTIVATE`
	` DATABASE command`
	` completed successfu`
	`lly.`
	
	Bawa database target online, dan arahkan koneksi aplikasi ke Db2 di Amazon. EC2	

Pemecahan masalah

Isu	Solusi
Jika beberapa database memiliki nama instance dan nama database yang sama pada host yang berbeda (DEV, QA, PROD), backup dan log mungkin masuk ke subdirektori yang sama.	Gunakan bucket S3 yang berbeda untuk DEV, QA, dan PROD, dan tambahkan nama host sebagai awalan subdirektori untuk menghindari kebingungan.
Jika ada beberapa gambar cadangan di lokasi yang sama, Anda akan mendapatkan kesalahan berikut saat memulihkan: `SQL2522N More than one backup file matches the time stamp value provided for the backed up database image.`	Dalam `restore` perintah, tambahkan stempel waktu cadangan: `db2 restore db sample from DB2REMOTE://DB2AWSS3/logshi pmig-db2/SAMPLE_backup taken at 20230628164042 replace existing`

Sumber daya terkait

- Operasi pencadangan dan pemulihan Db2 antara sistem operasi dan platform perangkat keras yang berbeda
- Mengatur Db2 STORAGE ACCESS ALIAS dan REMOTE DB2
- Perintah Db2 ROLLFORWARD
- Metode arsip log sekunder Db2

Migrasi Db2 untuk LUW ke EC2 Amazon dengan pemulihan bencana ketersediaan tinggi

Feng Cai, Aruna Gangireddy, dan Venkatesan Govindan, Amazon Web Services

Ringkasan

Saat pelanggan memigrasikan beban kerja IBM Db2 LUW (Linux, UNIX, dan Windows) mereka ke Amazon Web Services (AWS), menggunakan Amazon Elastic Compute Cloud (EC2Amazon) dengan model Bring Your Own License (BYOL) adalah cara tercepat. Namun, memigrasikan sejumlah besar data dari Db2 lokal ke AWS dapat menjadi tantangan, terutama ketika jendela pemadaman pendek. Banyak pelanggan mencoba mengatur jendela pemadaman menjadi kurang dari 30 menit, yang menyisakan sedikit waktu untuk database itu sendiri.

Pola ini mencakup cara menyelesaikan migrasi Db2 dengan jendela pemadaman pendek dengan menggunakan Db2 high availability disaster recovery (HADR). Pendekatan ini berlaku untuk database Db2 yang berada di platform Linux kecil dan tidak menggunakan Fitur Partisi Data (DPF).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Instans Db2 yang berjalan pada EC2 instans Amazon yang cocok dengan tata letak sistem file lokal
- Bucket Amazon Simple Storage Service (Amazon S3) yang dapat diakses oleh instans EC2
- Kebijakan dan peran AWS Identity and Access Management (IAM) untuk melakukan panggilan terprogram ke Amazon S3
- Zona waktu dan jam sistem yang disinkronkan di Amazon EC2 dan server lokal
- Jaringan lokal yang terhubung ke AWS melalui AWS Site-to-SiteVPN atau AWS Direct Connect
- Komunikasi antara server lokal dan Amazon EC2 di port HADR

Keterbatasan

- Instans lokal Db2 dan Amazon EC2 harus berada di keluarga platform yang sama.
- HADR tidak didukung dalam lingkungan database yang dipartisi.

- HADR tidak mendukung penggunaan mentah I/O (akses disk langsung) untuk file log database.
- HADR tidak mendukung logging tak terbatas.
- `LOGINDEXBUILD`harus disetel ke`YES`, yang akan meningkatkan penggunaan log untuk membangun kembali indeks.
- Beban kerja lokal Db2 harus dicatat. Tetapkan `blocknonlogged=yes` dalam konfigurasi database untuk memblokir transaksi yang tidak tercatat.

Versi produk

- Db2 untuk LUW versi 11.5.9 dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Db2 di Linux x86_64

Tumpukan teknologi target

- Amazon EC2
- AWS Identity and Access Management (IAM)
- Amazon S3
- AWS Site-to-Site VPN

Arsitektur target

Dalam diagram berikut, Db2 di tempat berjalan `db2-server1` sebagai primer. Ini memiliki dua target siaga HADR. Satu target siaga ada di tempat dan bersifat opsional. Target siaga lainnya,`db2-ec2`, ada di Amazon EC2. Setelah database dipotong ke AWS, `db2-ec2` menjadi yang utama.

1. Log dialirkan dari database lokal utama ke database lokal siaga.
2. Menggunakan Db2 HADR, log dialirkan dari database lokal utama melalui VPN Site-to-Site ke Db2 di Amazon. EC2
3. Pencadangan Db2 dan log arsip dikirim dari database lokal utama ke bucket S3 di AWS.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Direct Connect menautkan jaringan internal Anda ke lokasi Direct Connect melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke layanan AWS publik sambil melewati penyedia layanan internet di jalur jaringan Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Site-to-Site VPN membantu Anda meneruskan lalu lintas antar instans yang Anda luncurkan di AWS dan jaringan jarak jauh Anda sendiri.

Alat lainnya

- db2cli adalah perintah CLI interaktif Db2.

Praktik terbaik

- Pada database target, gunakan titik akhir gateway untuk Amazon S3 untuk mengakses gambar cadangan database dan file log di Amazon S3.
- Pada database sumber, gunakan AWS PrivateLink untuk Amazon S3 untuk mengirim gambar cadangan database dan file log ke Amazon S3.

Epik

Tetapkan variabel lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan variabel lingkungan.	Pola ini menggunakan nama dan port berikut: 1. Nama host lokal Db2: `db2-server1` 2. Nama host siaga HADR: `db2-server2` (jika HADR saat ini berjalan di tempat) 3. EC2 Nama host Amazon: `db2-ec2` 4. Nama instans: `db2inst1` 5. Nama database: `SAMPLE` 6. Port HADR: • `db2-server1: 50010` • `db2-server2: 50011` • `db2-ec2: 50012` Anda dapat mengubahnya agar sesuai dengan lingkunga n Anda.	DBA

Konfigurasikan server Db2 lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS CLI.	Untuk mengunduh dan menginstal AWS CLI versi terbaru, jalankan perintah berikut:	Administrator Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"`
	`unzip awscliv2.zip`
	`sudo ./aws/install`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan tujuan lokal untuk log arsip Db2.	Kondisi seperti pekerjaan batch pembaruan berat dan perlambatan jaringan dapat menyebabkan server siaga HADR mengalami lag. Untuk catch up, server siaga membutuhkan log transaksi dari server utama. Urutan tempat untuk meminta log adalah sebagai berikut: • Direktori log aktif di server utama • `LOGARCHMETH2` Lokasi `LOGARCHMETH1` atau di server siaga • `LOGARCHMETH2` Lokasi `LOGARCHMETH1` atau di server utama Dalam pengaturan ini, `/db2logs` diatur oleh `LOGARCHMETH2` pada sumber sebagai area pementasan. Log yang diarsipkan di direktori ini akan disinkronkan ke Amazon S3 dan diakses oleh Db2 di Amazon. EC2 Pola yang digunakan `LOGARCHMETH2` karena `LOGARCHMETH1` mungkin telah dikonfigurasi untuk menggunakan alat vendor pihak ketiga yang tidak	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dapat diakses oleh perintah AWS CLI:
	`db2 connect to sample`
	`db2 update db cfg for SAMPLE using LOGARCHMETH2 disk:/db2logs`	

Jalankan backup database online.	Jalankan backup database online, dan simpan ke sistem file backup lokal: `db2 backup db sample online to /backup`	DBA

Siapkan bucket S3 dan kebijakan IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket S3 untuk server lokal untuk mengirim image Db2 cadangan dan file log ke AWS. Bucket akan diakses oleh Amazon EC2: `aws s3api create-bucket --bucket hadrmig-db2 --region us-east-1`	Administrator AWS
Buat kebijakan IAM.	`db2bucket.json` File berisi kebijakan IAM untuk mengakses bucket S3: `{`	Administrator AWS, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"kms:GenerateDataKey",
	"kms:Decrypt",
	"s3:PutObject",
	"s3:GetObject",
	"s3:AbortMultipartUpload",
	"s3:ListBucket",
	"s3:DeleteObject",
	"s3:GetObjectVersion",
	"s3:ListMultipartUploadParts"
	],
	"Resource": [
	"arn:aws:s3:::hadrmig-db2/*",
	"arn:aws:s3:::hadrmig-db2"
	]
	}
	]</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`}`
	
	Untuk membuat kebijakan, gunakan perintah AWS CLI berikut:
	
	`aws iam create-policy \ --policy-name db2s3hapolicy \ --policy-document file://db2bucket.json`
	
	Output JSON menunjukkan Nama Sumber Daya Amazon (ARN) untuk kebijakan, yang mewakili ID `aws_account_id` akun Anda:
	
	`"Arn": "arn:aws:iam::aws_account_id:policy/db2s3hapolicy"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan kebijakan IAM ke peran IAM.	Biasanya, EC2 instance dengan Db2 berjalan akan memiliki peran IAM yang ditetapkan oleh administrator sistem. Jika tidak ada peran IAM yang ditetapkan, Anda dapat memilih Ubah peran IAM di konsol Amazon EC2 . Lampirkan kebijakan IAM ke peran IAM yang terkait dengan instance. EC2 Setelah kebijakan dilampirkan, EC2 instance dapat mengakses bucket S3: `aws iam attach-role-policy --policy-arn "arn:aws:iam::aws_account_id:policy/db2s3hapolicy" --role-name db2s3harole`	

Kirim gambar cadangan database sumber dan file log ke Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan AWS CLI di server Db2 lokal.	Konfigurasikan AWS CLI dengan `Access Key ID` dan `Secret Access Key` yang Anda buat sebelumnya:
	
	`$ aws configure`
	`AWS Access Key ID [None]: **********`
		Administrator AWS, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`AWS Secret Access Key [None]: ********* ********** Default region name [None]: us-east-1 Default output format [None]: json`	
Kirim gambar cadangan ke Amazon S3.	Sebelumnya, cadangan database online disimpan ke direktori `/backup` lokal. Untuk mengirim gambar cadangan itu ke bucket S3, jalankan perintah berikut: `aws s3 sync /backup s3://hadrmig-db2/SAMPLE_backup`	Administrator AWS, administrator sistem AWS
Kirim log arsip Db2 ke Amazon S3.	Sinkronkan log arsip Db2 lokal dengan bucket Amazon S3 yang dapat diakses oleh instans Db2 target di Amazon: EC2 `aws s3 sync /db2logs s3://hadrmig-db2/SAMPLE_LOGS` Jalankan perintah ini secara berkala dengan menggunakan cron atau alat penjadwalan lainnya. Frekuensi tergantung pada seberapa sering database sumber mengarsipkan file log transaksi.	

Connect Db2 di Amazon EC2 ke Amazon S3 dan mulai sinkronisasi database awal

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat PKCS12 keystore.	Db2 menggunakan keystore enkripsi Public-Key Cryptography Standards (PKCS) untuk menjaga kunci akses AWS tetap aman. Buat keystore, dan konfigurasikan sumber Db2 untuk menggunakannya:
	`gsk8capicmd_64 -keydb -create -db "/home/db2inst1/.keystore/db2s3.p12" -pw "<password>" -type pkcs12 -stash`
	
	`db2 "update dbm cfg using keystore_location /home/db2inst1/.keystore/db2s3.p12 keystore_type pkcs12"`
		DBA

Buat alias akses penyimpanan Db2.	Db2 menggunakan alias akses penyimpanan untuk mengakses Amazon S3 secara langsung dengan`INGEST`,,, `LOAD` atau perintah. `BACKUP DATABASE` `RESTORE DATABASE` Karena Anda menetapkan peran IAM ke EC2 instance, `USER` dan tidak `PASSWORD` diperlukan:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`db2 "catalog storage access alias <alias_name> vendor S3 server <S3 endpoint> container '<bucket_name>'"`
	
	Misalnya, skrip Anda mungkin terlihat seperti berikut:
	
	`db2 "catalog storage access alias DB2AWSS3 vendor S3 server s3.us-east-1.amazonaws.com container 'hadrmig-db2'"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Atur area pementasan.	Sebaiknya gunakan`DB2_ENABL E_COS_SDK =ON` ,`DB2_OBJEC T_STORAGE_SETTINGS =EnableStreamingRe store` , dan tautan ke `awssdk` perpustakaan untuk melewati area pementasan Amazon S3 untuk pencadang an dan pemulihan basis data:
	`#By root:`
	`cp -rp /home/db2inst1/ sqllib/lib64/awssdk/ RHEL/7.6/* /home/db2 inst1/sqllib/lib64/`
	
	`#By db2 instance owner:`
	`db2set DB2_OBJEC T_STORAGE_LOCAL_ST AGING_PATH=/db2stage`
	`db2set DB2_ENABL E_COS_SDK=ON`
	`db2set DB2_OBJEC T_STORAGE_LOCAL_ST AGING_PATH=/db2stage`
	`db2stop`
	`db2start`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kembalikan database dari gambar cadangan.	Kembalikan database target di Amazon EC2 dari gambar cadangan di bucket S3:
	`db2 create db sample on /data1`
	`db2 restore db sample from DB2REMOTE://DB2AWSS3/hadrmig-db2/SAMPLE_backup replace existing`
		DBA

Siapkan HADR tanpa HADR di tempat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan server Db2 lokal sebagai yang utama.	Memperbarui pengaturan konfigurasi database untuk HADR pada `db2-server1` (sumber lokal) sebagai primer. Setel `HADR_SYNCMODE` ke `SUPERASYNC` mode, yang memiliki waktu respons transaksi terpendek: `db2 update db cfg for sample using HADR_LOCAL_HOST db2-server1 HADR_LOCAL_SVC 50010 HADR_REMOTE_HOST db2-ec2 HADR_REMOTE_SVC 50012 HADR_REMOTE_INST db2inst1 HADR_SYNCMODE SUPERASYNC DB20000`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`I The UPDATE DATABASE CONFIGURATION command completed successfu lly` Beberapa penundaan jaringan antara pusat data lokal dan AWS diharapkan terjadi. (Anda dapat menetapka n `HADR_SYNCMODE` nilai yang berbeda berdasarkan keandalan jaringan. Untuk informasi selengkapnya, lihat bagian Sumber daya terkait).	

Ubah tujuan arsip log database target.	Ubah tujuan arsip log database target agar sesuai dengan EC2 lingkungan Amazon:
	
	`db2 update db cfg for SAMPLE using LOGARCHMETH1 'DB2REMOTE://DB2AWSS3//SAMPLE_LOGS/' LOGARCHMETH2 OFF`
	`DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan HADR untuk Db2 di server Amazon. EC2	Perbarui konfigurasi database untuk HADR aktif `db2-ec2` sebagai siaga: `db2 update db cfg for sample using HADR_LOCAL_HOST db2-ec2 HADR_LOCAL_SVC 50012 HADR_REMOTE_HOST db2-server1 HADR_REMOTE_SVC 50010 HADR_REMOTE_INST db2inst1 HADR_SYNCMODE SUPERASYNC DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi pengaturan HADR.	Verifikasi parameter HADR pada sumber dan target server Db2.
	
	Untuk memverifikasi pengaturan`db2-server1` , jalankan perintah berikut:
	
	`db2 get db cfg for sample|grep HADR`
	`HADR database role = PRIMARY`
	`HADR local host name (HADR_LOCAL_HOST) = db2-server1`
	`HADR local service name (HADR_LOCAL_SVC) = 50010`
	`HADR remote host name (HADR_REMOTE_HOST) = db2-ec2`
	`HADR remote service name (HADR_REMOTE_SVC) = 50012`
	`HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1`
	`HADR timeout value (HADR_TIMEOUT) = 120`
	`HADR target list (HADR_TARGET_LIST) =`
	`HADR log write synchronization mode`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>(HADR_SYNCMODE) = NEARSYNC
	HADR spool log data limit (4KB)
	(HADR_SPOOL_LIMIT) = AUTOMATIC(52000)
	HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0
	HADR peer window duration (seconds)
	(HADR_PEER_WINDOW) = 0
	HADR SSL certificate label
	(HADR_SSL_LABEL) =
	HADR SSL Hostname Validation
	(HADR_SSL_HOST_VAL) = OFF</pre>
	Untuk memverifikasi pengaturan`db2-ec2`, jalankan perintah berikut:
	<pre>db2 get db cfg for sample|grep HADR
	HADR database role
	
	= STANDBY
	HADR local host name (HADR_LOCAL_HOST) = db2-ec2
	HADR local service name
	(HADR_LOCAL_SVC) = 50012
	HADR remote host name (HADR_REMOTE_HOST) = db2-server1</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	HADR remote service name (HADR_REMOTE_SVC) = 50010 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = HADR log write synchronization mode (HADR_SYNCMODE) = SUPERASYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF HADR_REMOTE_SVC Parameter HADR_LOCAL_HOST HADR_LOCA	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	L_SVC ,HADR_REMO TE_HOST ,, dan menunjukkan pengaturan HADR satu primer dan satu siaga.	

Mulai instance Db2 HADR.	Mulai instance Db2 HADR di server siaga terlebih dahulu: db2-ec2
	`db2 start hadr on db sample as standby DB20000I The START HADR ON DATABASE command completed successfully.`
	Mulai Db2 HADR di server utama (sumber): db2-serve r1
	`db2 start hadr on db sample as primary DB20000I The START HADR ON DATABASE command completed successfully.`
	Koneksi HADR antara Db2 di tempat dan di Amazon kini EC2 telah berhasil dibuat. Server utama Db2 db2-server1 memulai streaming catatan log transaksi db2-ec2 secara real time.
		DBA

Mengatur HADR ketika HADR ada di tempat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan Db2 di Amazon EC2 sebagai siaga tambahan.	Jika HADR berjalan pada instans Db2 lokal, Anda dapat menambahkan Db2 di Amazon EC2 sebagai siaga tambahan menggunakan HADR_TARGET_LIST dengan menjalankan perintah berikut pada: db2-ec2 `db2 update db cfg for sample using HADR_LOCAL_HOST db2-ec2 HADR_LOCAL_SVC 50012 HADR_REMOTE_HOST db2-server1 HADR_REMOTE_SVC 50010 HADR_REMOTE_INST db2inst1 HADR_SYNCMODE SUPERASYNC DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully. db2 update db cfg for sample using HADR_TARGET_LIST "db2-server1:50010 |db2-server2:50011 " DB20000I The UPDATE DATABASE CONFIGURATION command`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`completed successfully.`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan informasi siaga tambahan ke server lokal.	Pembaruan `HADR_TARGET_LIST` pada dua server lokal (primer dan siaga). `db2-server1` Aktif, jalankan kode berikut: `db2 update db cfg for sample using HADR_TARGET_LIST "db2-server2:50011|db2-ec2:50012" DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully. SQL1363W One or more of the parameters submitted for immediate modification were not changed dynamically. For these configuration parameters, the database must be shutdown and reactivated before the configuration parameter changes become effective.`  `db2-server2` Aktif, jalankan kode berikut:  `db2 update db cfg for sample using HADR_TARGET_LIST "db2-serv`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	er1:50010|db2-ec2: 50012" DB20000I The UPDATE DATABASE CONFIGURATION command completed successfu lly. SQL1363W One or more of the parameter s submitted for immediate modificat ion were not changed dynamically. For these configura tion parameters, the database must be shutdown and reactivated before the configuration parameter changes become effective.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi pengaturan HADR.	Verifikasi parameter HADR pada sumber dan target server Db2. `db2-server1` Aktif, jalankan kode berikut: `db2 get db cfg for sample|grep HADR HADR database role = PRIMARY HADR local host name (HADR_LOCAL_HOST) = db2-server1 HADR local service name (HADR_LOCAL_SVC) = 50010 HADR remote host name (HADR_REMOTE_HOST) = db2-server2 HADR remote service name (HADR_REMOTE_SVC) = 50011 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = db2-server2:50011|db2-ec2:50012`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>HADR log write
	synchronization mode
	(HADR_SYNCMODE) =
	NEARSYNC
	HADR spool log data
	limit (4KB)
	(HADR_SPOOL_LIMIT) =
	AUTOMATIC(52000)
	HADR log replay delay
	(seconds) (HADR_REP
	LAY_DELAY) = 0
	HADR peer window
	duration (seconds)
	(HADR_PEER_WINDOW) = 0
	HADR SSL certifica
	te label
	(HADR_SSL_LABEL) =
	HADR SSL Hostname
	Validation
	(HADR_SSL_HOST_VAL) =
	OFF</pre>	

	`db2-server2` Aktif, jalankan kode berikut:
	<pre>db2 get db cfg for
	sample|grep HADR
	HADR database role
	
	= STANDBY
	HADR local host name
	(HADR_LOC
	AL_HOST) = db2-server2
	HADR local service
	name
	(HADR_LOCAL_SVC) =
	50011
	HADR remote host name
	(HADR_REM
	OTE_HOST) = db2-serve
	r1</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	HADR remote service name (HADR_REMOTE_SVC) = 50010 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = db2-server1:50010|db2-ec2:50012 HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF db2-ec2Aktif, jalankan kode berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`db2 get db cfg for sample|grep HADR`
	`HADR database role = STANDBY`
	`HADR local host name (HADR_LOCAL_HOST) = db2-ec2`
	`HADR local service name (HADR_LOCAL_SVC) = 50012`
	`HADR remote host name (HADR_REMOTE_HOST) = db2-server1`
	`HADR remote service name (HADR_REMOTE_SVC) = 50010`
	`HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1`
	`HADR timeout value (HADR_TIMEOUT) = 120`
	`HADR target list (HADR_TARGET_LIST) = db2-server1:50010|db2-server2:50011`
	`HADR log write synchronization mode (HADR_SYNCMODE) = SUPERASYNC`
	`HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF HADR_TARGET_LIST Parameter HADR_LOCAL_HOST HADR_LOCAL_SVC ,HADR_REMOTE_HOST ,HADR_REMOTE_SVC ,, dan menunjukkan satu pengaturan HADR utama dan dua siaga.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Berhenti dan mulai Db2 HADR.	`HADR_TARGET_LIST` sekarang diatur di ketiga server. Setiap server Db2 mengetahui dua lainnya. Hentikan dan mulai ulang HADR (pemadaman singkat) untuk memanfaatkan konfigurasi baru.
	
	`db2-server1` Aktif, jalankan perintah berikut:
	
	`db2 stop hadr on db sample`
	`db2 deactivate db sample`
	`db2 activate db sample`
	
	`db2-server2` Aktif, jalankan perintah berikut:
	
	`db2 deactivate db sample`
	`db2 start hadr on db sample as standby`
	`SQL1766W The command completed successfully`
	
	`db2-ec2`Aktif, jalankan perintah berikut:
	
	`db2 start hadr on db sample as standby`
	`SQL1766W The command completed successfully`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	db2-server1 Aktif, jalankan perintah berikut: `db2 start hadr on db sample as primary SQL1766W The command completed successfully` Koneksi HADR antara Db2 di tempat dan di Amazon sekarang berhasil EC2 dibuat. Server utama Db2 db2-server1 memulai streaming catatan log transaksi ke keduanya db2-server2 dan db2-ec2 secara real time.	

Jadikan Db2 di Amazon EC2 sebagai yang utama selama jendela cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pastikan tidak ada lag HADR di server siaga.	Periksa status HADR dari server db2-server1 utama. Jangan khawatir ketika HADR_STATE dalam REMOTE_CATCHUP status, yang normal ketika HADR_SYNCMODE diatur keSUPERASYNC . PRIMARY_LOG_TIME Dan STANDBY_REPLAY_LOG_TIME menunjukkan bahwa mereka sinkron:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`db2pd -hadr -db sample`
	
	`HADR_ROLE = PRIMARY`
	
	`REPLAY_TYPE = PHYSICAL`
	
	`HADR_SYNCMODE = SUPERASYNC`
	
	`STANDBY_ID = 2`
	
	`LOG_STREAM_ID = 0`
	
	`HADR_STATE = REMOTE_CATCHUP`
	`.....`
	
	`PRIMARY_LOG_TIME = 10/26/2022 02:11:32.000000 (1666750292)`
	
	`STANDBY_LOG_TIME = 10/26/2022 02:11:32.000000 (1666750292)`
	`STANDBY_REPLAY_LOG_TIME = 10/26/2022 02:11:32.000000 (1666750292)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan pengambilalihan HADR.	Untuk menyelesaikan migrasi, buat database `db2-ec2` utama dengan menjalank an perintah pengambilalihan HADR. Gunakan perintah db2pd untuk memverifikasi `HADR_ROLE` nilai:
	
	`db2 TAKEOVER HADR ON DATABASE sample`
	`DB20000I The TAKEOVER HADR ON DATABASE command completed successfully.`
	
	`db2pd -hadr -db sample`
	`Database Member 0 -- Database SAMPLE -- Active -- Up 0 days 00:03:25 -- Date 2022-10-26-02.46.45.048988`
	
	`HADR_ROLE = PRIMARY`
	`REPLAY_TYPE = PHYSICAL`
	
	Untuk menyelesaikan migrasi ke AWS, arahkan koneksi aplikasi ke Db2 di Amazon. EC2	

Pemecahan masalah

Isu	Solusi
Jika Anda menggunakan NAT untuk alasan firewall dan keamanan, host dapat memiliki dua alamat IP (satu internal dan satu eksternal), yang dapat menyebabkan kegagalan pemeriksaan alamat IP HADR. `START HADR ON DATABASE`Perintah akan mengembalikan pesan berikut: `HADR_LOCAL_HOST:HADR_LOCAL_SVC (-xx-xx-xx-xx.:50011 (xx.xx.xx .xx:50011)) on remote database is different from HADR_REMOTE_HOST:H ADR_REMOTE_SVC (xx-xx-xx- xx.:50011 (x.x.x.x:50011)) on local database.`	Untuk mendukung HADR di lingkungan NAT, Anda dapat mengonfigurasi `HADR_LOCA L_HOST` dengan alamat internal dan eksternal . Misalnya, jika server Db2 memiliki nama internal `host1` dan nama eksternal`host1E`, `HADR_LOCAL_HOST` bisa. `HADR_LOCA L_HOST: "host1 | host1E"`

Sumber daya terkait

- Operasi pencadangan dan pemulihan Db2 antara sistem operasi dan platform perangkat keras yang berbeda
- Mengatur Db2 STORAGE ACCESS ALIAS dan REMOTE DB2
- Pemulihan bencana ketersediaan tinggi Db2
- hadr_syncmode - Mode sinkronisasi HADR untuk penulisan log dalam parameter konfigurasi status rekan

Migrasi aplikasi yang dihosting IIS ke Amazon dengan menggunakan appcmd.exe EC2

Deepak Kumar, Amazon Web Services

Ringkasan

Saat memigrasikan aplikasi yang dihosting Internet Information Services (IIS) ke instans Amazon Elastic Compute Cloud (Amazon EC2), Anda perlu mengatasi beberapa tantangan autentikasi. Tantangan ini termasuk memasukkan kembali kredensil domain untuk identitas kumpulan aplikasi dan berpotensi meregenerasi kunci mesin untuk fungsionalitas situs web yang tepat. Anda dapat menggunakan AWS Directory Service untuk membangun hubungan kepercayaan dengan Active Directory lokal atau membuat Active Directory terkelola baru. AWS Pola ini menjelaskan pendekatan migrasi bersih yang menggunakan fungsionalitas pencadangan dan pemulihan IIS di EC2 instans Amazon. Pendekatan ini menggunakan appcmd.exe untuk menghapus dan menginstal ulang IIS pada EC2 instance target, memungkinkan migrasi situs web yang dihosting IIS, identitas kumpulan aplikasi, dan kunci mesin yang berhasil.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS untuk server target.
- Server IIS sumber fungsional dengan situs web yang dihosting di dalamnya.
- Memahami prinsip-prinsip kerja IIS seperti administrasi dan konfigurasi.
- Akses administrator sistem pada server sumber dan target.
- Migrasi lengkap dari server IIS sumber ke target Akun AWS. Anda dapat menggunakan alat migrasi seperti AWS Application Migration Service pendekatan berbasis snapshot Amazon Machine Image (AMI), atau alat migrasi lainnya.

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- IIS 8.5 atau IIS 10.0

Arsitektur

Tumpukan teknologi sumber

- Windows Server dengan IIS 8.5 atau IIS 10.0 diinstal

Tumpukan teknologi target

- Windows Server dengan IIS 8.5 atau IIS 10.0 diinstal
- Application Migration Service

Arsitektur target

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Solusinya meliputi langkah-langkah berikut:

1. Instal dan konfigurasikan Agen AWS Replikasi di server IIS sumber di pusat data perusahaan Anda. Agen ini memulai proses replikasi dan mengelola transfer data ke. AWS
2. Agen AWS Replikasi membuat koneksi aman ke Layanan Migrasi Aplikasi dan mulai mereplikasi data server sumber, termasuk konfigurasi IIS, situs web, dan file aplikasi.
3. Layanan Migrasi Aplikasi meluncurkan EC2 instance di subnet aplikasi dengan data yang direplikasi. EC2 Instans target menjalankan IIS dan berisi aplikasi yang dimigrasi dengan volume Amazon Elastic Block Store (Amazon EBS) terkait. Setelah replikasi awal, Layanan Migrasi Aplikasi terus menyinkronkan perubahan hingga Anda siap untuk memotong ke lingkungan baru.

Alat

Layanan AWS

- AWS Application Migration Servicembantu Anda meng-host ulang (mengangkat dan menggeser) aplikasi ke aplikasi AWS Cloud tanpa perubahan dan dengan waktu henti minimal.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon. EC2
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

Alat lainnya

- Internet Information Services (IIS) untuk Windows Server adalah server web dengan arsitektur terukur dan terbuka untuk hosting apa pun di Web. IIS menyediakan seperangkat alat administrasi, termasuk alat administrasi dan baris perintah (misalnya, appcmd.exe), kode terkelola dan skrip APIs, dan dukungan Windows PowerShell .

Epik

Cadangkan IIS di sumber sebelum migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cadangan situs web yang dihosting IIS, kunci konfigura si, dan kunci. WAS	Untuk membuat cadangan untuk situs web yang dihosting IIS, kunci konfigurasi (`iisConfigurationKey`), dan WAS kunci (`iisWasKey`), gunakan appcmd.exe di server sumber. Gunakan langkah-l angkah berikut: 1. Buka command prompt dan jalankan sebagai administr ator. 2. Jalankan perintah berikut: `c:\windows\system3 2\inetsrv` 3. Jalankan perintah berikut, ganti `<backupname>` dengan nilai Anda: `appcmd.exe add backup "<backupn ame>"`	Administrator IIS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk mengekspor kunci konfigurasi dan WAS kunci, lakukan hal berikut: 1. Jalankan perintah berikut: `cd c:\windows\Microsoft.NET\Framework\v2.0.50727\ aspnet_regiis.exe -px "iisConfigurationKey"` 2. Jalankan perintah berikut: `C:\temp\iisConfKey.xml -pri` 3. Jalankan perintah berikut: `aspnet_regiis.exe -px "iisWasKey" C:\temp\iisWasKey.xml -pri`	

Copot pemasangan dan instal ulang IIS di server target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Copot pemasangan IIS di server target.	Untuk menghapus IIS di server target, gunakan langkah-langkah berikut: 1. Masuk ke server dengan hak administrator. 2. Pergi ke Manajer Server.	Administrator IIS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pilih Kelola, Hapus Peran dan fitur, Berikutnya, dan Berikutnya. 4. Kosongkan kotak centang IIS web dan kemudian pilih Berikutnya, Berikutnya. 5. Pilih kotak centang Restart server tujuan secara otomatis. Ketika penghapus an instalasi selesai, sistem reboot. 6. Validasi bahwa IIS berhasil dihapus dengan membuka `C:\Windows\System32\inetsrv` dan mengonfirmasi bahwa folder ini kosong.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal IIS di server target.	Untuk menginstal IIS di server target, gunakan langkah-l angkah berikut: 1. Masuk ke server dengan hak administrator. 2. Pergi ke Manajer Server. 3. Pilih Tambahkan peran dan fitur, Berikutnya, Berikutnya, Berikutnya, dan Berikutnya. 4. Pilih kotak centang Webserver IIS. Saat diminta, pilih Tambahkan fitur. 5. Pilih semua kotak centang yang ditampilkan dalam Pengembangan aplikasi, Tambahkan fitur, Pilih semua fitur. Anda dapat memilih fitur pada tingkat granular tepat sebelum konfirmasi. 6. Pilih Instal.	Administrator IIS

Kembalikan situs web dan konfigurasi IIS dari cadangan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan situs web dan konfigurasi IIS.	Untuk mengembalikan cadangan IIS yang Anda buat dari server sumber di server target, gunakan langkah-l angkah berikut:	Administrator IIS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Buat folder bernama `backup` di `C:\windows\system32\inetsrv` folder dan tempatkan folder cadangan sumber di dalamnya. 2. Buka command prompt dan jalankan sebagai administrator. 3. Jalankan perintah berikut: `cd C:\windows\system32\inetsrv` 4. Jalankan perintah berikut: `C:\windows\system32\inetsrv\appcmd.exe list backup` 5. Jalankan perintah berikut: `C:\windows\system32\inetsrv\appcmd.exe restore backup <sourcebackup>` 6. Jalankan perintah berikut: `cd c:\windows\Microsoft.NET\Framework64\v2.0.50727\` 7. Jalankan perintah berikut: `aspnet_regiis.exe -pi "iisConfigurationKey" C:\temp\iisConfKey.xml -exp`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. Jalankan perintah berikut: `aspnet_regiis.exe -pi "iisWasKey" C:\temp\iisWasKey.xml -exp` 9. Setel ulang IIS, dan buka Pengaturan lanjutan Identitas Kolam Aplikasi dan Situs Web untuk memasukkan kembali kata sandi.	

Sumber daya terkait

AWS dokumentasi

- Menginstal Agen AWS Replikasi (AWS Application Migration Service dokumentasi)

AWS Bimbingan Preskriptif

- Memigrasi VM lokal ke Amazon dengan menggunakan EC2 AWS Application Migration Service
- Menggunakan AMIs atau snapshot Amazon EBS untuk backup

Sumber daya Microsoft

- Identitas kumpulan aplikasi
- Dokumentasi IIS
- Dokumentasi IIS 8 appcmd.exe
- Dokumentasi IIS 10 appcmd.exe
- Alat Admin Kuat

Memigrasi database Microsoft SQL Server lokal ke Amazon EC2 menggunakan Layanan Migrasi Aplikasi

Senthil Ramasamy, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah untuk memigrasikan database Microsoft SQL Server dari pusat data lokal ke instans Amazon Elastic Compute Cloud (Amazon). EC2 Ini menggunakan AWS Application Migration Service (AWS MGN) untuk meng-host ulang database Anda menggunakan migrasi otomatis lift-and-shift. AWS MGN melakukan replikasi tingkat blok dari server basis data sumber Anda.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Database Microsoft SQL Server sumber di pusat data lokal

Batasan

- Bandwidth jaringan Anda mungkin terbatas antara pusat data lokal dan AWS.
- AWS MGN terbatas pada database yang di-host di server mandiri dengan penyimpanan khusus. Itu tidak mendukung migrasi sistem database berkerumun dan sistem database di mana tingkat perubahan melebihi throughput jaringan.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

- Semua versi database Microsoft SQL Server
- Sistem operasi Windows dan Linux yang mendukung AWS MGN

Arsitektur

Tumpukan teknologi sumber

Database Microsoft SQL Server lokal

Tumpukan teknologi target

Database Microsoft SQL Server pada instans Amazon EC2

Arsitektur target

Arsitektur ini menggunakan AWS MGN untuk mereplikasi data dari pusat data perusahaan lokal ke. AWS Diagram menunjukkan proses replikasi data, komunikasi API, dan fase pengujian dan pemotongan.

1. Replikasi data:
   - AWS MGN mereplikasi data dari pusat data perusahaan lokal ke AWS dan memulai replikasi perubahan yang sedang berlangsung.
   - Server replikasi di subnet pementasan menerima dan memproses data.
2. Komunikasi API:
   - Server replikasi terhubung ke titik akhir API AWS MGN, Amazon EC2, dan Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) melalui port TCP 443.
   - AWS MGN mengelola migrasi.
   - Amazon EC2 mengelola operasi instans.
3. Tes dan cutover:
   - Instans pengujian diluncurkan di subnet operasional menggunakan data yang direplikasi.
   - Setelah pengujian berhasil, AWS MGN membuat instance cutover untuk migrasi akhir.

Alat

- AWS Application Migration Service (AWS MGN) membantu Anda meng-host ulang (mengangkat dan menggeser) aplikasi ke aplikasi AWS Cloud tanpa perubahan dan dengan waktu henti minimal.
- Direct Connectmenghubungkan jaringan internal Anda ke lokasi Direct Connect melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke AWS layanan publik sambil melewati penyedia layanan internet di jalur jaringan Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Praktik terbaik

- Siapkan titik akhir regional API untuk AWS MGN EC2, Amazon, dan Amazon S3 di virtual private cloud (VPC) untuk melarang akses publik dari internet.
- Siapkan pengaturan peluncuran AWS MGN untuk meluncurkan server basis data target di subnet pribadi.
- Izinkan hanya port yang diperlukan dalam grup keamanan database.
- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Mengatur AWS MGN

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan AWS MGN.	Cari di AWS Application Migration Service dalam Konsol Manajemen AWS, dan memulai proses setup. Ini akan membuat template replikasi dan mengarahk an Anda ke halaman server Sumber konsol MGN. Saat Anda mengonfigurasi layanan MGN, pilih peran layanan dari daftar yang dihasilkan.	DBA, insinyur Migrasi
Tambahkan server sumber.	Tambahkan detail server database sumber lokal Anda, lalu tambahkan server.	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal agen AWS MGN di server sumber.	Unduh penginstal agen AWS MGN ke sistem lokal Anda, dan transfer penginstal ke server basis data sumber Anda. Untuk memvalida si hash penginstal, lihat Memvalidasi penginstal Agen AWS Replikasi yang diunduh untuk Windows 2012.	DBA, insinyur Migrasi

Instal agen AWS MGN pada mesin sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan kredensil IAM klien.	Sebelum Anda menginstal agen AWS MGN, buat AWS kredensi dengan membuat pengguna IAM baru dengan izin yang sesuai. Untuk informasi selengkap nya, lihat kebijakan terkelola AWS untuk AWS Applicati on Migration Service dan Membuat AWS kredensil yang diperlukan.	DBA, insinyur Migrasi
Instal agen di server sumber.	Instal agen pada mesin sumber yang menghosting database Microsoft SQL Server. Untuk informasi selengkapnya, lihat Menginstal Agen AWS Replikasi di server Windows.	Pemilik aplikasi, DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Berikan kredensi AWS berikut: • Wilayah AWS • AWS ID kunci akses • AWS kunci akses rahasia AWS Kredensi unik Anda memungkinkan agen AWS MGN untuk mengautentikasi dan melakukan tugas migrasi.	
Pilih disk untuk direplikasi.	Setelah memasukkan AWS kredensil Anda, penginstal memverifikasi bahwa server Anda memenuhi persyarat an minimum untuk instalasi agen (misalnya, apakah server memiliki ruang disk yang cukup untuk menginstal agen MGN). AWS Pemasang menampilkan label volume dan detail penyimpanan. Untuk mereplikasi database Anda menggunakan layanan AWS MGN, pilih disk yang berlaku di server sumber Anda. Masukkan jalur setiap disk, dipisahkan dengan koma. Jika Anda ingin mereplikasi semua disk, biarkan jalurnya kosong. Setelah Anda mengkonfirmasi disk yang dipilih, instalasi dilanjutkan.	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pantau kemajuan sinkronisasi.	AWS Agen Replikasi memulai proses sinkronisasi dengan terlebih dahulu mengambil snapshot dari disk yang dipilih dan kemudian mereplikasi data. Anda dapat memantau kemajuan sinkronisasi dari halaman server Sumber di konsol AWS MGN. Untuk informasi selengkapnya, lihat Memantau server dalam siklus hidup migrasi.	DBA, insinyur Migrasi

Replikasi menggunakan AWS MGN

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kelola kemajuan replikasi.	Setelah Anda memulai sinkronisasi awal, server sumber Anda muncul di konsol AWS MGN, tempat Anda dapat mengelola dan memantau migrasi. Konsol menampilkan perkiraan waktu untuk replikasi lengkap, yang didasarkan pada ukuran total disk yang dipilih dan bandwidth jaringan yang tersedia.	DBA, insinyur Migrasi
Verifikasi sinkronisasi.	Setelah disk di server sumber disinkronkan sepenuhnya, verifikasi bahwa semua disk	Pemilik aplikasi, DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang dipilih terdaftar sebagai disinkronkan sepenuhnya dan tidak ada kesalahan yang dilaporkan di konsol. Konsol AWS MGN kemudian akan secara otomatis mentransisikan status siklus hidup migrasi ke Siap untuk pengujian, yang menunjukkan bahwa lingkungan yang direplikasi AWS disiapkan untuk pengujian kinerja dan fungsionalitas.	

Uji dan potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan peluncuran.	Pilih server sumber di konsol AWS MGN, dan perbarui pengaturan peluncuran untuk instance pengujian target. Dari halaman Detail server sumber, navigasikan ke tab Pengaturan peluncuran untuk mengonfigurasi instance pengujian. Pilih jenis instans yang hemat biaya dan jenis volume Amazon Elastic Block Store (Amazon EBS) Elastic Block Store (Amazon EBS), lalu konfigurasikan grup keamanan dan persyaratan jaringan.	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya, lihat Pengaturan peluncuran.	
Luncurkan instance uji target.	Arahkan ke konsol AWS MGN dari mesin sumber yang disinkronkan, dan luncurkan instance pengujian target dengan memilih Uji dan potong lalu Luncurkan instance pengujian. Ini menciptakan pekerjaan peluncuran yang menyebarkan instance pengujian menggunakan pengaturan yang dikonfigurasi. Instance diluncurkan di AWS Cloud dan mereplikasi lingkungan server database sumber Anda. Pantau kemajuan peluncuran dari halaman Riwayat peluncuran, tempat Anda dapat melacak pembuatan instans dan mengatasi masalah apa pun.	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi contoh uji target.	Validasi server EC2 database Amazon: 1. Pastikan konsol AWS MGN menunjukkan bahwa instance pengujian berjalan dengan sukses. 2. Gunakan klien RDP untuk masuk. 3. Dari menu Start, buka SQL Server Configuration Manager. 4. Verifikasi status layanan SQL Server, termasuk SQL Server (`MSSQLSERVER`) dan SQL Server Agent, memastikan mereka utuh dan diatur ke jenis startup yang benar. 5. Bandingkan pengaturan disk dan konfigurasi antara contoh pengujian dan server basis data sumber, konfirmasikan bahwa huruf drive, volume, alokasi disk, dan direktori yang diperlukan dipetakan dengan benar. 6. Connect ke SQL Server pada EC2 instance Amazon pengujian dan verifikasi bahwa semua database sumber telah bermigrasi dan muncul dalam daftar database.	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Melakukan tes validasi untuk memastikan fungsi database seperti yang diharapkan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ganti nama server.	AWS Migrasi MGN melibatkan salinan tingkat penyimpanan server sumber lokal Anda. EC2 Instance SQL Server Anda hanya berisi rincian server sumber asli dalam binari, jadi perbarui informasi biner untuk mencerminkan nama server baru.
	
	1. Gunakan SQL Server Management Studio (SSMS) untuk terhubung ke instance SQL Server Anda. EC2
	2. Periksa nama server:
	
	`SELECT @@SERVERNAME AS 'Current Server Name';`
	
	3. Ganti nama instance SQL Server dengan mengganti `NEW_SERVER_NAME` dengan nama server Anda:
	
	`EXEC sp_dropserver 'OLD_SERVER_NAME'`
	`EXEC sp_addserver 'NEW_SERVER_NAME', 'local';`
	
	4. Verifikasi bahwa nama server sudah benar:
		DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` SELECT @@SERVERNAME AS 'Updated Server Name'; ``` 5. Mulai ulang instance SQL Server.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan instance cutover.	Di konsol AWS MGN, di halaman Server sumber, konfirmasikan bahwa status siklus hidup migrasi server Siap untuk dipotong. Konfigura sikan setelan peluncura n untuk instance cutover, memastikan bahwa pengatura n mencerminkan lingkungan lokal Anda. Sebelum memulai cutover, matikan database lokal Anda, yang memastikan hal-hal berikut: • Semua transaksi yang sedang berlangsung selesai. • Tidak ada transaksi baru yang akan terjadi selama proses cutover. • Sinkronisasi data antara disk sumber dan target selesai. Memulai instance cutover di konsol AWS MGN. Saat instance cutover beroperas i, masuk ke instance dan lakukan pengujian berikut: 1. Pastikan SQL Server dimulai dengan benar dan database dapat diakses.	Pemilik aplikasi, DBA, insinyur Migrasi, Pimpinan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Validasi bahwa data Anda utuh dan konsisten dengan server sumber. 3. Jalankan pengujian aplikasi apa pun untuk mengonfir masi bahwa mereka melakukan seperti yang diharapkan. 4. Di konsol AWS MGN, atur status migrasi ke Cutover selesai. 5. Mulai routing lalu lintas ke EC2 instance.	

Pemecahan masalah

Isu	Solusi
Sinkronisasi awal gagal pada langkah otentikasi.	Ini adalah masalah konektivitas jaringan. Server replikasi tidak dapat terhubung ke AWS MGN.

Sumber daya terkait

Dokumentasi AWS

- Memulai dengan AWS Application Migration Service
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2
- Apa itu Microsoft SQL Server di Amazon? EC2

Video

- Melakukan Migrasi Lift dan Shift dengan AWS Application Migration Service (video)

Migrasikan beban kerja F5 BIG-IP ke F5 BIG-IP VE di AWS Cloud

Suresh Veeragoni dan Deepak Kumar, Amazon Web Services

Ringkasan

Organizations mencari untuk bermigrasi ke AWS Cloud untuk meningkatkan kelincahan dan ketahanan mereka. Setelah memigrasikan solusi keamanan dan manajemen lalu lintas F5 BIG-IP ke F5 AWS Cloud, Anda dapat fokus pada kelincahan dan adopsi model operasional bernilai tinggi di seluruh arsitektur perusahaan Anda.

Pola ini menjelaskan cara memigrasikan beban kerja F5 BIG-IP ke beban kerja F5 BIG-IP Virtual Edition (VE) pada file. AWS Cloud Beban kerja akan dimigrasikan dengan menghosting ulang lingkungan yang ada dan menerapkan aspek replatforming, seperti penemuan layanan dan integrasi API. AWS CloudFormation template mempercepat migrasi beban kerja Anda ke file. AWS Cloud

Pola ini ditujukan untuk tim teknik dan arsitektur teknis yang memigrasi solusi keamanan dan manajemen lalu lintas F5, dan menyertai panduan Migrasi dari F5 BIG-IP ke F5 BIG-IP VE di situs web Prescriptive Guidance. AWS Cloud AWS

Prasyarat dan batasan

Prasyarat

- Beban kerja F5 BIG-IP lokal yang ada.
- Lisensi F5 yang ada untuk versi BIG-IP VE.
- Aktif Akun AWS.
- Virtual private cloud (VPC) yang sudah ada dikonfigurasi dengan jalan keluar melalui gateway NAT atau alamat IP Elastic, dan dikonfigurasi dengan akses ke titik akhir berikut: Amazon Simple Storage Service (Amazon S3), Amazon Elastic Compute Cloud (Amazon), (), dan Amazon. EC2 AWS Security Token Service AWS STS CloudWatch Anda juga dapat memodifikasi arsitektur VPC Modular dan scalable Quick Start sebagai blok bangunan untuk penerapan Anda.
- Satu atau dua Availability Zone yang ada, tergantung pada kebutuhan Anda.
- Tiga subnet pribadi yang ada di setiap Availability Zone.
- AWS CloudFormation template, tersedia di GitHub repositori F5.

Selama migrasi, Anda juga dapat menggunakan yang berikut ini, tergantung pada kebutuhan Anda:

- Ekstensi Failover Cloud F5 untuk mengelola pemetaan alamat IP Elastis, pemetaan IP sekunder, dan perubahan tabel rute.
- Jika Anda menggunakan beberapa Availability Zone, Anda harus menggunakan F5 Cloud Failover Extensions untuk menangani pemetaan IP Elastis ke server virtual.
- Anda harus mempertimbangkan untuk menggunakan F5 Application Services 3 (AS3), F5 Application Services Templates (FAST), atau model infrastruktur lain sebagai kode (IAc) untuk mengelola konfigurasi. Mempersiapkan konfigurasi dalam model IAc dan menggunakan repositori kode akan membantu migrasi dan upaya manajemen Anda yang sedang berlangsung.

Keahlian

- Pola ini membutuhkan keakraban dengan bagaimana satu atau lebih VPCs dapat dihubungkan ke pusat data yang ada. Untuk informasi selengkapnya tentang ini, lihat opsi konektivitas Network-to-Amazon VPC di dokumentasi Amazon VPC.
- Keakraban juga diperlukan dengan produk dan modul F5, termasuk Traffic Management Operating System (TMOS), Local Traffic Manager (LTM), Global Traffic Manager (GTM), Access Policy Manager (APM), Application Security Manager (ASM), Advanced Firewall Manager (AFM),dan BIG-IQ.

Versi produk

- Kami menyarankan Anda menggunakan F5 BIG-IP versi 13.1 atau yang lebih baru, meskipun polanya mendukung F5 BIG-IP versi 12.1 atau yang lebih baru.

Arsitektur

Tumpukan teknologi sumber

- Beban kerja BIG-IP F5

Tumpukan teknologi target

- Amazon CloudFront
- CloudWatch
- Amazon EC2
- Amazon S3

- Amazon VPC
- AWS Global Accelerator
- AWS STS
- AWS Transit Gateway
- F5 IP BESAR

Arsitektur target

Alat

- CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- Amazon CloudFront mempercepat distribusi konten web Anda dengan mengirimkannya melalui jaringan pusat data di seluruh dunia, yang menurunkan latensi dan meningkatkan kinerja.
- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat menskalakannya ke atas atau ke bawah.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Security Token Service (AWS STS) membantu Anda meminta kredensil hak istimewa terbatas sementara untuk pengguna.
- AWS Transit Gatewayadalah hub pusat yang menghubungkan virtual private cloud (VPCs) dan jaringan lokal.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Epik

Penemuan dan penilaian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menilai kinerja F5 BIG-IP.	Kumpulkan dan catat metrik kinerja aplikasi di server virtual, dan metrik sistem yang akan dimigrasikan. Ini akan membantu mengukur AWS infrastruktur target dengan benar untuk optimalisasi biaya yang lebih baik.	F5 Arsitek, Insinyur dan Arsitek Jaringan, Insinyur
Evaluasi sistem operasi dan konfigurasi F5 BIG-IP.	Evaluasi objek mana yang akan dimigrasikan dan apakah struktur jaringan perlu dipertahankan, seperti VLANs.	F5 Arsitek, Insinyur
Evaluasi opsi lisensi F5.	Evaluasi model lisensi dan konsumsi mana yang akan Anda butuhkan. Penilaian ini harus didasarkan pada evaluasi Anda terhadap sistem operasi dan konfigurasi F5 BIG-IP.	F5 Arsitek, Insinyur
Mengevaluasi aplikasi publik.	Tentukan aplikasi mana yang memerlukan alamat IP publik. Sejajarkan aplikasi tersebut dengan instance dan cluster yang diperlukan untuk memenuhi persyaratan kinerja dan perjanjian tingkat layanan (SLA).	Arsitek F5, Arsitek Cloud, Arsitek Jaringan, Insinyur, Tim Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengevaluasi aplikasi internal.	Evaluasi aplikasi mana yang akan digunakan oleh pengguna internal. Pastikan Anda tahu di mana pengguna internal tersebut duduk di organisasi dan bagaimana lingkungan tersebut terhubung ke organisasi AWS Cloud. Anda juga harus memastikan aplikasi tersebut dapat menggunakan sistem nama domain (DNS) sebagai bagian dari domain default.	Arsitek F5, Arsitek Cloud, Arsitek Jaringan, Insinyur, Tim Aplikasi
Selesaikan AMI.	Tidak semua versi F5 BIG-IP dibuat sebagai Amazon Machine Images (). AMIs Anda dapat menggunakan F5 BIG-IP Image Generator Tool jika Anda memiliki versi teknik perbaikan cepat (QFE) khusus yang diperlukan. Untuk informasi selengkapnya tentang alat ini, lihat bagian “Sumber daya terkait”.	Arsitek F5, Arsitek Cloud, Insinyur
Selesaikan jenis dan arsitektur instance.	Tentukan jenis instance, arsitektur VPC, dan arsitektur yang saling berhubungan.	Arsitek F5, Arsitek Cloud, Arsitek Jaringan, Insinyur

Kegiatan keamanan dan kepatuhan yang lengkap

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dokumentasikan kebijakan keamanan F5 yang ada.	Kumpulkan dan dokumenta sikan kebijakan keamanan F5 yang ada. Pastikan Anda membuat salinannya di repositori kode aman.	F5 Arsitek, Insinyur
Enkripsi AMI.	(Opsional) Organisasi Anda mungkin memerlukan enkripsi data saat istirahat. Untuk informasi selengkapnya tentang membuat gambar Bring Your Own License (BYOL) kustom, lihat bagian “Sumber daya terkait”.	Arsitek F5, Insinyur Arsitek Cloud, Insinyur
Keraskan perangkat.	Ini akan membantu melindungi dari potensi kerentanan.	F5 Arsitek, Insinyur

Konfigurasikan AWS lingkungan baru Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat akun tepi dan keamanan.	Masuk ke Konsol Manajemen AWS dan buat Akun AWS yang akan menyediakan dan mengoperasikan layanan tepi dan keamanan. Akun ini mungkin berbeda dari akun yang beroperasi VPCs untuk layanan dan aplikasi bersama. Langkah ini dapat diselesaikan	Arsitek Awan, Insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sebagai bagian dari landing zone.	
Menyebarkan tepi dan keamanan VPCs.	Siapkan dan konfigurasikan VPCs yang diperlukan untuk memberikan layanan tepi dan keamanan.	Arsitek Awan, Insinyur
Connect ke pusat data sumber.	Connect ke pusat data sumber yang menampung beban kerja F5 BIG-IP Anda.	Arsitek Cloud, Arsitek Jaringan, Insinyur
Menyebarkan koneksi VPC.	Hubungkan tepi dan layanan keamanan VPCs ke aplikasi VPCs.	Arsitek Jaringan, Insinyur
Menyebarkan instance.	Terapkan instance dengan menggunakan CloudForm ation templat dari bagian “Sumber daya terkait”.	F5 Arsitek, Insinyur
Uji dan konfigurasikan failover instance.	Pastikan bahwa template AWS Advanced HA iApp atau F5 Cloud Failover Extension dikonfigurasi dan beroperasi dengan benar.	F5 Arsitek, Insinyur

Mengkonfigurasi jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan topologi VPC.	Buka konsol VPC Amazon dan pastikan VPC Anda memiliki semua subnet dan perlindun	Arsitek Jaringan, Arsitek F5, Arsitek Cloud, Insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	gan yang diperlukan untuk penerapan F5 BIG-IP VE.	
Siapkan titik akhir VPC Anda.	Siapkan titik akhir VPC untuk Amazon, Amazon EC2 S3, dan AWS STS jika beban kerja F5 BIG-IP tidak memiliki akses ke NAT Gateway atau alamat IP Elastis pada antarmuka TMM.	Arsitek Awan, Insinyur

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan konfigurasi.	Migrasikan konfigurasi F5 BIG-IP ke F5 BIG-IP VE di file. AWS Cloud	F5 Arsitek, Insinyur
Kaitkan yang sekunder IPs.	Alamat IP server virtual memiliki hubungan dengan alamat IP sekunder yang ditetapkan untuk instance. Tetapkan alamat IP sekunder dan pastikan “Izinkan pemetaan/penugasan ulang” dipilih.	F5 Arsitek, Insinyur

Konfigurasi uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi konfigurasi server virtual.	Uji server virtual.	Arsitek F5, Tim Aplikasi

Selesaikan operasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat strategi cadangan.	Sistem harus dimatikan untuk membuat snapshot penuh. Untuk informasi lebih lanjut, lihat “Memperbarui mesin virtual F5 BIG-IP” di bagian “Sumber daya terkait”.	Arsitek F5, Arsitek Cloud, Insinyur
Buat runbook failover cluster.	Pastikan proses runbook failover selesai.	F5 Arsitek, Insinyur
Mengatur dan memvalidasi logging.	Konfigurasikan Streaming Telemetri F5 untuk mengirim log ke tujuan yang diperlukan.	F5 Arsitek, Insinyur

Selesaikan cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Potong ke penyebaran baru.		Arsitek F5, Arsitek Cloud, Arsitek Jaringan, Insinyur, AppTeams

Sumber daya terkait

Panduan migrasi

- Migrasi dari F5 BIG-IP ke F5 BIG-IP VE di AWS Cloud

Sumber daya F5

- CloudFormation template dalam repositori F5 GitHub
- F5 di AWS Marketplace

- Ikhtisar F5 BIG-IP VE
- Contoh Quickstart - Edisi Virtual BIG-IP dengan WAF (LTM+ASM)
- Layanan aplikasi F5 pada AWS: ikhtisar (video)
- Layanan Aplikasi F5 3 Panduan Pengguna Ekstensi
- Dokumentasi cloud F5
- F5 iControl REST wiki
- F5 Ikhtisar file konfigurasi tunggal (11.x - 15.x)
- Whitepaper F5
- Alat Generator Gambar F5 BIG-IP
- Memperbarui mesin virtual F5 BIG-IP VE
- Ikhtisar opsi “platform-migrate” arsip UCS

Memigrasikan aplikasi web Go lokal ke AWS Elastic Beanstalk menggunakan metode biner

Suhas Basavaraj dan Shumaz Mukhtar Kazi, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan aplikasi web Go lokal ke AWS Elastic Beanstalk. Setelah aplikasi dimigrasikan, Elastic Beanstalk membuat biner untuk bundel sumber dan menerapkannya ke instance Amazon Elastic Compute Cloud (Amazon). EC2

Sebagai strategi migrasi rehost, pendekatan pola ini cepat dan tidak memerlukan perubahan kode, yang berarti lebih sedikit waktu pengujian dan migrasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Aplikasi web Go lokal.
- GitHub Repositori yang berisi kode sumber aplikasi Go Anda. Jika Anda tidak menggunakan GitHub, ada cara lain untuk membuat bundel sumber aplikasi untuk Elastic Beanstalk.

Versi produk

- Versi Go terbaru yang didukung oleh Elastic Beanstalk. Untuk informasi lebih lanjut, lihat dokumentasi Elastic Beanstalk.

Arsitektur

Tumpukan teknologi sumber

- Aplikasi web Go lokal

Tumpukan teknologi target

- AWS Elastic Beanstalk
- Amazon CloudWatch

Arsitektur target

Alat

- AWS Elastic Beanstalk dengan cepat menyebarkan dan mengelola aplikasi di AWS Cloud tanpa pengguna harus mempelajari infrastruktur yang menjalankan aplikasi tersebut. Elastic Beanstalk mengurangi kompleksitas manajemen tanpa membatasi pilihan atau kontrol.
- GitHubadalah sistem kontrol versi terdistribusi open-source.

Epik

Buat berkas bundel sumber aplikasi web Go .zip

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bundel sumber untuk aplikasi Go.	Buka GitHub repositori yang berisi kode sumber aplikasi Go Anda dan siapkan bundel sumber. Bundel sumber berisi file `application.go` sumber di direktori root, yang menghosting paket utama untuk aplikasi Go Anda. Jika Anda tidak menggunakan GitHub, lihat bagian Prasyarat sebelumnya dalam pola ini untuk cara lain untuk membuat bundel sumber aplikasi Anda.	Admin Sistem, Pengembang Aplikasi
Buat file konfigurasi.	Buat `.ebextensions` folder di bundel sumber Anda, lalu buat `options.config` file di dalam folder ini. Untuk informasi lebih lanjut, lihat dokumentasi Elastic Beanstalk .	Admin Sistem, Pengembang Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file bundel sumber.zip.	Jalankan perintah berikut. `git archive -o ../godemoapp.zip HEAD` Ini membuat file bundel sumber .zip. Unduh dan simpan file.zip sebagai file lokal. **Important** File.zip tidak boleh melebihi 512 MB dan tidak dapat menyertakan folder induk atau direktori tingkat atas.	Admin Sistem, Pengembang Aplikasi

Migrasikan aplikasi web Go ke Elastic Beanstalk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih aplikasi Elastic Beanstalk.	1. Masuk ke AWS Management Console dan buka konsol Elastic Beanstalk. 2. Dari daftar Wilayah, pilih Wilayah AWS Anda. 3. Di panel navigasi, pilih Aplikasi, lalu pilih aplikasi Elastic Beanstalk yang ada atau buat.	Admin Sistem, Pengembang Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk petunjuk tentang cara membuat aplikasi Elastic Beanstalk, lihat dokumentasi Elastic Beanstalk.	
Memulai lingkungan server web Elastic Beanstalk.	1. Pada halaman ikhtisar aplikasi, pilih Buat lingkungan baru, lalu pilih Lingkungan server Web. 2. Lengkapi bidang nama Lingkungan dan nama Domain. 3. Pilih versi Platform, dan pilih Go sebagai platform Anda.	Admin Sistem, Pengembang Aplikasi
Unggah file bundel sumber.zip ke Elastic Beanstalk.	1. Dalam Kode aplikasi, pilih Unggah kode Anda, lalu pilih File lokal. 2. Pilih file.zip yang berisi bundel sumber Anda. 3. Di label Versi, beri file nama unik, lalu pilih Buat lingkungan.	Admin Sistem, Pengembang Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aplikasi web Go yang digunakan.	Anda akan dialihkan ke halaman ikhtisar aplikasi Elastic Beanstalk. Di bagian atas ikhtisar, di samping Environment ID, pilih URL yang diakhiri `elasticbeanstalk.com` untuk menavigasi ke aplikasi Anda. Aplikasi Anda harus menggunakan nama ini dalam file konfigurasi sebagai variabel lingkungan dan menampilkannya di halaman web.	Admin Sistem, Pengembang Aplikasi

Pemecahan masalah

Isu	Solusi
Tidak dapat mengakses aplikasi melalui Application Load Balancer.	Periksa kelompok target yang berisi aplikasi Elastic Beanstalk Anda. Jika tidak sehat, masuk ke instance Elastic Beanstalk Anda dan `nginx.conf` periksa konfigurasi file untuk memverifikasi bahwa itu merutekan ke URL status kesehatan yang benar. Anda mungkin perlu mengubah URL pemeriksaan kesehatan grup target.

Sumber daya terkait

- Versi platform Go didukung oleh Elastic Beanstalk
- Menggunakan file konfigurasi dengan Elastic Beanstalk
- Membuat contoh aplikasi di Elastic Beanstalk

Memigrasi server SFTP lokal untuk menggunakan AWS AWS Transfer for SFTP

Akash Kumar, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan solusi transfer file lokal yang menggunakan Secure Shell (SSH) File Transfer Protocol (SFTP) ke menggunakan layanan. AWS Cloud AWS Transfer for SFTP Pengguna umumnya terhubung ke server SFTP baik melalui nama domainnya atau dengan IP tetap. Pola ini mencakup kedua kasus.

AWS Transfer for SFTP adalah anggota dari AWS Transfer Family. Ini adalah layanan transfer aman yang dapat Anda gunakan untuk mentransfer file masuk dan keluar dari layanan AWS penyimpanan melalui SFTP. Anda dapat menggunakannya AWS Transfer for SFTP dengan Amazon Simple Storage Service (Amazon S3) atau Amazon Elastic File System (Amazon EFS). Pola ini menggunakan Amazon S3 untuk penyimpanan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Nama domain SFTP yang ada atau IP SFTP tetap.

Batasan

- Objek terbesar yang dapat Anda transfer dalam satu permintaan saat ini 5 GiB. Untuk file yang lebih besar dari 100 MiB, pertimbangkan untuk menggunakan unggahan multipart Amazon S3.

Arsitektur

Tumpukan teknologi sumber

- File datar lokal atau file dump database.

Tumpukan teknologi target

- AWS Transfer for SFTP
- Amazon S3

- Amazon Virtual Private Cloud (Amazon VPC)
- AWS Identity and Access Management (IAM) peran dan kebijakan
- Alamat IP elastis
- Grup keamanan
- Amazon CloudWatch Log (opsional)

Arsitektur target

Otomatisasi dan skala

Untuk mengotomatiskan arsitektur target untuk pola ini, gunakan CloudFormation templat terlampir:

- `amazon-vpc-subnets.yml`menyediakan virtual private cloud (VPC) dengan dua subnet publik dan dua subnet pribadi.
- `amazon-sftp-server.yml`ketentuan server SFTP.
- `amazon-sftp-customer.yml`menambahkan pengguna.

Alat

Layanan AWS

- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Pola ini menggunakan Amazon S3 sebagai sistem penyimpanan untuk transfer file.
- AWS Transfer for SFTPmembantu Anda mentransfer file masuk dan keluar dari layanan AWS penyimpanan melalui protokol SFTP.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Epik

Buat VPC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC dengan subnet.	Buka konsol Amazon VPC. Buat virtual private cloud (VPC) dengan dua subnet publik. (Subnet kedua menyediakan ketersediaan tinggi.) — atau — Anda dapat menerapkan CloudFormation template terlampir,`amazon-vpc-subnets.yml` , di CloudFormation konsol untuk mengotomatiskan tugas dalam epik ini.	Pengembang, Administrator sistem
Tambahkan gateway internet.	Menyediakan gateway internet dan melampirkannya ke VPC.	Pengembang, Administrator sistem
Migrasikan IP yang ada.	Lampirkan IP yang ada ke alamat IP Elastis. Anda dapat membuat alamat IP Elastis dari kumpulan alamat Anda dan menggunakannya.	Pengembang, Administrator sistem

Menyediakan server SFTP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server SFTP.	Buka konsol AWS Transfer Family. Ikuti petunjuk di Buat	Pengembang, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	titik akhir yang menghadap internet untuk server Anda dalam AWS Transfer Family dokumentasi untuk membuat server SFTP dengan titik akhir yang menghadap ke internet. Untuk tipe Endpoint, pilih VPC yang dihosting. Untuk Akses, pilih Internet Facing. Untuk VPC, pilih VPC yang Anda buat di epik sebelumnya. — atau — Anda dapat menerapkan CloudFormation template terlampir,`amazon-sftp-server.yml` , di CloudFormation konsol untuk mengotomatiskan tugas dalam epik ini.	
Migrasikan nama domain.	Lampirkan nama domain yang ada ke nama host kustom. Jika Anda menggunakan nama domain baru, gunakan alias DNS Amazon Route 53. Untuk nama domain yang ada, pilih DNS Lainnya. Untuk informasi selengkapnya, lihat Bekerja dengan nama host kustom dalam AWS Transfer Family dokumentasi.	Pengembang, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan peran CloudWatch logging.	(Opsional) jika Anda ingin mengaktifkan CloudWatch logging, buat `Transfer` peran dengan operasi CloudWatch Logs API `logs:CreateLogGroup logs:CreateLogStream` , `logs:DescribeLogStreams` , dan `logs:PutLogEvents` . Untuk informasi selengkapnya, lihat Aktivitas CloudWatch log dengan AWS Transfer Family dokumentasi.	Pengembang, administrator sistem
Simpan dan kirimkan.	Pilih Simpan. Untuk Tindakan, pilih Mulai dan tunggu server SFTP dibuat dengan status Online.	Pengembang, Administrator sistem

Petakan alamat IP Elastis ke server SFTP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan server sehingga Anda dapat mengubah pengaturan.	Di AWS Transfer Family konsol, pilih Server, lalu pilih server SFTP yang Anda buat. Untuk Tindakan, pilih Berhenti. Saat server offline, pilih Edit untuk mengubah pengaturannya.	Pengembang, administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih Availability Zones dan subnet.	Di bagian Availability Zones, pilih Availability Zones dan subnet untuk VPC Anda.	Pengembang, Administrator sistem
Tambahkan alamat IP Elastis.	Untuk IPv4 Alamat, pilih alamat IP Elastis untuk setiap subnet, lalu pilih Simpan.	Pengembang, Administrator sistem

Tambahkan pengguna

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM bagi pengguna untuk mengakses bucket S3.	Buat peran IAM untuk `Transfer` dan tambahkan `s3:ListBucket` `s3:GetBucketLocation` , dan `s3:PutObject` dengan nama bucket S3 sebagai sumber daya. Untuk informasi selengkapnya, lihat Membuat peran dan kebijakan IAM dalam AWS Transfer Family dokumentasi. — atau — Anda dapat menerapkan CloudFormation template terlampir,`amazon-sftp-customer.yml` , di CloudFormation konsol untuk mengotomatiskan tugas dalam epik ini.	Pengembang, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket S3 untuk aplikasi.	Pengembang, Administrator sistem
Buat folder opsional.	(Opsional) Jika Anda ingin menyimpan file untuk pengguna secara terpisah, di folder Amazon S3 tertentu, tambahkan folder yang sesuai.	Pengembang, Administrator sistem
Buat kunci publik SSH.	Untuk membuat key pair SSH, lihat Menghasilkan kunci SSH dalam dokumentasi. AWS Transfer Family	Pengembang, Administrator sistem
Tambah pengguna.	Di AWS Transfer Family konsol, pilih Server, pilih server SFTP yang Anda buat, lalu pilih Tambah pengguna. Untuk direktori Home, pilih bucket S3 yang Anda buat. Untuk kunci publik SSH, tentukan bagian kunci publik dari key pair SSH. Tambahkan pengguna untuk server SFTP, lalu pilih Tambah.	Pengembang, Administrator sistem

Uji server SFTP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui grup keamanan.	Di bagian Grup Keamanan server SFTP Anda, tambahkan IP mesin uji Anda	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk mendapatkan akses SFTP.	
Gunakan utilitas klien SFTP untuk menguji server.	Uji transfer file dengan menggunakan utilitas klien SFTP apa pun. Untuk daftar klien dan instruksi, lihat Mentransfer file menggunakan klien dalam AWS Transfer Family dokumentasi.	Developer

Sumber daya terkait

- AWS Transfer Family Panduan Pengguna
- Panduan Pengguna Amazon S3
- Alamat IP elastis dalam EC2 dokumentasi Amazon

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Memigrasikan VM lokal ke Amazon dengan EC2 menggunakan AWS Application Migration Service

Thanh Nguyen, Amazon Web Services

Ringkasan

Dalam hal migrasi aplikasi, organisasi dapat mengambil pendekatan berbeda untuk meng-host ulang (mengangkat dan menggeser) server aplikasi dari lingkungan lokal ke Amazon Web Services (AWS) Cloud. Salah satu caranya adalah dengan menyediakan instance Amazon Elastic Compute Cloud (Amazon EC2) baru dan kemudian menginstal dan mengonfigurasi aplikasi dari awal. Pendekatan lain adalah dengan menggunakan layanan migrasi asli pihak ketiga atau AWS untuk memigrasikan beberapa server secara bersamaan.

Pola ini menguraikan langkah-langkah untuk memigrasikan mesin virtual (VM) yang didukung ke EC2 instans Amazon di AWS Cloud dengan menggunakan AWS Application Migration Service. Anda dapat menggunakan pendekatan dalam pola ini untuk memigrasikan satu atau beberapa mesin virtual secara manual, satu per satu, atau secara otomatis dengan membuat skrip otomatisasi yang sesuai berdasarkan langkah-langkah yang diuraikan.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif di salah satu Wilayah AWS yang mendukung Layanan Migrasi Aplikasi
- Konektivitas jaringan antara server sumber dan EC2 server target melalui jaringan pribadi dengan menggunakan AWS Direct Connect atau jaringan pribadi virtual (VPN), atau melalui internet

Batasan

- Untuk daftar terbaru Wilayah yang didukung, lihat Wilayah AWS yang Didukung.
- Untuk daftar sistem operasi yang didukung, lihat bagian Sistem operasi yang didukung dan Umum Amazon EC2 FAQs.

Arsitektur

Tumpukan teknologi sumber

- Server fisik, virtual, atau yang dihosting cloud yang menjalankan sistem operasi yang didukung oleh Amazon EC2

Tumpukan teknologi target

- EC2 Instans Amazon yang menjalankan sistem operasi yang sama dengan sumber VM
- Amazon Elastic Block Store (Amazon EBS)

Arsitektur sumber dan target

Diagram berikut menunjukkan arsitektur tingkat tinggi dan komponen utama dari solusi. Di pusat data lokal, ada mesin virtual dengan disk lokal. Di AWS, terdapat area pementasan dengan server replikasi dan area sumber daya yang dimigrasi dengan EC2 instance untuk pengujian dan pemotongan. Kedua subnet berisi volume EBS.

1. Inisialisasi Layanan Migrasi Aplikasi AWS.
2. Siapkan konfigurasi dan pelaporan server area pementasan, termasuk sumber daya area pementasan.
3. Instal agen di server sumber, dan gunakan replikasi data tingkat blok berkelanjutan (dikompresi dan dienkripsi).
4. Otomatiskan orkestrasi dan konversi sistem untuk mempersingkat jendela cutover.

Arsitektur jaringan

Diagram berikut menunjukkan arsitektur tingkat tinggi dan komponen utama solusi dari perspektif jaringan, termasuk protokol dan port yang diperlukan untuk komunikasi antara komponen utama di pusat data lokal dan AWS.

Alat

- AWS Application Migration Service membantu Anda meng-host ulang (mengangkat dan memindahkan) aplikasi ke AWS Cloud tanpa perubahan dan dengan waktu henti minimal.

Praktik terbaik

- Jangan mengambil server sumber offline atau melakukan reboot sampai cutover ke EC2 instance target selesai.
- Memberikan kesempatan yang cukup bagi pengguna untuk melakukan pengujian penerimaan pengguna (UAT) pada server target untuk mengidentifikasi dan menyelesaikan masalah apa pun. Idealnya, pengujian ini harus dimulai setidaknya dua minggu sebelum cutover.
- Sering memantau status replikasi server pada konsol Layanan Migrasi Aplikasi untuk mengidentifikasi masalah sejak dini.
- Gunakan kredenal AWS Identity and Access Management (IAM) sementara untuk instalasi agen, bukan kredenal pengguna IAM permanen.

Epik

Menghasilkan kredensi AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM AWS Replication Agent.	Masuk dengan izin administratif ke akun AWS. Pada konsol AWS Identity and Access Management (IAM), buat peran IAM: 1. Di konsol IAM, pilih Peran. 2. Pilih Buat peran. 3. Pada halaman Pilih entitas tepercaya, di bagian Jenis entitas tepercaya, pilih Akun AWS. 4. Di bagian Akun AWS, pilih Akun ini (< account-id>). 5. Pilih Berikutnya. 6. Pada halaman Tambahkan izin, cari `AWSApplic`	Administrator AWS, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ationMigrationAgen tInstallationPolic` y kebijakan, pilih kotak centang di sebelah nama kebijakan. 7. Pilih Berikutnya. 8. Pada halaman Rincian peran, masukkan MGN_AGENT_INSTALLA TION_ROLE sebagai nama peran. 9. Verifikasi bahwa bidang sudah benar, lalu pilih Buat peran.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan kredensil keamanan sementara.	Pada mesin dengan AWS Command Line Interface (AWS CLI) diinstal, masuk dengan izin administratif. Atau sebagai alternatif (dalam Wilayah AWS yang didukung), di AWS Management Console, masuk dengan izin administr atif ke akun AWS, dan buka AWS CloudShell. Buat kredensi sementara dengan perintah berikut, ganti `<account-id>` dengan ID akun AWS. `aws sts assume-role --role-arn arn:aws:iam::<account-id>:role/MGN_Agent_Installation_Role --role-session-name mgn_installation_session_role` Dari output perintah, salin nilai untuk`AccessKeyId` , **`SecretAccessKey`** , dan **`SessionToken`** . Simpan di lokasi yang aman untuk digunakan nanti. ⚠ Important Kredensi sementara ini akan kedaluwarsa	Administrator AWS, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	setelah satu jam. Jika Anda membutuhkan kredensil setelah satu jam, ulangi langkah sebelumnya.	

Inisialisasi Layanan Migrasi Aplikasi dan buat template Pengaturan Replikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi layanan.	Di konsol, masuk dengan izin administratif ke akun AWS. Pilih Layanan Migrasi Aplikasi, lalu pilih Memulai.	Administrator AWS, Insinyur migrasi
Buat dan konfigurasikan template Pengaturan Replikasi .	1. Berikan detail konfigurasi berikut: a. Pilih subnet area pementasan. b. Pilih jenis instance server replikasi (secara `t3.small` default). c. Pilih jenis volume EBS (gp3 secara default). d. Pilih opsi enkripsi EBS. e. Pastikan kotak centang Selalu gunakan grup keamanan Layanan Migrasi Aplikasi dipilih. f. Pilih kotak centang Gunakan IP pribadi untuk replikasi data	Administrator AWS, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(VPN DirectConnect,, VPC peering) jika Anda menggunakan konektivitas jaringan pribadi antara lingkungan lokal dan AWS. g. Pilih kotak centang bandwidth jaringan Throttle (per server - dalam Mbps) jika Anda ingin membatasi bandwidth jaringan untuk Layanan Migrasi Aplikasi. 2. Pilih Buat templat. Layanan Migrasi Aplikasi akan secara otomatis membuat semua peran IAM yang diperlukan untuk memfasilitasi replikasi data dan peluncuran server yang dimigrasi.	

Instal Agen Replikasi AWS di mesin sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kredensi AWS yang diperlukan.	Ketika Anda menjalankan file installer di server sumber, Anda harus memasukkan kredensi sementara yang Anda buat sebelumnya, termasuk`AccessKeyId` ,,	Insinyur migrasi, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan. `SecretAccessKey` `SessionToken`	
Untuk server Linux, instal agen.	Salin perintah penginstal, masuk ke server sumber Anda, dan jalankan penginstal. Untuk petunjuk terperinci, lihat dokumentasi AWS.	Administrator AWS, Insinyur migrasi
Untuk server Windows, instal agen.	Unduh file installer ke setiap server, lalu jalankan perintah installer. Untuk petunjuk terperinci, lihat dokumentasi AWS.	Administrator AWS, Insinyur migrasi
Tunggu replikasi data awal selesai.	Ketika agen telah diinstal, server sumber akan muncul di konsol Layanan Migrasi Aplikasi, di bagian Server sumber. Tunggu sementara server mengalami replikasi data awal.	Administrator AWS, Insinyur migrasi

Mengonfigurasi pengaturan peluncuran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan detail server.	Pada konsol Layanan Migrasi Aplikasi, pilih bagian Server sumber, lalu pilih nama server dari daftar untuk mengakses detail server.	Administrator AWS, Insinyur migrasi
Konfigurasikan pengaturan peluncuran.	Pilih tab Pengaturan Luncurkan. Anda dapat	Administrator AWS, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengonfigurasi berbagai pengaturan, termasuk pengaturan peluncuran umum dan pengaturan templat EC2 peluncuran. Untuk petunjuk terperinci, lihat dokumentasi AWS.	

Lakukan tes

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji server sumber.	1. Pada konsol Layanan Migrasi Aplikasi, di bagian Server sumber, pastikan bahwa siklus hidup Migrasi server sumber Siap untuk pengujian dan status replikasi Data Sehat. 2. Pilih kotak centang di sebelah kiri setiap server sumber. 3. Pilih Test dan Cutover, lalu pilih Launch Test Instance. 4. Saat diminta, pilih Luncurkan. Server akan diluncurkan.	Administrator AWS, Insinyur migrasi
Verifikasi bahwa tes selesai dengan sukses.	Setelah server uji benar-ben ar diluncurkan, status Alerts pada halaman akan ditampilk	Administrator AWS, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	an Diluncurkan untuk setiap server.	
Uji server.	Lakukan pengujian terhadap server uji untuk memastikan bahwa itu berfungsi seperti yang diharapkan.	Administrator AWS, Insinyur migrasi

Jadwalkan dan lakukan cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jadwalkan jendela cutover.	Jadwalkan jangka waktu cutover yang sesuai dengan tim yang relevan.	Administrator AWS, Insinyur migrasi
Lakukan cutover.	1. Pada konsol Migrasi Aplikasi, pada halaman Server Sumber, pilih kotak centang di sebelah kiri setiap server sumber. 2. Pilih Test and Cutover, dan pilih Tandai sebagai 'Siap untuk cutover'. 3. Verifikasi bahwa setiap siklus hidup Migrasi server sumber Siap untuk dipotong. 4. Pilih Uji dan Cutover, lalu pilih Luncurkan instance cutover. 5. Saat diminta, pilih Luncurkan. Server akan diluncurkan.	Administrator AWS, Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Siklus hidup Migrasi server sumber akan berubah menjadi Cutover yang sedang berlangsung.	
Verifikasi bahwa cutover berhasil diselesaikan.	Setelah server cutover sepenuhnya diluncurkan, status Alerts pada halaman Server Sumber akan ditampilkan Diluncurkan untuk setiap server.	Administrator AWS, Insinyur migrasi
Uji server.	Lakukan pengujian terhadap server cutover untuk memastikan bahwa itu berfungsi seperti yang diharapkan.	Administrator AWS, Insinyur migrasi
Selesaikan cutover.	Pilih Uji dan Cutover, lalu pilih Finalize cutover untuk menyelesaikan proses migrasi.	Administrator AWS, Insinyur migrasi

Sumber daya terkait

- Layanan Migrasi Aplikasi AWS
- Panduan Pengguna Layanan Migrasi Aplikasi AWS

Migrasikan kumpulan data kecil dari lokasi ke Amazon S3 menggunakan AWS SFTP

Charles Gibson dan Sergiy Shevchenko, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan kumpulan data kecil (5 TB atau kurang) dari pusat data lokal ke Amazon Simple Storage Service (Amazon S3) dengan menggunakan (). AWS Transfer for SFTP AWS SFTP Data dapat berupa dump database atau file datar.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Direct Connect Tautan yang dibuat antara pusat data Anda dan AWS

Batasan

- File data harus kurang dari 5 TB. Untuk file di atas 5 TB, Anda dapat melakukan pengunggahan multipart ke Amazon S3 atau memilih metode transfer data lain.

Arsitektur

Tumpukan teknologi sumber

- File datar lokal atau dump database

Tumpukan teknologi target

- Amazon S3

Arsitektur sumber dan target

Alat

- AWS SFTP— Memungkinkan transfer file langsung masuk dan keluar dari Amazon S3 menggunakan Secure File Transfer Protocol (SFTP).

- AWS Direct Connect— Menetapkan koneksi jaringan khusus dari pusat data lokal Anda ke. AWS
- Titik akhir VPC — Memungkinkan Anda menghubungkan VPC secara pribadi ke layanan endpoint VPC yang didukung Layanan AWS dan didukung AWS PrivateLink oleh tanpa gateway internet, perangkat terjemahan alamat jaringan (NAT), koneksi VPN, atau koneksi. Direct Connect Instans dalam VPC tidak memerlukan alamat IP publik untuk berkomunikasi dengan sumber daya dalam layanan.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dokumentasikan persyaratan SFTP saat ini.		Pemilik aplikasi, SA
Identifikasi persyaratan otentikasi.	Persyaratan dapat mencakup otentikasi berbasis kunci, nama pengguna atau kata sandi, atau penyedia identitas (iDP).	Pemilik aplikasi, SA
Identifikasi persyaratan integrasi aplikasi.		Pemilik aplikasi
Identifikasi pengguna yang membutuhkan layanan.		Pemilik aplikasi
Tentukan nama DNS untuk endpoint server SFTP.		Jaringan
Tentukan strategi cadangan.		SA, DBA (jika data ditransfer)
Identifikasi migrasi aplikasi atau strategi cutover.		Pemilik aplikasi, SA, DBA

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat satu atau lebih virtual private cloud (VPCs) dan subnet di Anda Akun AWS.		Pemilik aplikasi, AMS
Buat grup keamanan dan daftar kontrol akses jaringan (ACL).		Keamanan, Jaringan, AMS
Buat bucket Amazon S3.		Pemilik aplikasi, AMS
Buat peran AWS Identity and Access Management (IAM).	Buat kebijakan IAM yang menyertakan izin AWS SFTP untuk mengaktifkan akses bucket Amazon S3 Anda. Kebijakan IAM ini menentuka n tingkat akses yang Anda berikan kepada pengguna SFTP. Buat kebijakan IAM lain untuk membangun hubungan kepercayaan dengan AWS SFTP.	Keamanan, AMS
Kaitkan domain terdaftar (opsional).	Jika Anda memiliki domain terdaftar sendiri, Anda dapat mengaitkannya dengan server SFTP. Anda dapat merutekan lalu lintas SFTP ke titik akhir server SFTP Anda dari domain atau dari subdomain.	Jaringan, AMS
Buat server SFTP.	Tentukan jenis penyedia identitas yang digunakan oleh	Pemilik aplikasi, AMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	layanan untuk mengauten tikasi pengguna Anda.	
Buka klien SFTP.	Buka klien SFTP dan konfigurasikan koneksi untuk menggunakan host titik akhir SFTP. AWS SFTP mendukung klien SFTP standar apa pun. Klien SFTP yang umum digunakan termasuk OpenSSH, WinSCP, Cyberduck, dan. FileZilla Anda bisa mendapatkan nama host server SFTP dari konsol. AWS SFTP	Pemilik aplikasi, AMS

Rencanakan dan uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rencanakan migrasi aplikasi.	Rencanakan perubahan konfigurasi aplikasi yang diperlukan, atur tanggal migrasi, dan tentukan jadwal pengujian.	Pemilik aplikasi, AMS
Uji infrastrukturnya.	Uji di lingkungan non-produksi.	Pemilik aplikasi, AMS

Sumber daya terkait

Referensi

- AWS Transfer for SFTP Panduan Pengguna
- AWS Direct Connect sumber daya
- Titik Akhir VPC

Tutorial dan video

- AWS Transfer for SFTP (video)
- AWS Transfer for SFTP panduan pengguna
- AWS Papan Tulis SA - Direct Connect (video)

Memigrasi database Oracle lokal ke Oracle di Amazon EC2

Baji Shaik dan Pankaj Choudhary, Amazon Web Services

Ringkasan

Pola ini memandu Anda melalui langkah-langkah untuk memigrasikan database Oracle lokal ke Oracle pada instans Amazon Elastic Compute Cloud (Amazon). EC2 Ini menjelaskan dua opsi untuk migrasi: menggunakan AWS Data Migration Service (AWS DMS) atau menggunakan alat Oracle asli seperti RMAN, impor/ekspor Pompa Data, ruang meja yang dapat diangkut, dan Oracle. GoldenGate

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Oracle sumber di pusat data lokal

Batasan

- Sistem operasi target (OS) harus didukung oleh Amazon EC2. Untuk daftar lengkap sistem yang didukung, lihat Amazon EC2 FAQs.

Versi produk

- Oracle versi 10.2 dan yang lebih baru (untuk versi 10.x), 11g dan hingga 12.2, dan 18c untuk edisi Enterprise, Standard, Standard One, dan Standard Two. Untuk daftar versi terbaru yang didukung oleh AWS DMS, lihat “Database EC2 instans lokal dan Amazon” di Sumber untuk Migrasi Data dalam dokumentasi AWS DMS.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal

Tumpukan teknologi target

- Sebuah instance database Oracle di Amazon EC2

Arsitektur target

Arsitektur migrasi data

Menggunakan AWS DMS:

Menggunakan alat Oracle asli:

Alat

- AWS DMS - AWS Database Migration Services (AWS DMS) mendukung beberapa jenis basis data sumber dan target. Untuk informasi tentang versi dan edisi database yang didukung, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif.
- Alat Oracle Asli - RMAN, Impor/ekspor Pompa Data, ruang meja yang dapat diangkut, Oracle GoldenGate

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi database sumber dan target.		DBA
Identifikasi versi OS target.		DBA, SysAdmin
Identifikasi persyaratan perangkat keras untuk instance server target berdasarkan daftar kompatibi litas Oracle dan persyaratan kapasitas.		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, SysAdmin
Identifikasi persyarat an jaringan (latensi dan bandwidth).		DBA, SysAdmin
Pilih jenis instans yang tepat berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.		DBA, SysAdmin
Identifikasi persyaratan keamanan network/host akses untuk basis data sumber dan target.		DBA, SysAdmin
Identifikasi daftar pengguna OS yang diperlukan untuk instalasi perangkat lunak Oracle.		DBA, SysAdmin
Unduh AWS Schema Conversion Tool (AWS SCT) dan driver.		DBA
Buat proyek AWS SCT untuk beban kerja, dan sambungkan ke database sumber.		DBA
Hasilkan file SQL untuk pembuatan objek (tabel, indeks, urutan, dll.).		DBA
Tentukan strategi cadangan.		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan persyaratan ketersediaan.		DBA
Identifikasi strategi migrasi/peralihan aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat virtual private cloud (VPC) dan subnet di akun AWS Anda.		SysAdmin
Buat grup keamanan dan daftar kontrol akses jaringan (ACLs).		SysAdmin
Konfigurasikan dan mulai EC2 instance.		SysAdmin

Instal perangkat lunak Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna dan grup OS yang diperlukan untuk perangkat lunak Oracle.		DBA, SysAdmin
Unduh versi perangkat lunak Oracle yang diperlukan.		
Instal perangkat lunak Oracle pada EC2 instance.		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat objek seperti tabel, kunci utama, tampilan, dan urutan dengan menggunakan skrip yang dihasilkan oleh AWS SCT.		DBA

Migrasikan data - opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan alat Oracle asli atau alat pihak ketiga untuk memigrasikan objek dan data database.	Alat Oracle termasuk impor/ekspor Pompa Data, RMAN, ruang meja yang dapat diangkut, dan. GoldenGate	DBA

Migrasikan data - opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan metode migrasi.		DBA
Buat instance replikasi di konsol AWS DMS.		DBA
Buat titik akhir sumber dan target.		DBA
Buat tugas replikasi.		DBA
Aktifkan pengambilan data perubahan (CDC) untuk menangkap perubahan untuk replikasi berkelanjutan.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan tugas replikasi dan pantau log.		DBA
Buat objek sekunder seperti indeks dan kunci asing saat beban penuh selesai.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi cutover/switch-over aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS Secrets Manager sementara.		DBA, SysAdmin
Meninjau dan memvalidasi dokumen proyek.		DBA, SysAdmin, Pemilik aplikasi
Kumpulkan metrik di sekitar waktu untuk bermigrasi,%		DBA, SysAdmin, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
manual vs. alat, penghematan biaya, dll.		
Tutup proyek dan berikan umpan balik.		

Sumber daya terkait

Referensi

- Strategi untuk Migrasi Database Oracle ke AWS
- Migrasi database Oracle ke AWS Cloud
- EC2 Situs Amazon
- Situs AWS DMS
- Posting blog AWS DMS
- EC2 Harga Amazon
- Melisensikan Perangkat Lunak Oracle di Lingkungan Cloud Computing

Tutorial dan video

- Memulai dengan Amazon EC2
- Memulai dengan AWS DMS
- Pengantar Amazon EC2 - Elastic Cloud Server & Hosting dengan AWS (video)

Memigrasikan database Oracle lokal ke Amazon menggunakan Oracle Data EC2 Pump

Navakanth Talluri, Amazon Web Services

Ringkasan

Saat memigrasikan database, Anda harus mempertimbangkan faktor-faktor seperti mesin dan versi basis data sumber dan target, alat dan layanan migrasi, dan periode waktu henti yang dapat diterima. Jika Anda memigrasikan database Oracle lokal ke Amazon Elastic Compute Cloud (Amazon EC2), Anda dapat menggunakan alat Oracle, seperti Oracle Data Pump dan Oracle Recovery Manager (RMAN). Untuk informasi selengkapnya tentang strategi, lihat Memigrasi database Oracle ke AWS Cloud.

Oracle Data Pump membantu Anda mengekstrak cadangan database yang logis dan konsisten dan mengembalikannya ke EC2 instance target. Pola ini menjelaskan cara memigrasikan database Oracle lokal ke EC2 instance dengan menggunakan Oracle Data Pump dan `NETWORK_LINK` parameternya, dengan waktu henti minimal. `NETWORK_LINK`Parameter memulai impor melalui link database. Klien Oracle Data Pump Import (impdp) pada EC2 instance target terhubung ke database sumber, mengambil data darinya, dan menulis data langsung ke database pada instance target. Tidak ada cadangan, atau dump, file yang digunakan dalam solusi ini.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Database Oracle lokal yang:
  - Bukankah database Oracle Real Application Clusters (RAC)
  - Bukankah database Oracle Automatic Storage Management (Oracle ASM)
  - Berada dalam mode baca-tulis.
- Anda telah membuat tautan AWS Direct Connect antara pusat data lokal dan AWS. Untuk informasi selengkapnya, lihat Membuat sambungan (dokumentasi Direct Connect).

Versi produk

- Oracle Database 10g rilis 1 (10.1) dan yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Server database Oracle mandiri (non-RAC dan non-ASM) di pusat data lokal

Tumpukan teknologi target

- Database Oracle yang berjalan di Amazon EC2

Arsitektur target

Pilar keandalan AWS Well-Architected Framework merekomendasikan pembuatan cadangan data untuk membantu menyediakan ketersediaan dan ketahanan yang tinggi. Untuk informasi selengkapnya, lihat Arsitektur untuk ketersediaan tinggi dalam Praktik Terbaik untuk Menjalankan Database Oracle di AWS. Pola ini mengatur basis data primer dan siaga pada EC2 instance dengan menggunakan Oracle Active Data Guard. Untuk ketersediaan tinggi, EC2 instance harus berada di Availability Zone yang berbeda. Namun, Availability Zone dapat berada di Wilayah AWS yang sama atau di Wilayah AWS yang berbeda.

Active Data Guard menyediakan akses read-only ke database siaga fisik dan menerapkan perubahan redo terus menerus dari database utama. Berdasarkan tujuan titik pemulihan (RPO) dan tujuan waktu pemulihan (RTO), Anda dapat memilih antara opsi transportasi redo sinkron dan asinkron.

Gambar berikut menunjukkan arsitektur target jika EC2 instance primer dan siaga berada di Wilayah AWS yang berbeda.

Arsitektur migrasi data

Setelah selesai menyiapkan arsitektur target, Anda menggunakan Oracle Data Pump untuk memigrasikan data dan skema lokal ke instance utama. EC2 Selama cutover, aplikasi tidak dapat mengakses database lokal atau database target. Anda mematikan aplikasi ini sampai mereka dapat terhubung ke database target baru pada EC2 instance utama.

Gambar berikut menunjukkan arsitektur selama migrasi data. Dalam contoh arsitektur ini, EC2 instans primer dan siaga berada di Wilayah AWS yang berbeda.

Alat

Layanan AWS

- AWS Direct Connect menautkan jaringan internal Anda ke lokasi Direct Connect melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke layanan AWS publik sambil melewati penyedia layanan internet di jalur jaringan Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

Alat dan layanan lainnya

- Oracle Active Data Guard membantu Anda membuat, memelihara, mengelola, dan memantau database siaga.
- Oracle Data Pump membantu Anda memindahkan data dan metadata dari satu database ke database lainnya dengan kecepatan tinggi.

Praktik terbaik

- Praktik Terbaik untuk Menjalankan Oracle Database di AWS
- Mengimpor data menggunakan NETWORK_LINK

Epik

Siapkan EC2 instans di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi konfigurasi perangkat keras sumber untuk host lokal dan parameter kernel.	Validasi konfigurasi lokal, termasuk ukuran penyimpanan, input/output operasi per detik (IOPS), dan CPU. Ini penting untuk lisensi Oracle, yang didasarkan pada core CPU.	DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat infrastruktur di AWS.	Buat virtual private cloud (VPCs), subnet pribadi, grup keamanan, daftar kontrol akses jaringan (ACLs), tabel rute, dan gateway internet. Untuk informasi selengkapnya, lihat berikut ini: • VPCs dan subnet • Tutorial: Buat VPC untuk digunakan dengan instance database	DBA, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan EC2 instance dengan menggunakan Active Data Guard.	Konfigurasikan EC2 instans AWS dengan menggunakan konfigurasi Active Data Guard, seperti yang dijelaskan dalam AWS Well-Architected Framework. Versi Oracle Database pada EC2 instance dapat berbeda dari versi lokal karena pola ini menggunakan backup logis. Perhatikan hal-hal berikut: • Letakkan database target dalam mode baca-tulis. • Pada database target, berikan detail Transparent Network Substrate (TNS) untuk database sumber. Untuk informasi lebih lanjut, lihat: • Memulai Database (dokumentasi Oracle) • Membuat dan Mengkonfigurasi Database Oracle (dokumentasi Oracle)	DBA, administrator sistem AWS

Migrasi database ke Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dblink ke database lokal dari instance. EC2	Buat link database (dblink) antara database Oracle	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pada EC2 instance dan database Oracle lokal. Untuk informasi selengkapnya, lihat Menggunakan Impor Tautan Jaringan untuk Memindahkan Data (dokumentasi Oracle).	
Verifikasi koneksi antara EC2 instans dan host lokal.	Gunakan dblink untuk mengonfirmasi bahwa koneksi antara EC2 instance dan database lokal berfungsi . Untuk petunjuk, lihat MEMBUAT TAUTAN DATABASE (dokumentasi Oracle).	DBA
Hentikan semua aplikasi yang terhubung ke database lokal.	Setelah downtime database disetujui, matikan semua aplikasi dan pekerjaan dependen yang terhubung ke database lokal Anda. Anda dapat melakukan ini baik dari aplikasi secara langsung atau dari database dengan menggunakan cron. Untuk informasi selengkapnya, lihat Menggunakan Utilitas Crontab untuk Menjadwalkan Tugas di Oracle Linux.	DBA, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jadwalkan pekerjaan migrasi data.	Pada host target, gunakan perintah `impdb` untuk menjadwalkan impor Pompa Data. Ini menghubungkan database target ke host lokal dan memulai migrasi data. Untuk informasi selengkapnya, lihat Impor Pompa Data dan NETWORK_LINK (dokumentasi Oracle).	DBA
Validasi migrasi data.	Validasi data adalah langkah penting. Untuk validasi data, Anda dapat menggunakan alat khusus atau alat Oracle, seperti kombinasi kueri dblink dan SQL.	DBA

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Letakkan database sumber dalam mode read-only.	Konfirmasikan bahwa aplikasi dimatikan dan tidak ada perubahan yang dilakukan pada database sumber. Buka database sumber dalam mode read-only. Ini membantu Anda menghindari transaksi terbuka. Untuk informasi selengkapnya, lihat `ALTER DATABASE` di Pernyataan SQL (dokumentasi Oracle).	DBA, DevOps insinyur, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi jumlah objek dan data.	Untuk memvalidasi data dan objek, gunakan alat kustom atau alat Oracle, seperti kombinasi kueri dblink dan SQL.	DBA, Pengembang aplikasi
Connect aplikasi ke database pada EC2 instance utama.	Ubah atribut koneksi aplikasi untuk menunjuk ke database baru yang Anda buat pada EC2 instance utama.	DBA, Pengembang aplikasi
Validasi kinerja aplikasi.	Mulai aplikasi. Validasi fungsionalitas dan kinerja aplikasi dengan menggunak an Automated Workload Repository (dokumentasi Oracle).	Pengembang aplikasi, DevOps insinyur, DBA

Sumber daya terkait

Referensi AWS

- Migrasi database Oracle ke AWS Cloud
- Amazon EC2 untuk Oracle
- Migrasi database Oracle yang besar ke AWS untuk lingkungan lintas platform
- VPCs dan subnet
- Tutorial: Buat VPC untuk digunakan dengan instance database

Referensi Oracle

- Konfigurasi Oracle Data Guard
- Impor Pompa Data

Migrasikan sistem RHEL BYOL ke instans AWS License-Included dengan menggunakan AWS MGN

Mike Kuznetsov, Amazon Web Services

Ringkasan

Saat memigrasikan beban kerja Anda ke AWS dengan menggunakan AWS Application Migration Service (AWS MGN), Anda mungkin harus mengangkat dan memindahkan (rehost) instans Red Hat Enterprise Linux (RHEL) Anda dan mengubah lisensi dari model Bring Your Own License (BYOL) default ke model AWS License Included (LI) selama migrasi. AWS MGN mendukung pendekatan skalabel yang menggunakan Amazon Machine Image (AMI). IDs Pola ini menjelaskan cara menyelesaikan perubahan lisensi pada server RHEL selama migrasi rehost dalam skala besar. Ini juga menjelaskan cara mengubah lisensi untuk sistem RHEL yang sudah berjalan di Amazon Elastic Compute Cloud (Amazon EC2).

Prasyarat dan batasan

Prasyarat

- Akses ke akun AWS target
- AWS MGN diinisialisasi di akun AWS target dan Wilayah untuk migrasi (tidak diperlukan jika Anda telah bermigrasi dari sistem lokal ke AWS)
- Server RHEL sumber dengan lisensi RHEL yang valid

Arsitektur

Pola ini mencakup dua skenario:

- Migrasi sistem dari lokasi langsung ke instans AWS LI dengan menggunakan AWS MGN. Untuk skenario ini, ikuti instruksi dalam epik pertama (Migrasi ke instans LI - opsi 1) dan epik ketiga.
- Mengubah model lisensi dari BYOL ke LI untuk sistem RHEL yang sebelumnya dimigrasi yang sudah berjalan di Amazon. EC2 Untuk skenario ini, ikuti instruksi dalam epik kedua (Migrasi ke instans LI - opsi 2) dan epik ketiga.

> **Note**
>
> Epik ketiga melibatkan konfigurasi ulang instans RHEL baru untuk menggunakan server Red Hat Update Infrastructure (RHUI) yang disediakan oleh AWS. Proses ini sama untuk kedua skenario.

Alat

Layanan AWS

- AWS Application Migration Service (AWS MGN) membantu Anda meng-host ulang (mengangkat dan memindahkan) aplikasi ke AWS Cloud tanpa perubahan dan dengan waktu henti minimal.

Epik

Migrasi ke instans LI - opsi 1 (untuk sistem RHEL lokal)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Temukan ID AMI instans RHEL AWS LI di Wilayah target.	Kunjungi AWS Marketplace atau gunakan EC2 konsol Amazon untuk menemukan ID AMI RHEL yang cocok dengan versi sistem sumber RHEL (misalnya, RHEL-7.7) , dan tuliskan ID AMI. Di EC2 konsol Amazon, Anda dapat memfilter AMIs dengan menggunakan salah satu istilah pencarian berikut: • Deskripsi = Disediakan oleh Red Hat, Inc. • Nama AMI = RHEL-7.7	Administrator awan
Konfigurasikan pengaturan peluncuran AWS MGN.	1. Di konsol AWS MGN, tambahkan sistem RHEL	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber: Instal AWS Replication Agent dan tambahkan server sumber dengan mengikuti instruksi dalam dokumentasi AWS MGN. 2. Pada halaman Source server, pilih sistem RHEL sumber, lalu pilih tab Launch settings. 3. Di bagian Pengaturan peluncuran umum, pilih Edit. Untuk menonaktifkan pemilihan otomatis dan menentukan jenis instans target secara manual, ubah ukuran kanan tipe Instance ke Tidak Ada, lalu pilih Simpan pengaturan. Ini memungkinkan Anda menggunakan jenis instans yang Anda konfigurasikan di template EC2 peluncura n Amazon Anda. Untuk informasi selengkapnya, lihat dokumentasi AWS MGN. 4. Di bagian Template EC2 Luncurkan, pilih Ubah. Dalam kotak dialog Tentang memodifikasi templat EC2 peluncuran, pilih Ubah lagi. Ini membuka EC2 konsol Amazon sehingga Anda	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dapat mengubah template untuk instance ini. 5. Tinjau pertimbangan utama dalam dokumentasi AWS MGN. **Note** Anda dapat mengabaikan peringatan untuk tidak memilih AMI Anda sendiri. 6. Di EC2 konsol Amazon, di templat peluncuran baru, ubah yang berikut ini: • Untuk AMI, tentukan ID AMI yang Anda identifik asi sebelumnya, atau cari RHEL- x dan tentukan versi yang Anda perlukan (misalnya, RHEL-7.7). • Untuk tipe Instance, tetapkan jenis instance target yang diinginkan. • Biarkan bagian berikut tidak berubah: Pasangan kunci (login), Pengatura n jaringan (kecuali jika Anda ingin menentukan subnet target dan grup keamanan), Penyimpan an, Tag sumber daya	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(kecuali jika Anda ingin menambahkan atau memodifikasi tag apa pun). • (Opsional) Di bagian Detail lanjutan, tentukan peran profil instans IAM, jika diperlukan untuk manajemen future oleh AWS Systems Manager. 7. Pilih Buat versi templat, lalu pilih tautan dalam pesan sukses untuk melihat templat peluncuran. 8. Pilih Tindakan, Setel versi default. Untuk versi Template, pilih versi terbaru (versi 2 untuk sistem baru), lalu pilih Set as default version. AWS MGN sekarang akan menggunakan versi template peluncuran ini untuk meluncurkan instance pengujian atau pemotongan. Untuk informasi selengkapnya, lihat dokumentasi AWS MGN.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi pengaturan.	1. Di konsol AWS MGN, di halaman Server sumber, pilih server sumber Anda, lalu pilih tab Pengaturan peluncuran. 2. Di bagian EC2 Launch Template, verifikasi bahwa parameter Instance type, Subnet, dan Security group disetel dengan benar. **Note** Bagian ini tidak menampilkan ID AMI yang Anda pilih. Untuk melihat ID, Anda dapat membuka EC2 konsol Amazon, tampilan Luncurkan Template, dan mencari ID templat yang ditampilkan di bagian ini.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan instans LI baru.	1. Saat sinkronisasi awal selesai, kolom siklus hidup Migrasi untuk server di halaman server Sumber konsol AWS MGN berubah menjadi Siap untuk pengujian. Untuk meluncurkan instance pengujian baru, pilih server sumber Anda, buka menu Test dan cutover, lalu pilih Launch test instance. Pilih Lihat detail pekerjaan untuk memantau status pekerjaan peluncuran. Untuk informasi selengkap nya, lihat dokumentasi AWS MGN. 2. Tunggu hingga pekerjaan peluncuran selesai, lalu buka halaman detail EC2 instance yang diluncurkan. Pilih tab Detail dan verifikasi bahwa bagian Detail instans berisi hal-hal berikut: • Detail platform: “Red Hat Enterprise Linux” • Nama AMI: Nama AMI yang Anda tentukan dalam template EC2 peluncuran 3. Potong instans LI baru dengan mengikuti instruksi	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dalam dokumentasi AWS MGN. 4. Konfigurasikan ulang instance baru untuk menggunakan server RHUI yang disediakan AWS dengan mengikuti langkah-langkah dalam epik terakhir.	

Migrasi ke instance LI - opsi 2 (untuk instance RHEL EC2 BYOL)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan instans RHEL BYOL Anda ke EC2 instans AWS LI.	Anda dapat mengalihkan sistem RHEL yang sebelumnya Anda migrasi ke AWS sebagai instans BYOL ke AWS LI dengan memindahkan disknya (volume Amazon Elastic Block Store) dan melampirkannya ke instans LI baru. Untuk beralih ini, ikuti langkah-langkah ini: 1. Luncurkan instance RHEL target baru dari RHEL LI AMI. Pastikan AMI yang Anda pilih: • Menggunakan versi RHEL yang sama dengan instans RHEL Anda saat ini. • Memiliki proses boot-up yang sama (BIOS atau	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	UEFI) dengan instance RHEL Anda saat ini. Misalnya, jika server sumber berbasis BIOS, gunakan AWS Marketplace RHEL AMI yang juga berbasis BIOS; untuk sistem berbasis UEFI, pilih AMI berbasis UEFI. 2. Hentikan kedua instance: instans LI baru dan instance sumber asli. 3. Lepaskan semua volume EBS (termasuk disk root) dari instance LI baru dan hapus. 4. Lepaskan semua volume EBS (termasuk disk root) dari instance sumber lama dan lampirkan ke instance LI baru. Simpan pemetaan volume yang sama ke perangkat. (Misalnya, volume EBS yang sebelumnya dilampirkan ke `/dev/sda` drive harus dilampirkan `/dev/sda` ke instance baru.) 5. Hapus instance sumber (sekarang tanpa disk). 6. Mulai contoh LI baru. Masuk ke instance dan konfigurasikan ulang untuk menggunakan server	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	RHUI yang disediakan AWS dengan mengikuti langkah-langkah dalam epik berikutnya.	

Konfigurasikan ulang RHEL OS untuk menggunakan RHUI yang disediakan AWS - kedua opsi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Deregister OS dari langganan dan lisensi Red Hat.	Setelah migrasi dan pemotongan yang berhasil, sistem RHEL harus dihapus dari langganan Red Hat untuk menghentikan penggunaan lisensi Red Hat dan menghindari penagihan ganda. Untuk menghapus RHEL OS dari langganan Red Hat, ikuti proses yang dijelaskan dalam dokumentasi Red Hat Subscription Management (RHSM). Gunakan perintah CLI: `subscription-manager unregister` Anda juga dapat menonaktifkan plugin pengelola langganan untuk berhenti memeriksa status langganan pada setiap panggilan yum. Untuk melakukan	Linux atau administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ini, edit file konfigurasi `/etc/yum/pluginconf.d/subscription-manager.conf` dan ubah parameter `enabled=1` menjadi `enabled=0` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ganti konfigurasi pembaruan lama (RHUI, jaringan Red Hat Satellite, repositori yum) dengan RHUI yang disediakan AWS.	Anda harus mengkonfigurasi ulang sistem RHEL yang dimigrasi untuk menggunakan server RHUI yang disediakan AWS. Ini memberi Anda akses ke server RHUI dalam Wilayah AWS tanpa memerlukan infrastruktur pembaruan eksternal. Perubahan tersebut melibatkan proses berikut: 1. Cadangkan konfigurasi yum yang ada. 2. Hapus konfigurasi dan paket RHUI (yum repositori) lama. 3. Tambahkan konfigurasi dan paket sertifikat RHUI yang disediakan AWS baru. Anda harus mengambilnya dari instans RHEL lain di AWS karena paket konfigurasi ini hanya tersedia di server RHUI yang disediakan AWS. Berikut adalah langkah-l angkah rinci dan perintah: 1. Cadangkan konfigurasi dan sertifikat yum yang ada dengan menyalin semua `/etc/yum*` dan `/etc/`	Linux atau administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`pki/*` folder ke lokasi cadangan. Contoh:
	`mkdir yum-backup`
	`cp -ra /etc/yum* /etc/pki ./yum-backup`
	`tar czf yum-backup.tgz ./yum-backup`
	2. Hapus konfigurasi dan paket RHUI lama:
	a. Temukan semua paket RHUI yang diinstal:
	`sudo rpm -qa | grep rhui`
	b. Hapus paket-paket ini:
	`sudo yum remove $(rpm -qa | grep rhui)`
	c. Hapus `/etc/yum/vars/releasever` file, jika ada.
	3. Tambahkan paket RHUI dan sertifikat AWS yang disediakan oleh AWS baru. Anda harus mengambil ini dari instans RHEL lain di AWS. Ada beberapa cara untuk melakukan ini. Misalnya, Anda dapat mengikuti instruksi yang diberikan dalam artikel Red Hat Knowledgebase:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	a. Luncurkan instans RHEL (RHEL-EC2) lain dari AWS Marketplace.
	b. Unduh dua paket dari contoh ini: paket konfigurasi klien RHUI terbaru dan sertifikat otoritas sertifikat (CA). Misalnya, jalankan perintah ini dari desktop Anda:
	`ssh RHEL-EC2 "sudo yumdownloader ca-certificates rh-amazon-rhui-client"`
	c. Salin paket dari EC2 instance RHEL- ke sistem migrasi baru. Contoh:
	`scp RHEL-EC2:rh-amazon-rhui-client\* RHEL-EC2:ca-certificates\* .`
	`ssh <migrated-instance> "mkdir /tmp/amazon"`
	`scp rh-amazon-rhui-client* ca-certificates* <migrated-instance>:/tmp/amazon`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	d. Instal paket konfigurasi RHUI dan CA baru pada instance yang dimigrasi: `ssh <migrated-instance> "sudo rpm -Uhv /tmp/amazon/*"`	

Validasi konfigurasi.	Pada instance yang dimigrasi target, verifikasi bahwa konfigurasi baru sudah benar:
	`sudo yum clean all`
	`sudo yum repolist`
		Linux atau administrator sistem

Sumber daya terkait

- Panduan Pengguna AWS Application Migration Service (AWS MGN)
- Dapatkan paket klien AWS RHUI yang mendukung IMDSv2 (artikel Red Hat Knowledgebase)
- Templat EC2 peluncuran Amazon (EC2 dokumentasi Amazon)

Memigrasikan database SAP ASE lokal ke Amazon EC2

Sergey Dmitriev dan Gergely Cserdi, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database SAP Adaptive Server Enterprise (ASE) dari host lokal ke instans Amazon Elastic Compute Cloud (Amazon). EC2 Pola ini mencakup penggunaan AWS Database Migration Service (AWS DMS) atau alat asli SAP ASE seperti ASE Cockpit, Sybase Central for ASE, dan DBA Cockpit untuk migrasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber SAP ASE di pusat data lokal

Keterbatasan

- Database sumber harus kurang dari 64 TB

Versi produk

- SAP ASE versi 15.x dan 16.x atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Database SAP ASE lokal

Tumpukan teknologi target

- Database SAP ASE pada sebuah instance EC2

Arsitektur migrasi database

Menggunakan AWS DMS:

Menggunakan alat SAP ASE asli:

Alat

- AWS DMS - AWS Data Migration Service (AWS DMS) mendukung beberapa basis data sumber dan target yang berbeda. Untuk informasi selengkapnya, lihat Sumber untuk Migrasi Data dan Target untuk Migrasi Data. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif.
- SAP ASE - Alat asli termasuk ASE Cockpit, Sybase Central untuk ASE, dan DBA Cockpit.

Epik

Menganalisis migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target.		DBA
Identifikasi versi OS target.		DBA, SysAdmin
Identifikasi persyaratan perangkat keras untuk instance server target berdasarkan daftar kompatibi litas SAP ASE dan persyarat an kapasitas.		DBA, SysAdmin
Identifikasi persyaratan untuk jenis dan kapasitas penyimpan an.		DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi persyaratan jaringan termasuk latensi dan bandwidth.		DBA, SysAdmin
Pilih jenis instans, kapasitas , fitur penyimpanan, dan fitur jaringan yang tepat.		DBA, SysAdmin
Identifikasi persyaratan keamanan akses jaringan dan host untuk basis data sumber dan target.		DBA, SysAdmin
Identifikasi daftar pengguna sistem operasi yang diperluka n untuk instalasi perangkat lunak SAP ASE.		DBA, SysAdmin
Tentukan strategi cadangan.		DBA
Tentukan persyaratan ketersediaan.		DBA
Identifikasi migrasi aplikasi dan strategi peralihan.		DBA, SysAdmin, Pemilik aplikasi

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat virtual private cloud (VPC) dan subnet.		SysAdmin
Buat grup keamanan dan daftar kontrol akses jaringan (ACL).		SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan mulai EC2 instance.		SysAdmin

Instal perangkat lunak

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna dan grup OS yang diperlukan agar perangkat lunak SAP ASE berfungsi.		DBA, SysAdmin
Unduh versi perangkat lunak SAP ASE yang diperlukan.		DBA, SysAdmin
Instal database SAP ASE, perangkat lunak server cadangan, dan perangkat lunak server replikasi pada EC2 instance dan kemudian konfigurasikan server.		DBA, SysAdmin

Migrasikan data - opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan objek dan data database dengan menggunakan alat SAP ASE asli atau alat pihak ketiga.	Lihat dokumentasi untuk SAP ASE atau alat pihak ketiga. Ini termasuk ASE Cockpit, Sybase Central untuk ASE, dan DBA Cockpit.	DBA

Migrasikan data - opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan menggunakan AWS DMS.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi cutover atau switchover aplikasi.		DBA, SysAdmin, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, SysAdmin
Validasi dan tinjau dokumen proyek.		DBA, SysAdmin, Pemilik aplikasi
Kumpulkan metrik di sekitar waktu untuk bermigrasi, persen penghematan biaya		DBA, SysAdmin, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
manual versus alat, dan sebagainya.		
Tutup proyek dan berikan umpan balik apa pun.		DBA, SysAdmin, Pemilik aplikasi

Sumber daya terkait

Referensi

- Amazon EC2
- AWS DMS
- EC2 Harga Amazon

Tutorial dan video

- Memulai dengan Amazon EC2
- Memulai AWS Database Migration Service
- Layanan Migrasi Data AWS (video)
- Pengantar Amazon EC2 - Elastic Cloud Server & Hosting dengan AWS (video)

Memigrasi database Microsoft SQL Server lokal ke Amazon EC2

Senthil Ramasamy, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Microsoft SQL Server lokal ke Microsoft SQL Server di instans Amazon Elastic Compute Cloud (Amazon). EC2 Ini mencakup dua opsi untuk migrasi: menggunakan AWS Database Migration Service (AWS DMS) atau menggunakan alat Microsoft SQL Server asli seperti backup dan restore, Copy Database Wizard, atau copy and attach database.

Prasyarat dan batasan

Prasyarat

- AWS Akun aktif
- Sistem operasi yang didukung oleh Amazon EC2 (untuk daftar lengkap versi sistem operasi yang didukung, lihat Amazon EC2 FAQs)
- Database sumber Microsoft SQL Server di pusat data lokal

Versi produk

- Untuk database EC2 instans lokal dan Amazon, AWS DMS mendukung:
  - SQL Server versi 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, dan 2019
  - Edisi Enterprise, Standard, Workgroup, Developer, dan Web
- Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Microsoft SQL Server sebagai Target. AWS DMS

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server lokal

Tumpukan teknologi target

- Database Microsoft SQL Server pada sebuah instance EC2

Arsitektur target

Arsitektur migrasi data

- Menggunakan AWS DMS

- Menggunakan alat SQL Server asli

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan data ke dan dari basis data komersial dan sumber terbuka yang banyak digunakan, termasuk Oracle, SQL Server, MySQL, dan PostgreSQL. Anda dapat menggunakannya AWS DMS untuk memigrasikan data ke AWS Cloud, antar instans lokal (melalui AWS Cloud penyiapan), atau antara kombinasi pengaturan cloud dan lokal.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.
- Alat Microsoft SQL Server asli termasuk backup dan restore, Copy Database Wizard, dan salin dan lampirkan database.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target.		DBA
Identifikasi versi sistem operasi target.		DBA, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi persyaratan perangkat keras untuk instance server target berdasarkan daftar kompatibi litas Microsoft SQL Server dan persyaratan kapasitas.		DBA, Administrator sistem
Identifikasi persyaratan penyimpanan untuk jenis dan kapasitas.		DBA, Administrator sistem
Identifikasi persyaratan jaringan, termasuk latensi dan bandwidth.		DBA, Administrator sistem
Pilih jenis EC2 instans berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.		DBA, Administrator sistem
Identifikasi persyaratan keamanan akses jaringan dan host untuk basis data sumber dan target.		DBA, Administrator sistem
Identifikasi daftar pengguna yang diperlukan untuk instalasi perangkat lunak Microsoft SQL Server.		DBA, Administrator sistem
Tentukan strategi cadangan.		DBA
Tentukan persyaratan ketersediaan.		DBA
Identifikasi migrasi aplikasi dan strategi cutover.		DBA, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat virtual private cloud (VPC) dan subnet.		Administrator sistem
Buat grup keamanan dan daftar kontrol akses jaringan (ACL).		Administrator sistem
Konfigurasikan dan mulai sebuah EC2 instance.		Administrator sistem

Instal perangkat lunak

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna dan grup yang diperlukan untuk perangkat lunak Microsoft SQL Server.		DBA, Administrator sistem
Unduh perangkat lunak Microsoft SQL Server.		DBA, Administrator sistem
Instal perangkat lunak Microsoft SQL Server pada EC2 instance dan konfigura sikan server.		DBA, Administrator sistem

Migrasikan data - opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan alat Microsoft SQL Server asli atau alat pihak	Alat termasuk backup dan restore, Copy Database Wizard, dan copy dan attach	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
ketiga untuk memigrasikan objek dan data database.	database. Untuk informasi selengkapnya, lihat panduan Memigrasi database Microsoft SQL Server ke. AWS Cloud	

Migrasikan data - opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan menggunakan AWS DMS.	Untuk informasi selengkap nya tentang penggunaan AWS DMS, lihat tautan di bagian Sumber daya terkait.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.	Gunakan AWS Schema Conversion Tool (AWS SCT) untuk menganalisis dan memodifikasi kode SQL yang disematkan dalam kode sumber aplikasi.	DBA, Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi peralihan aplikasi.		DBA, Pemilik aplikasi, Administrator sistem

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan semua AWS sumber daya sementara.	Sumber daya sementara termasuk contoh AWS DMS replikasi dan EC2 instance untuk AWS SCT.	DBA, Administrator sistem
Tinjau dan validasi dokumen proyek.		DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik di sekitar waktu untuk bermigrasi, persen penghematan biaya manual versus alat, dan sebagainya.		DBA, Pemilik aplikasi, Administrator sistem
Tutup proyek dan berikan umpan balik.		DBA, Pemilik aplikasi, Administrator sistem

Sumber daya terkait

Referensi

- Migrasi database Microsoft SQL Server ke AWS Cloud
- Amazon EC2
- Amazon EC2 FAQs
- EC2 Harga Amazon
- AWS Database Migration Service
- Produk Microsoft di AWS
- Lisensi Microsoft pada AWS
- Microsoft SQL Server aktif AWS

Tutorial dan video

- Memulai dengan Amazon EC2
- Memulai dengan AWS Database Migration Service
- Bergabunglah dengan EC2 instans Amazon ke Simple AD Active Directory Anda
- Bergabunglah dengan EC2 instans Amazon ke Direktori AWS Managed Microsoft AD Aktif Anda
- AWS Database Migration Service(video)
- Pengantar Amazon EC2 - Elastic Cloud Server & Hosting dengan AWS (video)

Menghosting ulang beban kerja lokal di AWS Cloud: daftar periksa migrasi

Srikanth Rangavajhala, Amazon Web Services

Ringkasan

Menghosting ulang beban kerja lokal di Amazon Web Services (AWS) Cloud melibatkan fase migrasi berikut: perencanaan, pra-penemuan, penemuan, pembuatan, pengujian, dan pemotongan. Pola ini menguraikan fase dan tugas terkait mereka. Tugas dijelaskan pada tingkat tinggi dan mendukung sekitar 75% dari semua beban kerja aplikasi. Anda dapat menerapkan tugas-tugas ini selama dua hingga tiga minggu dalam siklus sprint yang gesit.

Anda harus meninjau dan memeriksa tugas-tugas ini dengan tim migrasi dan konsultan Anda. Setelah peninjauan, Anda dapat mengumpulkan masukan, menghilangkan atau mengevaluasi kembali tugas yang diperlukan untuk memenuhi kebutuhan Anda, dan memodifikasi tugas lain untuk mendukung setidaknya 75% dari beban kerja aplikasi dalam portofolio Anda. Anda kemudian dapat menggunakan alat manajemen proyek tangkas seperti Atlassian Jira atau Rally Software untuk mengimpor tugas, menetapkannya ke sumber daya, dan melacak aktivitas migrasi Anda.

Pola ini mengasumsikan bahwa Anda menggunakan AWS Cloud Migration Factory untuk meng-host ulang beban kerja Anda, tetapi Anda dapat menggunakan alat migrasi pilihan Anda.

Amazon Macie dapat membantu mengidentifikasi data sensitif di basis pengetahuan Anda, disimpan sebagai sumber data, log pemanggilan model, dan penyimpanan prompt di bucket Amazon Simple Storage Service (Amazon S3). Untuk informasi lebih lanjut, lihat dokumentasi Macie.

Prasyarat dan batasan

Prasyarat

- Alat manajemen proyek untuk melacak tugas migrasi (misalnya, Atlassian Jira atau Rally Software)
- Alat migrasi untuk menghosting ulang beban kerja Anda di AWS (misalnya, Cloud Migration Factory)

Arsitektur

Platform sumber

- Tumpukan sumber lokal (termasuk teknologi, aplikasi, database, dan infrastruktur)

Platform target

- Tumpukan target AWS Cloud (termasuk teknologi, aplikasi, database, dan infrastruktur)

Arsitektur

Diagram berikut menggambarkan rehosting (menemukan dan memigrasikan server dari lingkungan sumber lokal ke AWS) dengan menggunakan Cloud Migration Factory dan AWS Application Migration Service.

Alat

- Anda dapat menggunakan alat migrasi dan manajemen proyek pilihan Anda.

Epik

Tahap perencanaan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rawat backlog pra-penemuan.	Lakukan sesi kerja perawatan backlog pra-penemuan dengan pemimpin departemen dan pemilik aplikasi.	Manajer proyek, pemimpin scrum Agile
Lakukan sesi kerja perencana an sprint.	Sebagai latihan pelingkupan, distribusikan aplikasi yang ingin Anda migrasi melintasi sprint dan gelombang.	Manajer proyek, pemimpin scrum Agile

Fase pra-penemuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan pengetahuan aplikasi.	Konfirmasikan dan dokumenta sikan pemilik aplikasi dan	Spesialis migrasi (pewawanc ara)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengetahuan mereka tentang aplikasi. Tentukan apakah ada orang lain untuk pertanyaan teknis.	
Tentukan persyaratan kepatuhan aplikasi.	Konfirmasikan dengan pemilik aplikasi bahwa aplikasi tidak harus mematuhi persyaratan untuk Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS), Sarbanes-Oxley Act (SOX), informasi identitas pribadi (PII), atau standar lainnya. Jika persyaratan kepatuhan ada, tim harus menyelesaikan pemeriksaan kepatuhan mereka di server yang akan dimigrasikan.	Spesialis migrasi (pewawancara)
Konfirmasikan persyaratan rilis produksi.	Konfirmasikan persyaratan untuk merilis aplikasi yang dimigrasi ke produksi (seperti tanggal rilis dan durasi waktu henti) dengan pemilik aplikasi atau kontak teknis.	Spesialis migrasi (pewawancara)
Dapatkan daftar server.	Dapatkan daftar server yang terkait dengan aplikasi yang ditargetkan.	Spesialis migrasi (pewawancara)
Dapatkan diagram logis yang menunjukkan keadaan saat ini.	Dapatkan diagram keadaan saat ini untuk aplikasi dari arsitek perusahaan atau pemilik aplikasi.	Spesialis migrasi (pewawancara)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat diagram logis yang menunjukkan status target.	Buat diagram logis aplikasi yang menunjukkan arsitektur target di AWS. Diagram ini harus menggambarkan server, konektivitas, dan faktor pemetaan.	Arsitek perusahaan, Pemilik bisnis
Dapatkan informasi server.	Kumpulkan informasi tentang server yang terkait dengan aplikasi, termasuk detail konfigurasinya.	Spesialis migrasi (pewawancara)
Tambahkan informasi server ke template penemuan.	Tambahkan informasi server terperinci ke template penemuan aplikasi (lihat `mobilize-application-questionnaire.xlsx` di lampiran untuk pola ini). Template ini mencakup semua keamanan terkait aplikasi, infrastruktur, sistem operasi, dan detail jaringan.	Spesialis migrasi (pewawancara)
Publikasikan templat penemuan aplikasi.	Bagikan templat penemuan aplikasi dengan pemilik aplikasi dan tim migrasi untuk akses dan penggunaan umum.	Spesialis migrasi (pewawancara)

Fase penemuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan daftar server.	Konfirmasikan daftar server dan tujuan masing-masing	Spesialis migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	server dengan pemilik aplikasi atau pemimpin teknis.	
Identifikasi dan tambahkan grup server.	Identifikasi grup server seperti server web atau server aplikasi, dan tambahkan informasi ini ke template penemuan aplikasi. Pilih tingkat aplikasi (web, aplikasi, database) yang harus dimiliki setiap server.	Spesialis migrasi
Isi template penemuan aplikasi.	Lengkapi detail template penemuan aplikasi dengan bantuan tim migrasi, tim aplikasi, dan AWS.	Spesialis migrasi
Tambahkan detail server yang hilang (middleware dan tim OS).	Mintalah tim middleware dan sistem operasi (OS) untuk meninjau template penemuan aplikasi dan menambahkan detail server yang hilang, termasuk informasi database.	Spesialis migrasi
Dapatkan peraturan inbound/ outbound lalu lintas (tim jaringan).	Mintalah tim jaringan untuk mendapatkan peraturan inbound/outbound lalu lintas untuk server sumber dan tujuan. Tim jaringan juga harus menambahk an aturan firewall yang ada, mengekspornya ke format grup keamanan, dan menambahkan penyeimbang beban yang ada ke template penemuan aplikasi.	Spesialis migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi penandaan yang diperlukan.	Tentukan persyaratan penandaan untuk aplikasi.	Spesialis migrasi
Buat detail permintaan firewall.	Tangkap dan filter aturan firewall yang diperlukan untuk berkomunikasi dengan aplikasi.	Spesialis migrasi, arsitek Solusi, Pimpinan jaringan
Perbarui jenis EC2 instance.	Perbarui jenis instans Amazon Elastic Compute Cloud (Amazon EC2) yang akan digunakan di lingkungan target, berdasarkan infrastru ktur dan persyaratan server.	Spesialis migrasi, arsitek Solusi, Pimpinan jaringan
Identifikasi diagram keadaan saat ini.	Identifikasi atau buat diagram yang menunjukkan keadaan aplikasi saat ini. Diagram ini akan digunakan dalam permintaan keamanan informasi (InfoSec).	Spesialis migrasi, arsitek Solusi
Selesaikan diagram keadaan masa depan.	Selesaikan diagram yang menunjukkan status future (target) untuk aplikasi. Diagram ini juga akan digunakan dalam InfoSec permintaan.	Spesialis migrasi, arsitek Solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat firewall atau permintaan layanan grup keamanan.	Buat firewall atau permintaa n layanan grup keamanan (untuk pengembangan/QA, pra-produksi, dan produksi) . Jika Anda menggunakan Cloud Migration Factory, sertakan port khusus replikasi jika belum dibuka.	Spesialis migrasi, arsitek Solusi, Pimpinan jaringan
Tinjau firewall atau permintaan grup keamanan (InfoSec tim).	Pada langkah ini, InfoSec tim meninjau dan menyetujui firewall atau permintaan grup keamanan yang dibuat pada langkah sebelumnya.	InfoSec insinyur, Spesialis migrasi
Menerapkan permintaan grup keamanan firewall (tim jaringan).	Setelah InfoSec tim menyetuju i permintaan firewall, tim jaringan menerapkan aturan inbound/outbound firewall yang diperlukan.	Spesialis migrasi, arsitek Solusi, Pimpinan jaringan

Fase build (ulangi untuk pengembangan/QA, pra-produksi, dan lingkungan produksi)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Impor data aplikasi dan server.	1. Verifikasi bahwa Anda masuk ke server eksekusi migrasi sebagai pengguna domain dengan izin administrator lokal di server sumber dalam lingkup. 2. Gunakan formulir intake migrasi untuk mengimpor atribut untuk server sumber	Spesialis migrasi, Administr ator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dalam lingkup. Untuk informasi tambahan, lihat Panduan Implementasi Pabrik Migrasi Cloud. Jika Anda tidak menggunakan Cloud Migration Factory, ikuti petunjuk untuk menyiapkan alat migrasi Anda.	
Periksa prasyarat untuk server sumber.	Connect dengan server sumber dalam lingkup untuk memverifikasi prasyarat seperti port TCP 1500, port TCP 443, ruang bebas volume root, versi.NET Framework , dan parameter lainnya. Ini diperlukan untuk replikasi. Untuk informasi tambahan, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud
Buat permintaan layanan untuk menginstal agen replikasi.	Buat permintaan layanan untuk menginstal agen replikasi pada server dalam lingkup untuk pengembangan/QA, pra-produksi, atau produksi.	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal agen replikasi.	Instal agen replikasi pada server sumber dalam lingkup pada mesin pengembangan/QA, pra-produksi, atau produksi. Untuk informasi tambahan, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud
Dorong skrip pasca-peluncuran.	Layanan Migrasi Aplikasi mendukung skrip pasca-peluncuran untuk membantu Anda mengotomatiskan aktivitas tingkat OS seperti menginstal atau menghapus instalasi perangkat lunak setelah Anda meluncurkan instance target. Langkah ini mendorong skrip pasca-peluncuran ke mesin Windows atau Linux, tergantung pada server yang diidentifikasi untuk migrasi. Untuk petunjuknya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi status replikasi.	Konfirmasikan status replikasi untuk server sumber dalam lingkup secara otomatis dengan menggunakan skrip yang disediakan. Skrip berulang setiap lima menit sampai status semua server sumber dalam gelombang yang diberikan berubah menjadi Healthy. Untuk petunjuknya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud
Buat pengguna admin.	Admin lokal atau pengguna sudo di mesin sumber mungkin diperlukan untuk memecahkan masalah apa pun setelah pemindahan migrasi dari server sumber dalam cakupan ke AWS. Tim migrasi menggunakan pengguna ini untuk masuk ke server target ketika server otentikasi (misalnya, server DC atau LDAP) tidak dapat dijangkau. Untuk petunjuk langkah ini, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi template peluncuran.	Validasi metadata server untuk memastikannya berhasil dan tidak memiliki data yang tidak valid. Langkah ini memvalidasi metadata pengujian dan cutover. Untuk petunjuknya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud

Fase uji (ulangi untuk pengembangan/QA, pra-produksi, dan lingkungan produksi)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat permintaan layanan.	Buat permintaan layanan untuk tim infrastruktur dan tim lain untuk melakukan pemotongan aplikasi ke instans pengembangan/QA, pra-produksi, atau produksi.	Spesialis migrasi, Administrator Cloud
Konfigurasikan penyeimbang beban (opsional).	Konfigurasikan penyeimbang beban yang diperlukan, seperti Application Load Balancer atau penyeimbang beban F5 dengan iRULES.	Spesialis migrasi, Administrator Cloud
Luncurkan instance untuk pengujian.	Luncurkan semua mesin target untuk gelombang tertentu di Layanan Migrasi Aplikasi dalam mode uji. Untuk informasi tambahan, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi status instans target.	Verifikasi status instance target dengan memeriksa proses bootup untuk semua server sumber dalam lingkup dalam gelombang yang sama. Mungkin diperlukan waktu hingga 30 menit untuk instance target untuk boot up. Anda dapat memeriksa status secara manual dengan masuk ke EC2 konsol Amazon, mencari nama server sumber, dan meninjau kolom pemeriksaan Status. Pemeriksaan status 2/2 yang dilewati menunjukkan bahwa instance tersebut sehat dari perspektif infrastruktur.	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah entri DNS.	Ubah entri Sistem Nama Domain (DNS). (Gunakan `resolv.conf` atau `host.conf` untuk lingkungan Microsoft Windows.) Konfigurasikan setiap EC2 instance untuk menunjuk ke alamat IP baru dari host ini. **Note** Pastikan tidak ada konflik DNS antara server lokal dan AWS Cloud. Langkah ini dan langkah-langkah berikut bersifat opsional, tergantung pada lingkungan tempat server di-host.	Spesialis migrasi, Administrator Cloud
Uji konektivitas ke host backend dari EC2 instance.	Periksa login dengan menggunakan kredensi domain untuk server yang dimigrasi.	Spesialis migrasi, Administrator Cloud
Perbarui catatan DNS A.	Perbarui catatan DNS A untuk setiap host untuk menunjuk ke alamat IP EC2 pribadi Amazon yang baru.	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui catatan DNS CNAME.	Perbarui catatan DNS CNAME untuk virtual IPs (nama penyeimbang beban) untuk menunjuk ke cluster untuk server web dan aplikasi.	Spesialis migrasi, Administrator Cloud
Uji aplikasi di lingkungan yang berlaku.	Masuk ke EC2 instance baru dan uji aplikasi di lingkungan pengembangan/QA, pra-produksi, dan produksi.	Spesialis migrasi, Administrator Cloud
Tandai sebagai siap untuk cutover.	Ketika pengujian selesai, ubah status server sumber untuk menunjukkan bahwa itu siap untuk cutover, sehingga pengguna dapat meluncurkan instance cutover. Untuk petunjuknya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud

Fase pemotongan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat rencana penyebaran produksi.	Buat rencana penyebaran produksi (termasuk rencana backout).	Spesialis migrasi, Administrator Cloud
Beri tahu tim operasi tentang waktu henti.	Beri tahu tim operasi tentang jadwal downtime untuk server. Beberapa tim mungkin memerlukan permintaan perubahan atau tiket	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	permintaan layanan (CR/SR) untuk pemberitahuan ini.	
Replikasi mesin produksi.	Replikasi mesin produksi dengan menggunakan Layanan Migrasi Aplikasi atau alat migrasi lainnya.	Spesialis migrasi, Administrator Cloud
Matikan server sumber dalam lingkup.	Setelah memverifikasi status replikasi server sumber, Anda dapat mematikan server sumber untuk menghentikan transaksi dari aplikasi klien ke server. Anda dapat mematikan server sumber di jendela cutover. Untuk informasi selengkapnya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Administrator awan
Luncurkan instance untuk cutover.	Luncurkan semua mesin target untuk gelombang tertentu di Layanan Migrasi Aplikasi dalam mode cutover. Untuk informasi selengkapnya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ambil contoh IPs target.	Ambil contoh IPs untuk target. Jika pembaruan DNS adalah proses manual di lingkungan Anda, Anda perlu mendapatkan alamat IP baru untuk semua instance target. Untuk informasi selengkapnya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud
Verifikasi koneksi server target.	Setelah Anda memperbarui catatan DNS, sambungkan ke instance target dengan nama host untuk memverifikasi koneksi. Untuk informasi selengkapnya, lihat Panduan Implementasi Pabrik Migrasi Cloud.	Spesialis migrasi, Administrator Cloud

Sumber daya terkait

- Cara bermigrasi
- Panduan Implementasi Pabrik AWS Cloud Migration
- Mengotomatiskan migrasi server skala besar dengan Cloud Migration Factory
- Panduan Pengguna Layanan Migrasi Aplikasi AWS
- AWS Migration Acceleration Program

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menyiapkan infrastruktur Multi-AZ untuk SQL Server Always On FCI dengan menggunakan Amazon FSx

Manish Garg, TVRLPhani Kumar Dadi, Nishad Mankar, dan RAJNEESH TYAGI, Amazon Web Services

Ringkasan

Jika Anda perlu memigrasi sejumlah besar Microsoft SQL Server Always On Failover Cluster Instances (FCIs) dengan cepat, pola ini dapat membantu Anda meminimalkan waktu penyediaan. Dengan menggunakan otomatisasi dan Amazon FSx untuk Windows File Server, ini mengurangi upaya manual, kesalahan buatan manusia, dan waktu yang diperlukan untuk menyebarkan sejumlah besar cluster.

Pola ini mengatur infrastruktur untuk SQL Server FCIs dalam penerapan Multi-Availability Zone (Multi-AZ) di Amazon Web Services (AWS). Penyediaan layanan AWS yang diperlukan untuk infrastruktur ini diotomatiskan dengan menggunakan templat CloudFormationAWS. Instalasi SQL Server dan pembuatan node cluster pada instans Amazon Elastic Compute Cloud EC2 (Amazon) dilakukan dengan menggunakan PowerShell perintah.

Solusi ini menggunakan sistem file Amazon Multi-AZ FSx untuk Windows yang sangat tersedia sebagai saksi bersama untuk menyimpan file database SQL Server. Sistem FSx file Amazon dan instance EC2 Windows yang menghosting SQL Server digabungkan ke domain AWS Directory Service untuk Microsoft Active Directory (AWS Managed Microsoft AD) yang sama.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Pengguna AWS dengan izin yang memadai untuk menyediakan sumber daya menggunakan templat AWS CloudFormation
- AWS Directory Service untuk Microsoft Active Directory
- Kredensil di AWS Secrets Manager untuk mengautentikasi ke AWS Managed Microsoft AD dalam pasangan nilai kunci:
  - `ADDomainName`: <Domain Name>
  - `ADDomainJoinUserName`: <Domain Username>

- `ADDomainJoinPassword`: <Domain User Password>
- `TargetOU`: <Target OU Value>

> **Note**
>
> Anda akan menggunakan nama kunci yang sama dalam otomatisasi AWS Systems Manager untuk aktivitas gabungan AWS Managed Microsoft AD.

- File media SQL Server untuk instalasi SQL Server dan layanan Windows atau akun domain dibuat, yang akan digunakan selama pembuatan cluster
- Virtual Private Cloud (VPC), dengan dua subnet publik di Availability Zones terpisah, dua subnet pribadi di Availability Zones, gateway internet, gateway NAT, asosiasi tabel rute, dan server lompat

Versi produk

- Windows Server 2012 R2 dan Microsoft SQL Server 2016

Arsitektur

Tumpukan teknologi sumber

- SQL Server lokal dengan FCIs menggunakan drive bersama

Tumpukan teknologi target

- EC2 Instans AWS
- Amazon FSx untuk Server File Windows
- Runbook AWS Systems Manager Automation
- Konfigurasi jaringan (VPC, subnet, gateway internet, gateway NAT, server lompat, grup keamanan)
- AWS Secrets Manager
- AWS Dikelola Microsoft AD
- Amazon EventBridge
- AWS Identity and Access Management (IAM)

Arsitektur target

Diagram berikut menunjukkan akun AWS dalam satu Wilayah AWS, dengan VPC yang mencakup dua Availability Zone, dua subnet publik dengan gateway NAT, server lompat di subnet publik pertama, dua subnet pribadi, masing-masing dengan EC2 instance untuk node SQL Server dalam grup keamanan node, dan sistem FSx file Amazon yang terhubung ke masing-masing node SQL Server. AWS Directory Service, Amazon EventBridge, AWS Secrets Manager, dan AWS Systems Manager juga disertakan.

Otomatisasi dan skala

- Anda dapat menggunakan AWS Systems Manager untuk bergabung dengan AWS Managed Microsoft AD dan melakukan penginstalan SQL Server.

Alat

Layanan AWS

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS Directory Service menyediakan beberapa cara untuk menggunakan Microsoft Active Directory (AD) dengan layanan AWS lainnya seperti Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS) untuk SQL Server, dan FSx Amazon untuk Windows File Server.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, fungsi AWS Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun AWS lainnya.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.

- AWS Systems Manager membantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan manajemen aplikasi dan sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola sumber daya AWS Anda dengan aman dalam skala besar.

Alat-alat lainnya

- PowerShelladalah program manajemen otomatisasi dan konfigurasi Microsoft yang berjalan di Windows, Linux, dan macOS. Pola ini menggunakan PowerShell skrip.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub aws-windows-failover-cluster-automation.

Praktik terbaik

- Peran IAM yang digunakan untuk menerapkan solusi ini harus mematuhi prinsip hak istimewa paling sedikit. Untuk informasi selengkapnya, lihat dokumentasi IAM.
- Ikuti praktik CloudFormation terbaik AWS.

Epik

Menyebarkan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation tumpukan Systems Manager.	1. Masuk ke akun AWS Anda, dan buka AWS Management Console. 2. Arahkan ke CloudFormation konsol, dan buat CloudFormation tumpukan Systems Manager dengan mengunggah `ssm.yaml` template. Berikan nilai untuk parameter berikut:	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• StateUnJoinAssociationLoggingBucketName— Berikan nama untuk bucket S3 yang akan dibuat template untuk tujuan logging. • SSMAssociationADUnjoinNama — Berikan nama untuk `AWS::SSM::Association` sumber daya. • SSMAutomationDocumentName— Berikan nama untuk runbook Systems Manager Automation. • EventBridgeName— Berikan nama untuk bus EventBridge acara. 3. Menyebarkan CloudFormation tumpukan Systems Manager dengan meluncurkan `ssm.yaml` CloudFormation template. Template akan membuat runboook Systems Manager Atomation yang dimulai saat EC2 instance baru dengan tag diluncurkan. `ADJoined: FSXADD` Runbook Otomasi akan menambahkan instance ke direktori AWS Managed Microsoft AD.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan tumpukan infrastruktur.	Setelah berhasil menyebarkan tumpukan Systems Manager, buat `infra` tumpukan, yang mencakup node EC2 instance, grup keamanan, sistem file Amazon FSx untuk Windows File Server, dan peran IAM. 1. Arahkan ke CloudForm ation konsol dan luncurkan `infra-cf.yaml` template. Untuk menerapka n tumpukan ini, parameter berikut diperlukan: • `ActiveDir ectoryId` — ID untuk AWS Managed Microsoft AD • `ADDnsIpAd dresses1` — Alamat IP DNS utama AWS Managed Microsoft AD • `ADDnsIpAd dresses2` — Alamat IP DNS sekunder AWS Managed Microsoft AD • `FSxSecurityGroupNa me` — Nama grup FSx keamanan Amazon • `FSxWindowsFileSyst emName` — Nama FSx drive Amazon	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `ImageID`— ID gambar dasar Windows 2012 R2 atau Amazon Machine Image (AMI) yang digunakan untuk membuat node instance SQL Server • `KeyPairName` — Pasangan kunci-nilai untuk dilampirkan ke node EC2 instance untuk akses • `Node1SecurityGroupName` — Nama grup keamanan simpul pertama • `Node2SecurityGroupName` — Nama grup keamanan simpul kedua • `OUSecretName` — Nama rahasia yang berisi informasi AWS Managed Microsoft AD • `PrivateSubnet1` — ID subnet pribadi pertama • `PrivateSubnet2` — ID subnet pribadi kedua • `SqlFSxFCIName` — Nama tag yang diterapkan ke node primer dan sekunder dan ke Amazon FSx.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `SqlFSxServerNetBIOSName1` — Nama node EC2 instance utama (maksimal 15 karakter) • `SqlFSxServerNetBIOSName2` — Nama node EC2 instance sekunder (maksimal 15 karakter) • `VPC`— ID VPC • `WorkloadInstanceType` — Jenis EC2 contoh Menyebarkan `infra` tumpukan. Stack akan membuat semua komponen infrastruktur yang diperlukan untuk mengatur Windows SQL Server FCI. 2. Setelah node EC2 instance diluncurkan, dokumen Systems Manager Automation akan dipanggil untuk menggabungkan instans ini ke AWS Managed Microsoft AD. Anda dapat melacak progres di halaman Otomasi konsol Systems Manager.	

Mengatur Windows SQL Server Selalu Di FCI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal alat Windows.	1. Masuk ke EC2 instance utama, yaitu node 1. Untuk menginstal fitur Windows (Active Directory dan FCI Tools), jalankan PowerShell skrip berikut. `Install-WindowsFeature -Name RSAT-AD-Powershell,Failover-Clustering -IncludeManagementTools Install-WindowsFeature -Name RSAT-Clustering,RSAT-ADDS-Tools,RSAT-AD-Powershell,RSAT-DHCP,RSAT-DNS-Server` 2. Masuk ke EC2 instance sekunder, yaitu node 2, dan jalankan skrip yang sama untuk mengaktifkan fitur pada node 2.	AWS DevOps, DevOps insinyur, DBA
Prestage objek komputer cluster di Active Directory Domain Services.	Untuk melakukan prestage objek nama cluster (CNO) di Active Directory Domain Services (AD DS) dan prestage objek komputer virtual (VCO) untuk peran berkerumun, ikuti instruksi dalam dokumentasi Windows Server.	AWS DevOps, DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat WSFC.	Untuk membuat cluster Windows Server Failover Clustering (WSFC), lakukan hal berikut: 1. Masuk ke EC2 instance utama, yaitu node 1. Untuk membuat berbagi FSx file Amazon dan memberikan akses penuh ke akun layanan AD yang terdaftar, jalankan kode berikut. `Invoke-Command -ComputerName "<FSx Windows Remote PowerShell Endpoint>" -ConfigurationName FSxRemoteAdmin -scriptblock { New-FSxSmbShare -Name "SQLDB" -Path "D:\share" -Description "SQL Databases Share" -ContinuouslyAvailable $true -FolderEnumerationMode AccessBased -EncryptData $true grant-fsxsmbshareaccess -name SQLDB -AccountName "<domain\user>" -accessRight Full }` Perintah ini juga akan membuat file share	AWS DevOps, DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	terus tersedia (CA), yang dioptimalkan untuk digunakan oleh Microsoft SQL Server. 2. Untuk membuat cluster failover pada instance utama (node 1), jalankan perintah berikut. `New-Cluster -Name <CNO Name> -Node <Node1 Name>, <Node2 Name> -StaticAddress <Node1 Secondary Private IP>, <Node2 Secondary Private IP>` Perintah ini membutuhkan parameter berikut: • `Name`— Nama cluster (CNO) • `Node`— Nama-nama node primer dan sekunder, masing-masing • `StaticAddress` — Alamat IP sekunder dari node primer dan sekunder, masing-masing **Important** Administrator domain atau pengguna biasa harus memiliki izin	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	administrator pada kedua node untuk membuat cluster Windows Server Failover Clustering (WSFC). Jika tidak, perintah sebelumnya akan gagal dan mengembalikan pesan,`You do not have administrator privileges on servers.` 3. Setelah cluster dibuat, jalankan perintah berikut untuk melampirkan saksi berbagi file. `Set-ClusterQuorum -FileShareWitness \ \<FSx Windows Remote PowerShell Endpoint> \share\witness`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal cluster failover SQL Server.	Setelah cluster WSFC diatur, instal cluster SQL Server pada instance utama (node1). 1. Di drive T pada kedua node, buat `tempdb` dan `log` folder. Folder digunakan dalam PowerShell perintah. 2. Setelah Anda menyalin file media SQL Server untuk instalasi SQL Server pada kedua node, jalankan PowerShell perintah berikut pada node 1 untuk menginstal SQL Server pada node 1. `D:\setup.exe /Q ` `/ACTION=InstallFailoverCluster ` `/IACCEPTSQLSERVERLICENSETERMS ` `/FEATURES="SQL,IS,BC,Conn" ` `/INSTALLSHAREDDIR="C:\Program Files\Microsoft SQL Server" ` `/INSTALLSHAREDWOWDIR="C:\Program Files (x86)\Microsoft SQL Server" ` `/RSINSTALLMODE="FilesOnlyMode" ` `/INSTANCEID="MSSQLSERVER" `	AWS DevOps, DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`/INSTANCENAME="MSSQLSERVER" ` `/FAILOVERCLUSTERGROUP="SQL Server (MSSQLSERVER)" ` `/FAILOVERCLUSTERIPADDRESSES="IPv4;<2nd Sec Private Ip node1>;Cluster Network 1;<subnet mask>" ` `/FAILOVERCLUSTERNETWORKNAME="<Fail over cluster Network Name>" ` `/INSTANCEDIR="C:\Program Files\Microsoft SQL Server" ` `/ENU="True" ` `/ERRORREPORTING=0 ` `/SQMREPORTING=0 ` `/SAPWD="<Domain User password>" ` `/SQLCOLLATION="SQL_Latin1_General_CP1_CI_AS" ` `/SQLSYSADMINACCOUNTS="<domain\username>" ` `/SQLSVCACCOUNT="<domain\username>" /SQLSVCPASSWORD="<Domain User password>" ` `/AGTSVCACCOUNT="<domain\username>" /AGTSVCPASSWORD="<Domain User password>" ` `/ISSVCACCOUNT="<domain\username>" /ISSVCPASSWORD="<Domain User password>" `	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`/FTSVCACCOUNT="NT Service\MSSQLFDLauncher" ` `/INSTALLSQLDATADIR="\\<FSX DNS name>\share\Program Files\Microsoft SQL Server" ` `/SQLUSERDBDIR="\\<FSX DNS name>\share\data" ` `/SQLUSERDBLOGDIR="\\<FSX DNS name>\share\log" ` `/SQLTEMPDBDIR="T:\tempdb" ` `/SQLTEMPDBLOGDIR="T:\log" ` `/SQLBACKUPDIR="\\<FSX DNS name>\share\SQLBackup" ` `/SkipRules=Cluster_VerifyForErrors ` `/INDICATEPROGRESS`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan simpul sekunder ke cluster.	Untuk menambahkan SQL Server ke node sekunder (node 2), jalankan PowerShell perintah berikut. `D:\setup.exe /Q ` `/ACTION=AddNode ` `/IACCEPTSQLSERVERLICENSETERMS ` `/INSTANCENAME="MSSQLSERVER" ` `/FAILOVERCLUSTERGROUP="SQL Server (MSSQLSERVER)" ` `/FAILOVERCLUSTERIPADDRESSES="IPv4;<2nd Sec Private Ip node2>;Cluster Network 2;<subnet mask>" ` `/FAILOVERCLUSTERNETWORKNAME="<Fail over cluster Network Name>" ` `/CONFIRMIPDEPENDENCYCHANGE=1 ` `/SQLSVCACCOUNT="<domain\username>" /SQLSVCPASSWORD="<Domain User password>" ` `/AGTSVCACCOUNT="domain\username>" /AGTSVCPASSWORD="<Domain User password>" ` `/FTSVCACCOUNT="NT Service\MSSQLFDLauncher" ` `/SkipRules=Cluster_VerifyForErrors `	AWS DevOps, DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`/INDICATEPROGRESS`	
Uji SQL Server FCI.	1. Pada instance Windows untuk salah satu node, di Administrative Tools, luncurkan Failover Cluster Manager. 2. Arahkan ke Node, dan konfirmasikan bahwa status node adalah Status Running. 3. Pilih Peran, buka menu konteks (klik kanan) untuk SQL Server (MSSQLSERVER), dan pilih Pindahkan dan Pilih Node. 4. Setelah pemilihan node, SQL Server harus berjalan pada node lain.	DBA, DevOps insinyur

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Untuk membersihkan sumber daya, gunakan proses penghapusan CloudFormation tumpukan AWS: 1. Buka CloudFormation konsol AWS. 2. Pada halaman Stacks, pilih `infra` tumpukan.	AWS DevOps, DBA, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tumpukan pasti sedang berjalan. 3. Di panel detail tumpukan, pilih Hapus. 4. Pilih Hapus tumpukan saat diminta. 5. Ulangi langkah 2-4 untuk `ssm` tumpukan. Setelah penghapusan tumpukan selesai, tumpukan akan berada dalam status. `DELETE_COMPLETE` Tumpukan dalam `DELETE_COMPLETE` status tidak ditampilkan di CloudFormation konsol secara default. Untuk menampilkan tumpukan yang dihapus, Anda harus mengubah filter tampilan tumpukan seperti yang dijelaskan dalam Melihat tumpukan yang dihapus di konsol CloudFormation AWS. Jika penghapusan gagal, tumpukan akan berada dalam status. `DELETE_FAILED` Untuk solusi, lihat Menghapus tumpukan gagal dalam CloudFormation dokumentasi.	

Pemecahan masalah

Isu	Solusi
Kegagalan CloudFormation template AWS	Jika CloudFormation template gagal selama penerapan, lakukan hal berikut: 1. Buka CloudFormation konsol AWS. 2. Pada halaman Stacks di CloudFormation konsol, pilih tumpukan. 3. Pilih Acara, dan periksa status tumpukan.
Kegagalan bergabung dengan Microsoft AD yang Dikelola AWS	Untuk memecahkan masalah bergabung, ikuti langkah-langkah berikut: 1. Buka konsol System Manager. 2. Pilih Wilayah penyebaran. 3. Di panel kiri, pilih Otomasi, dan temukan runbook Otomasi yang gagal. 4. Buka runbook Otomasi, dan periksa status Eksekusi dan langkah-langkah Eksekusi. 5. Selidiki detail langkah yang gagal untuk melihat kesalahan atau kegagalan yang tepat.

Sumber daya terkait

- Sederhanakan penerapan ketersediaan tinggi Microsoft SQL Server Anda menggunakan FSx Amazon untuk Windows File Server
- Menggunakan FSx untuk Windows File Server dengan Microsoft SQL Server

Gunakan kueri BMC Discovery untuk mengekstrak data migrasi untuk perencanaan migrasi

Ben Tailor-Hamblin, Emma Baldry, Simon Cunningham, dan Shabnam Khan, Amazon Web Services

Ringkasan

Panduan ini memberikan contoh kueri dan langkah-langkah untuk membantu Anda mengekstrak data dari infrastruktur dan aplikasi lokal menggunakan BMC Discovery. Pola ini menunjukkan cara menggunakan kueri BMC Discovery untuk memindai infrastruktur Anda dan mengekstrak informasi perangkat lunak, layanan, dan ketergantungan. Data yang diekstraksi diperlukan untuk menilai dan memobilisasi fase migrasi skala besar ke Amazon Web Services (AWS) Cloud. Anda dapat menggunakan data ini untuk membuat keputusan penting tentang aplikasi mana yang akan dimigrasi bersama sebagai bagian dari rencana migrasi Anda.

Prasyarat dan batasan

Prasyarat

- Lisensi untuk BMC Discovery (sebelumnya BMC ADDM) atau perangkat lunak sebagai layanan (SaaS) versi BMC Helix Discovery
- BMC Discovery versi lokal atau SaaS, diinstal

> **Note**
>
> Untuk BMC Discovery versi lokal, Anda harus menginstal aplikasi di jaringan klien dengan akses ke semua perangkat jaringan dan server yang berada dalam cakupan migrasi di beberapa pusat data. Akses ke jaringan klien harus disediakan sesuai dengan instruksi instalasi aplikasi. Jika pemindaian informasi Windows Server diperlukan, maka Anda harus mengatur perangkat manajer proxy Windows di jaringan.

- Akses jaringan untuk memungkinkan aplikasi memindai perangkat di seluruh pusat data, jika Anda menggunakan BMC Helix Discovery

Versi produk

- Penemuan BMC 22.2 (12,5)
- Penemuan BMC 22.1 (12.4)

- Penemuan BMC 21.3 (12.3)
- Penemuan BMC 21.05 (12.2)
- Penemuan BMC 20.08 (12.1)
- Penemuan BMC 20.02 (12.0)
- Penemuan BMC 11.3
- Penemuan BMC 11.2
- Penemuan BMC 11.1
- Penemuan BMC 11.0
- Penemuan Atrium BMC 10.2
- Penemuan Atrium BMC 10.1
- Penemuan Atrium BMC 10.0

Arsitektur

Diagram berikut menunjukkan bagaimana manajer aset dapat menggunakan kueri BMC Discovery untuk memindai aplikasi yang dimodelkan BMC di SaaS dan lingkungan lokal.

Diagram menunjukkan alur kerja berikut: Manajer aset menggunakan BMC Discovery atau BMC Helix Discovery untuk memindai database dan instance perangkat lunak yang berjalan di server virtual yang dihosting di beberapa server fisik. Alat ini dapat memodelkan aplikasi dengan komponen yang mencakup beberapa server virtual dan fisik.

Tumpukan teknologi

- Penemuan BMC
- Penemuan Helix BMC

Alat

- BMC Discovery adalah alat penemuan pusat data yang membantu Anda menemukan pusat data secara otomatis.
- BMC Helix Discovery adalah sistem pemodelan penemuan dan ketergantungan berbasis SaaS yang membantu Anda memodelkan aset data dan dependensinya secara dinamis.

Praktik terbaik

Ini adalah praktik terbaik untuk memetakan data aplikasi, ketergantungan, dan infrastruktur saat Anda bermigrasi ke cloud. Pemetaan membantu Anda memahami kompleksitas lingkungan Anda saat ini dan dependensi di antara berbagai komponen.

Informasi aset yang diberikan oleh kueri ini penting karena beberapa alasan:

1. Perencanaan — Memahami ketergantungan antar komponen membantu Anda merencanakan proses migrasi dengan lebih efektif. Misalnya, Anda mungkin perlu memigrasikan komponen tertentu terlebih dahulu untuk memastikan komponen lain berhasil dimigrasi.
2. Penilaian risiko — Memetakan dependensi antar komponen dapat membantu Anda mengidentifikasi potensi risiko atau masalah yang dapat timbul selama proses migrasi. Misalnya, Anda mungkin menemukan bahwa komponen tertentu bergantung pada teknologi usang atau tidak didukung yang dapat menyebabkan masalah di cloud.
3. Arsitektur cloud — Memetakan data aplikasi dan infrastruktur Anda juga dapat membantu Anda merancang arsitektur cloud yang sesuai yang memenuhi kebutuhan organisasi Anda. Misalnya, Anda mungkin perlu merancang arsitektur multi-tier untuk mendukung persyaratan ketersediaan atau skalabilitas yang tinggi.

Secara keseluruhan, pemetaan aplikasi, ketergantungan, dan data infrastruktur merupakan langkah penting dalam proses migrasi cloud. Latihan pemetaan dapat membantu Anda lebih memahami lingkungan Anda saat ini, mengidentifikasi potensi masalah atau risiko, dan merancang arsitektur cloud yang sesuai.

Epik

Identifikasi dan evaluasi alat penemuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi pemilik ITSM.	Identifikasi pemilik IT Service Management (ITSM) (biasanya dengan menghubungi tim pendukung operasional).	Pimpin migrasi
Periksa CMDB.	Mengidentifikasi jumlah database manajemen	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	konfigurasi (CMDBs) yang berisi informasi aset, dan kemudian mengidentifikasi sumber informasi tersebut.	
Identifikasi alat penemuan dan periksa penggunaan BMC Discovery.	Jika organisasi Anda menggunakan BMC Discovery untuk mengirim data tentang lingkungan Anda ke alat CMDB, periksa cakupan dan cakupan pemindaiannya. Misalnya, periksa apakah BMC Discovery memindai semua pusat data dan apakah server akses berada di zona perimeter.	Pimpin migrasi
Periksa tingkat pemodelan aplikasi.	Periksa apakah aplikasi dimodelkan dalam BMC Discovery. Jika tidak, rekomendasikan penggunaa n alat BMC Discovery untuk memodelkan instance perangkat lunak yang sedang berjalan yang menyediakan aplikasi dan layanan bisnis.	Insinyur migrasi, pimpinan Migrasi

Ekstrak data infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak data pada server fisik dan virtual.	Untuk mengekstrak data pada server fisik dan virtual yang dipindai oleh BMC Discovery,	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	gunakan Query Builder untuk menjalankan kueri berikut: `search Host show key as 'Serverid', virtual, name as 'HOSTNAME', os_type as 'osName', os_version as 'OS Version', num_logical_processors as 'Logical Processor Counts', cores_per_processor as 'Cores per Processor', logical_ram as 'Logical RAM', #Consumer:StorageUse:Provider:DiskDrive.size as 'Size'` **Note** Anda dapat menggunakan data yang diekstrak untuk menentukan ukuran instans yang sesuai untuk migrasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak data pada aplikasi yang dimodelkan.	Jika aplikasi Anda dimodelkan dalam BMC Discovery, Anda dapat mengekstrak data tentang server yang menjalankan perangkat lunak aplikasi. Untuk mendapatkan nama server, gunakan Query Builder untuk menjalankan query berikut: `search SoftwareInstance show key as 'ApplicationID', #RunningSoftware:HostedSoftware:Host:Host.key as 'ReferenceID', type, name` **Note** Aplikasi dimodelkan dalam BMC Discovery oleh kumpulan instance perangkat lunak yang berjalan. Aplikasi ini tergantung pada semua server yang menjalankan perangkat lunak aplikasi.	Pemilik aplikasi BMC Discovery

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak data pada database.	Untuk mendapatkan daftar semua database yang dipindai dan server basis data ini berjalan, gunakan Query Builder untuk menjalankan kueri berikut: `search Database show key as 'Key', name, type as 'Source Engine Type', #Detail:Detail:ElementWithDetail:SoftwareInstance.name as 'Software Instance', #Detail:Detail:ElementWithDetail:SoftwareInstance.product_version as 'Product Version', #Detail:Detail:ElementWithDetail:SoftwareInstance.edition as 'Edition', #Detail:Detail:ElementWithDetail:SoftwareInstance.#RunningSoftware:HostedSoftware:Host:Host.key as 'ServerID'`	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak data pada komunikasi server.	Untuk mendapatkan informasi tentang semua komunikasi jaringan antar server yang dikumpulkan oleh BMC Discovery dari log komunikasi jaringan bersejarah, gunakan Query Builder untuk menjalankan kueri berikut: `search Host TRAVERSE InferredElement:Inference:Associate:DiscoveryAccess TRAVERSE DiscoveryAccess:DiscoveryAccessResult:DiscoveryResult:NetworkConnectionList TRAVERSE List:List:Member:DiscoveredNetworkConnection PROCESS WITH networkConnectionInfo`	Pemilik aplikasi BMC Discovery
Ekstrak data tentang penemuan aplikasi.	Untuk mendapatkan informasi tentang dependensi aplikasi, gunakan Query Builder untuk menjalankan kueri berikut: `search SoftwareInstance show key as 'SRC App ID', #Dependant:Dependency:DependedUpon:SoftwareInstance.key as 'DEST App ID'`	Pemilik aplikasi BMC Discovery

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak data tentang layanan bisnis.	Untuk mengekstrak data pada layanan bisnis yang disediakan oleh host, gunakan Query Builder untuk menjalankan kueri berikut: `search Host show name, #Host:HostedSoftware:AggregateSoftware:BusinessService.name as 'Name'`	Pemilik aplikasi BMC Discovery

Pemecahan masalah

Isu	Solusi
Kueri gagal dijalankan atau berisi kolom yang tidak terisi.	Tinjau catatan aset di BMC Discovery dan tentukan bidang mana yang Anda butuhkan. Kemudian, ganti bidang ini dalam kueri dengan menggunakan Query Builder.
Rincian aset dependen tidak diisi.	Ini kemungkinan karena izin akses atau konektivitas jaringan. Alat penemuan mungkin tidak memiliki izin yang diperlukan untuk mengakses aset tertentu, terutama jika mereka berada di jaringan yang berbeda atau di lingkungan yang berbeda. Kami menyarankan Anda bekerja sama dengan ahli materi pelajaran penemuan untuk memastikan bahwa semua aset yang relevan diidentifikasi.

Sumber daya terkait

Referensi

- Hak Lisensi BMC Discovery (dokumentasi BMC)
- Fitur dan komponen BMC Discovery (dokumentasi BMC)
- Panduan Pengguna BMC Discovery (dokumentasi BMC)
- Mencari data (pada BMC Discovery) (dokumentasi BMC)
- Penemuan dan analisis portofolio untuk migrasi (AWS Prescriptive Guidance)

Tutorial dan video

- Penemuan BMC: Webinar - Praktik Terbaik Kueri Pelaporan (Bagian 1) () YouTube

Pindah

Topik

- Migrasikan database Amazon RDS for Oracle ke database Akun AWS lain Wilayah AWS dan gunakan untuk replikasi yang sedang berlangsung AWS DMS
- Migrasikan instans Amazon RDS DB ke VPC atau akun lain
- Migrasi cluster Amazon Redshift ke Wilayah AWS di Tiongkok
- Transportasi database PostgreSQL antara dua instans Amazon RDS DB menggunakan pg_transport

Migrasikan database Amazon RDS for Oracle ke database Akun AWS lain Wilayah AWS dan gunakan untuk replikasi yang sedang berlangsung AWS DMS

Durga Prasad Cheepuri dan Eduardo Valentim, Amazon Web Services

Ringkasan

Peringatan: Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Pola ini memandu Anda melalui langkah-langkah untuk memigrasikan Amazon Relational Database Service (Amazon RDS) untuk database sumber Oracle ke database sumber yang berbeda dan. Akun AWS Wilayah AWS Pola ini menggunakan snapshot DB untuk pemuatan data penuh satu kali, dan memungkinkan AWS Database Migration Service (AWS DMS) untuk replikasi yang sedang berlangsung.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS yang berisi sumber Amazon RDS for Oracle database, yang telah dienkripsi menggunakan kunci () non-default AWS Key Management Service AWS KMS
- Aktif Akun AWS dalam database yang berbeda Wilayah AWS dari sumber, untuk digunakan untuk target Amazon RDS for Oracle database
- Virtual private cloud (VPC) mengintip antara sumber dan target VPCs
- Keakraban dengan menggunakan database Oracle sebagai sumber AWS DMS
- Keakraban dengan menggunakan database Oracle sebagai target AWS DMS

Versi produk

- Oracle versi 11g (versi 11.2.0.3.v1 dan yang lebih baru) dan hingga 12.2, dan 18c. Untuk daftar terbaru versi dan edisi yang didukung, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS dan dengan Menggunakan database Oracle sebagai target AWS DMS dalam dokumentasi. AWS Untuk versi Oracle yang didukung oleh Amazon RDS, lihat Oracle di Amazon RDS.

Arsitektur

Tumpukan teknologi sumber dan target

- Amazon RDS for Oracle DB instans

Arsitektur replikasi yang sedang berlangsung

Alat

Alat yang digunakan untuk pemuatan data penuh satu kali

- Amazon Relational Database Service (Amazon RDS) membuat snapshot volume penyimpanan instans DB Anda, mencadangkan seluruh instans DB dan bukan hanya database individual. Saat membuat snapshot DB, Anda perlu mengidentifikasi instans DB mana yang akan Anda cadangkan, kemudian beri nama snapshot DB sehingga Anda dapat memulihkannya nanti. Jumlah waktu yang diperlukan untuk membuat snapshot bervariasi sesuai ukuran basis data Anda. Karena snapshot menyertakan seluruh volume penyimpanan, ukuran file, seperti file sementara, juga memengaruhi jumlah waktu yang diperlukan untuk membuat snapshot. Untuk informasi selengkapnya tentang penggunaan snapshot DB, lihat Membuat Snapshot DB di dokumentasi Amazon RDS.
- AWS Key Management Service (AWS KMS) membuat kunci untuk enkripsi Amazon RDS. Saat Anda membuat instans DB terenkripsi, Anda juga dapat menyediakan pengenal kunci untuk AWS KMSkunci enkripsi Anda. Jika Anda tidak menentukan pengenal AWS KMSkunci, Amazon RDS menggunakan kunci enkripsi default untuk instans DB baru Anda. AWS KMSmembuat kunci enkripsi default Anda untuk Anda Akun AWS. Anda Akun AWS memiliki kunci enkripsi default yang berbeda untuk masing-masing Wilayah AWS. Untuk pola ini, instans Amazon RDS DB harus dienkripsi menggunakan kunci non-default. AWS KMS Untuk informasi selengkapnya tentang penggunaan AWS KMSkunci untuk enkripsi Amazon RDS, lihat Mengenkripsi sumber daya Amazon RDS di dokumentasi Amazon RDS.

Alat yang digunakan untuk replikasi yang sedang berlangsung

- AWS Database Migration Service (AWS DMS) digunakan untuk mereplikasi perubahan yang sedang berlangsung dan untuk menjaga database sumber dan target tetap sinkron. Untuk

informasi selengkapnya tentang penggunaan AWS DMS untuk replikasi yang sedang berlangsung, lihat Bekerja dengan instance AWS DMS replikasi dalam dokumentasi. AWS DMS

Epik

Konfigurasikan sumber Anda Akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan sumber Oracle DB instance.	Biarkan instans Amazon RDS for Oracle DB berjalan dalam mode ARCHIVELOG, dan atur periode retensi. Untuk detailnya, lihat Bekerja dengan database Oracle AWS terkelola sebagai sumber untuk AWS DMS.	DBA
Tetapkan logging tambahan untuk instans DB Oracle sumber.	Tetapkan pencatatan tambahan tingkat database dan tingkat tabel untuk instans Amazon RDS for Oracle DB. Untuk detailnya, lihat Bekerja dengan database Oracle AWS terkelola sebagai sumber untuk AWS DMS.	DBA
Perbarui kebijakan AWS KMS utama di akun sumber.	Perbarui kebijakan AWS KMS kunci di sumber Akun AWS untuk memungkinkan target Akun AWS menggunakan kunci Amazon AWS KMS RDS terenkripsi. Untuk detailnya, lihat AWS KMS dokumentasi.	SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat snapshot Amazon RDS DB manual dari instans DB sumber.		Pengguna AWS IAM
Bagikan snapshot Amazon RDS manual yang dienkripsi dengan target. Akun AWS	Untuk detailnya, lihat Berbagi snapshot DB.	Pengguna AWS IAM

Konfigurasikan target Anda Akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan kebijakan.	Di target Akun AWS, lampirkan kebijakan AWS Identity and Access Management (IAM) ke pengguna IAM root, untuk memungkinkan pengguna IAM menyalin snapshot DB terenkripsi menggunakan kunci bersama. AWS KMS	SysAdmin
Beralih ke sumbernya Wilayah AWS.		Pengguna AWS IAM
Salin snapshot bersama.	Di konsol Amazon RDS, di panel Snapshots, pilih Dibagikan dengan Saya, dan pilih snapshot bersama. Salin snapshot ke Wilayah AWS sama dengan database sumber dengan menggunak an Amazon Resource Name (ARN) untuk kunci AWS KMS yang digunakan oleh database	Pengguna AWS IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber. Untuk detailnya, lihat Menyalin snapshot DB.	
Beralih ke target Wilayah AWS, dan buat AWS KMS kunci baru.		Pengguna AWS IAM
Salin snapshot.	Beralih ke sumbernya Wilayah AWS. Di konsol Amazon RDS, di panel Snapshots, pilih Dimiliki oleh Saya, dan pilih snapshot yang disalin. Salin snapshot ke target Wilayah AWS dengan menggunakan AWS KMS kunci untuk target Wilayah AWS baru.	Pengguna AWS IAM
Pulihkan snapshot yang telah disalin.	Beralih ke target Wilayah AWS. Di konsol Amazon RDS, di panel Snapshots , pilih Dimiliki oleh Saya. Pilih snapshot yang disalin dan kembalikan ke instans Amazon RDS for Oracle DB. Untuk detailnya, lihat Memulihkan dari snapshot DB.	Pengguna AWS IAM

Siapkan database sumber Anda untuk replikasi yang sedang berlangsung

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna Oracle dengan izin yang sesuai.	Buat pengguna Oracle dengan hak istimewa yang diperlukan untuk Oracle sebagai sumber untuk. AWS DMS Untuk	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	detailnya, lihat AWS DMS dokumentasi.	
Konfigurasikan database sumber untuk Oracle LogMiner atau Oracle Binary Reader.		DBA

Siapkan database target Anda untuk replikasi yang sedang berlangsung

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna Oracle dengan izin yang sesuai.	Buat pengguna Oracle dengan hak istimewa yang diperlukan untuk Oracle sebagai target. AWS DMS Untuk detailnya, lihat AWS DMS dokumentasi.	DBA

Buat AWS DMS komponen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat contoh replikasi di target Wilayah AWS.	Buat instance replikasi di VPC target. Wilayah AWS Untuk detailnya, lihat AWS DMS dokumentasi.	Pengguna AWS IAM
Buat titik akhir sumber dan target dengan enkripsi yang diperlukan, dan uji koneksi.	Untuk detailnya, lihat AWS DMS dokumentasi.	DBA
Buat tugas replikasi.	1. Untuk jenis migrasi, pilih replikasi berkelanjutan.	Pengguna IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Untuk titik awal change data capture (CDC), gunakan Oracle system change number (SCN) saat snapshot Amazon RDS diambil untuk pemuatan penuh, atau stempel waktu saat beban penuh diambil. 3. Untuk `TargetTablePrepMode`, pilih DO_NOTHING. Jika tugas memiliki tabel data objek biner besar (LOB), pilih mode LOB terbatas, dan atur ukuran LOB maks ke ukuran maksimum data LOB dalam tabel. 4. Aktifkan logging. 5. Kelompokkan tabel yang terkait melalui kunci ke dalam satu tugas. Jika ada tabel dengan sejumlah besar data LOB dan tabel tidak memiliki hubungan dengan tabel lain, buat tugas terpisah untuk itu dengan pengaturan LOB yang dijelaskan sebelumnya. Untuk detailnya, lihat AWS DMS dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai tugas dan pantau mereka.	Untuk detailnya, lihat AWS DMS dokumentasi.	Pengguna AWS IAM
Aktifkan validasi pada tugas jika diperlukan.	Perhatikan bahwa mengaktif kan validasi memang memiliki dampak kinerja pada replikasi . Untuk detailnya, lihat AWS DMS dokumentasi.	Pengguna AWS IAM

Sumber daya terkait

- Mengubah kebijakan utama
- Membuat snapshot Amazon RDS DB manual
- Berbagi snapshot Amazon RDS DB manual
- Menyalin snapshot
- Memulihkan dari snapshot Amazon RDS DB
- Memulai dengan AWS DMS
- Menggunakan database Oracle sebagai sumber AWS DMS
- Menggunakan database Oracle sebagai target untuk AWS DMS
- AWS DMS pengaturan menggunakan pengintip VPC
- Bagaimana cara berbagi snapshot Amazon RDS DB manual atau snapshot cluster DB dengan yang lain? Akun AWS (Artikel AWS Knowledge Center)

Migrasikan instans Amazon RDS DB ke VPC atau akun lain

Dhrubajyoti Mukherjeee, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan instans DB Amazon Relational Database Service (Amazon RDS) dari satu cloud pribadi virtual (VPC) ke yang lain di akun AWS yang sama, atau dari satu akun AWS ke akun AWS lainnya.

Pola ini berguna jika Anda ingin memigrasikan instans Amazon RDS DB Anda ke VPC atau akun lain untuk alasan pemisahan atau keamanan (misalnya, ketika Anda ingin menempatkan tumpukan aplikasi dan database Anda di tempat yang berbeda). VPCs

Migrasi instans DB ke akun AWS lain melibatkan langkah-langkah seperti mengambil snapshot manual, membagikannya, dan memulihkan snapshot di akun target. Proses ini bisa memakan waktu, tergantung pada perubahan database dan tingkat transaksi. Ini juga menyebabkan downtime database, jadi rencanakan ke depan untuk migrasi. Pertimbangkan strategi blue/green penerapan untuk meminimalkan waktu henti. Atau, Anda dapat mengevaluasi AWS Data Migration Service (AWS DMS) untuk meminimalkan waktu henti untuk perubahan tersebut. Namun, pola ini tidak mencakup opsi ini. Untuk mempelajari lebih lanjut, lihat dokumentasi AWS DMS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Izin AWS Identity and Access Management (IAM) diperlukan untuk konsol VPC, subnet, dan Amazon RDS

Batasan

- Perubahan pada VPC menyebabkan reboot database, mengakibatkan pemadaman aplikasi. Kami menyarankan Anda bermigrasi selama waktu puncak rendah.
- Batasan saat memigrasikan Amazon RDS ke VPC lain:
  - Instans DB yang Anda migrasi harus berupa satu instance tanpa siaga. Itu tidak harus menjadi anggota cluster.
  - Amazon RDS tidak boleh berada di beberapa Availability Zone.

  - Amazon RDS tidak boleh memiliki replika baca.
  - Grup subnet yang dibuat di VPC target harus memiliki subnet dari Availability Zone tempat database sumber berjalan.
- Batasan saat memigrasikan Amazon RDS ke akun AWS lain:
  - Berbagi snapshot yang dienkripsi dengan kunci layanan default untuk Amazon RDS saat ini tidak didukung.

Arsitektur

Migrasi ke VPC di akun AWS yang sama

Diagram berikut menunjukkan alur kerja untuk memigrasikan instans Amazon RDS DB ke VPC lain di akun AWS yang sama.

Langkah-langkahnya terdiri dari yang berikut ini. Lihat bagian Epik untuk instruksi terperinci.

1. Buat grup subnet DB di VPC target. Grup subnet DB adalah kumpulan subnet yang dapat Anda gunakan untuk menentukan VPC tertentu saat Anda membuat instance DB.
2. Konfigurasikan instans Amazon RDS DB di VPC sumber untuk menggunakan grup subnet DB baru.
3. Terapkan perubahan untuk memigrasikan Amazon RDS DB ke VPC target.

Migrasi ke akun AWS yang berbeda

Diagram berikut menunjukkan alur kerja untuk memigrasikan instans Amazon RDS DB ke akun AWS yang berbeda.

Langkah-langkahnya terdiri dari yang berikut ini. Lihat bagian Epik untuk instruksi terperinci.

1. Akses instans Amazon RDS DB di akun AWS sumber.
2. Buat snapshot Amazon RDS di akun AWS sumber.
3. Bagikan snapshot Amazon RDS dengan akun AWS target.
4. Akses snapshot Amazon RDS di akun AWS target.
5. Buat instans Amazon RDS DB di akun AWS target.

Alat

Layanan AWS

- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Praktik terbaik

- Jika downtime database menjadi perhatian saat memigrasikan instans Amazon RDS DB ke akun lain, sebaiknya gunakan AWS DMS. Layanan ini menyediakan replikasi data, yang menyebabkan kurang dari lima menit waktu pemadaman.

Epik

Migrasi ke VPC lain di akun AWS yang sama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC baru.	Di konsol VPC Amazon, buat VPC dan subnet baru dengan properti dan rentang alamat IP yang diinginkan. Untuk petunjuk terperinci, lihat dokumentasi Amazon VPC.	Administrator
Buat grup subnet DB.	Di konsol Amazon RDS: 1. Pilih grup Subnet, Buat grup subnet DB. 2. Masukkan nama grup subnet, deskripsi, dan ID VPC.	Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Tambahkan subnet yang termasuk dalam grup subnet. Tambahkan subnet untuk mencakup setidaknya dua Availability Zone. 4. Pilih Buat. Untuk informasi tambahan, lihat dokumentasi Amazon RDS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah instans Amazon RDS DB untuk menggunakan grup subnet baru.	Di konsol Amazon RDS: 1. Di panel navigasi, pilih Databases, lalu pilih instans Amazon RDS DB yang akan dimigrasikan. 2. Di bagian Konektivitas, pilih grup subnet yang terkait dengan VPC target. 3. Di bagian Jadwal modifikasi, pilih Terapkan segera. Saat migrasi ke VPC target selesai, grup keamanan default VPC target ditetapkan ke instans Amazon RDS DB. Anda dapat mengonfigurasi grup keamanan baru untuk VPC tersebut dengan aturan masuk dan keluar yang diperlukan ke instans DB Anda. Atau, gunakan AWS Command Line Interface (AWS CLI) untuk melakukan migrasi ke VPC target dengan secara eksplisit memberikan ID grup keamanan VPC baru. Contoh: `aws rds modify-db-instance \ --db-instance-identifier testrds \`	Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--db-subnet-group-name new-vpc-subnet-group \ --vpc-security-group-ids sg-idxxxx \ --apply-immediately`	

Migrasi ke akun AWS yang berbeda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup VPC dan subnet baru di akun AWS target.	1. Di konsol VPC Amazon, buat VPC baru dengan properti dan rentang alamat IP yang diinginkan. Untuk petunjuk terperinci, lihat dokumentasi Amazon VPC. 2. Buat subnet untuk VPC baru dengan mengikuti petunjuk dalam dokumentasi Amazon VPC. 3. Di konsol Amazon RDS, buat grup subnet DB. Untuk petunjuk, lihat dokumentasi Amazon RDS.	Administrator
Bagikan snapshot manual database dan bagikan dengan akun target.	1. Ambil snapshot manual dari database sumber dengan mengikuti petunjuk dalam dokumentasi Amazon RDS. 2. Bagikan snapshot dengan akun AWS target dengan memberikan ID akun target. Untuk petunjuk, lihat artikel	Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	re:Post tentang berbagi snapshot DB dengan akun lain.	
Luncurkan instans Amazon RDS DB baru.	Luncurkan instans Amazon RDS DB baru dari snapshot bersama di akun AWS target. Untuk petunjuk, lihat dokumentasi Amazon RDS.	Administrator

Sumber daya terkait

- Dokumentasi Amazon VPC
- Dokumentasi Amazon RDS
- Bagaimana cara mengubah VPC untuk instance RDS DB? (AWS re:Artikel posting)
- Bagaimana cara mentransfer kepemilikan sumber daya Amazon RDS ke akun AWS yang berbeda? (AWS re:Artikel posting)
- Bagaimana cara membagikan snapshot Amazon RDS DB manual atau snapshot cluster Aurora DB dengan akun AWS lain? (AWS re:Artikel posting)
- Dokumentasi AWS DMS

Migrasi cluster Amazon Redshift ke Wilayah AWS di Tiongkok

Jing Yan, Amazon Web Services

Ringkasan

Pola ini menyediakan step-by-step pendekatan untuk memigrasikan cluster Amazon Redshift ke Wilayah AWS di Tiongkok dari Wilayah AWS lainnya.

Pola ini menggunakan perintah SQL untuk membuat ulang semua objek database, dan menggunakan perintah UNLOAD untuk memindahkan data ini dari Amazon Redshift ke bucket Amazon Simple Storage Service (Amazon S3) di Wilayah sumber. Data tersebut kemudian dimigrasikan ke bucket S3 di Wilayah AWS di Tiongkok. Perintah COPY digunakan untuk memuat data dari bucket S3 dan mentransfernya ke cluster Amazon Redshift target.

Amazon Redshift saat ini tidak mendukung fitur Lintas wilayah seperti penyalinan snapshot ke Wilayah AWS di Tiongkok. Pola ini menyediakan cara untuk mengatasi batasan itu. Anda juga dapat membalikkan langkah-langkah dalam pola ini untuk memigrasikan data dari Wilayah AWS di Tiongkok ke Wilayah AWS lainnya.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif di Wilayah Tiongkok dan Wilayah AWS di luar Tiongkok
- Cluster Amazon Redshift yang ada di Wilayah Tiongkok dan Wilayah AWS di luar Tiongkok

Batasan

- Ini adalah migrasi offline, yang berarti cluster Amazon Redshift sumber tidak dapat melakukan operasi tulis selama migrasi.

Arsitektur

Tumpukan teknologi sumber

- Cluster Amazon Redshift di Wilayah AWS di luar Tiongkok

Tumpukan teknologi target

- Cluster Amazon Redshift di Wilayah AWS di Tiongkok

Arsitektur target

Alat

Alat

- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja. Anda dapat menggunakan Amazon S3 untuk menyimpan data dari Amazon Redshift, dan Anda dapat menyalin data dari bucket S3 ke Amazon Redshift.
- Amazon Redshift — Amazon Redshift adalah layanan gudang data skala petabyte yang dikelola sepenuhnya di cloud.
- psql — psql adalah front-end berbasis terminal untuk PostgreSQL.

Epik

Bersiaplah untuk migrasi di Wilayah sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan dan konfigura sikan EC2 instance di Wilayah sumber.	Masuk ke AWS Managemen t Console dan buka konsol Amazon Elastic Compute Cloud (Amazon EC2). Wilayah Anda saat ini ditampilkan di bilah navigasi di bagian atas layar. Wilayah ini tidak bisa menjadi Wilayah AWS di Tiongkok. Dari dasbor EC2 konsol Amazon, pilih “Luncurkan instance”, lalu buat dan konfigurasikan EC2 instance. Penting: Pastikan grup EC2 keamanan Anda	DBA, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk aturan masuk mengizinkan akses tidak terbatas ke port TCP 22 dari mesin sumber Anda. Untuk petunjuk tentang cara meluncurkan dan mengonfigurasi EC2 instance, lihat bagian “Sumber daya terkait”.	
Instal alat psql.	Unduh dan instal PostgreSQL. Amazon Redshift tidak menyediakan alat psql, itu diinstal dengan PostgreSQL. Untuk informasi selengkapnya tentang menggunakan psql dan menginstal alat PostgreSQL, lihat bagian “Sumber daya terkait”.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rekam detail cluster Amazon Redshift.	Buka konsol Amazon Redshift, dan pilih “Clusters” di panel navigasi. Kemudian pilih nama cluster Amazon Redshift dari daftar. Pada tab “Properti”, di bagian “Konfigurasi basis data”, catat “Nama basis data” dan “Port.” Buka bagian “Detail koneksi” dan rekam “Titik Akhir”, yang ada dalam format “titik akhir:<port>/<databasename>”. Penting: Pastikan grup keamanan Amazon Redshift untuk aturan masuk mengizinkan akses tidak terbatas ke port TCP 5439 dari instans Anda. EC2	DBA
Hubungkan psql ke cluster Amazon Redshift.	<userid><databasename><port>Pada prompt perintah, tentukan informasi koneksi dengan menjalankan perintah “psql -h <endpoint>-U -d -p”. Pada prompt kata sandi psql, masukkan kata sandi untuk pengguna <userid>"”. Anda kemudian terhubung ke cluster Amazon Redshift dan dapat memasukkan perintah secara interaktif.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buka konsol Amazon S3, dan buat bucket S3 untuk menyimpan file yang diekspor dari Amazon Redshift. Untuk petunjuk tentang cara membuat bucket S3, lihat bagian “Sumber daya terkait”.	DBA, AWS Umum
Buat kebijakan IAM yang mendukung pembongkaran data.	Buka konsol AWS Identity and Access Management (IAM) dan pilih “Kebijakan.” Pilih “Buat kebijakan,” dan pilih tab “JSON”. Salin dan tempel kebijakan IAM untuk membongkar data dari bagian “Informasi tambahan”. Penting: Ganti “s3_bucket_name” dengan nama bucket S3 Anda. Pilih “Kebijakan tinjau”, lalu masukkan nama dan deskripsi untuk kebijakan tersebut. Pilih “Buat kebijakan.”	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk memungkinkan operasi UNLOAD untuk Amazon Redshift.	Buka konsol IAM dan pilih “Peran.” Pilih “Buat peran,” dan pilih “Layanan AWS” di “Pilih jenis entitas tepercaya.” Pilih “Redshift” untuk layanan, pilih “Redshift - Customizable,” dan kemudian pilih “Next.” Pilih kebijakan “Bongkar” yang Anda buat sebelumnya, dan pilih “Berikutnya.” Masukkan “Nama peran,” dan pilih “Buat peran.”	DBA
Kaitkan peran IAM dengan cluster Amazon Redshift.	Buka konsol Amazon Redshift, dan pilih “Kelola peran IAM.” Pilih “Peran yang tersedia” dari menu tarik-turun dan pilih peran yang Anda buat sebelumnya. Pilih “Terapkan perubahan.” Ketika “Status” untuk peran IAM pada “Kelola peran IAM” ditampilkan sebagai “In-sync”, Anda dapat menjalankan perintah UNLOAD.	DBA
Hentikan operasi penulisan ke cluster Amazon Redshift.	Anda harus ingat untuk menghentikan semua operasi penulisan ke cluster Amazon Redshift sumber hingga migrasi selesai.	DBA

Bersiaplah untuk migrasi di Wilayah target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan dan konfigura sikan EC2 instance di Wilayah target.	Masuk ke AWS Managemen t Console untuk Wilayah di Tiongkok, baik Beijing maupun Ningxia. Dari EC2 konsol Amazon, pilih “Luncurkan instance,” dan buat serta konfigurasikan EC2 instance. Penting: Pastikan grup EC2 keamanan Amazon Anda untuk aturan masuk mengizink an akses tidak terbatas ke port TCP 22 dari mesin sumber Anda. Untuk petunjuk lebih lanjut tentang cara meluncurk an dan mengonfigurasi EC2 instance, lihat bagian “Sumber daya terkait”.	DBA
Rekam detail cluster Amazon Redshift.	Buka konsol Amazon Redshift, dan pilih “Clusters” di panel navigasi. Kemudian pilih nama cluster Amazon Redshift dari daftar. Pada tab “Properti”, di bagian “Konfigurasi basis data”, catat “Nama basis data” dan “Port.” Buka bagian “Detail koneksi” dan rekam “Titik Akhir”, yang ada dalam format “titik akhir:<port>/ <databasename>”. Penting: Pastikan grup keamanan Amazon Redshift untuk aturan masuk mengizinkan akses	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tidak terbatas ke port TCP 5439 dari instans Anda. EC2	
Hubungkan psql ke cluster Amazon Redshift.	<userid><databasename><port>Pada prompt perintah, tentukan informasi koneksi dengan menjalankan perintah “psql -h <endpoint>-U -d -p”. Pada prompt kata sandi psql, masukkan kata sandi untuk pengguna <userid>"”. Anda kemudian terhubung ke cluster Amazon Redshift dan dapat memasukkan perintah secara interaktif.	DBA
Buat ember S3.	Buka konsol Amazon S3, dan buat bucket S3 untuk menyimpan file yang diekspor dari Amazon Redshift. Untuk bantuan dengan ini dan cerita lainnya, lihat bagian “Sumber daya terkait”.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM yang mendukung penyalinan data.	Buka konsol IAM dan pilih “Kebijakan.” Pilih “Buat kebijakan,” dan pilih tab “JSON”. Salin dan tempel kebijakan IAM untuk menyalin data dari bagian “Informas i tambahan”. Penting: Ganti “s3_bucket_name” dengan nama bucket S3 Anda. Pilih “Kebijakan peninjauan”, masukkan nama dan deskripsi untuk kebijakan tersebut. Pilih “Buat kebijakan.”	DBA
Buat peran IAM untuk memungkinkan operasi COPY untuk Amazon Redshift.	Buka konsol IAM dan pilih “Peran.” Pilih “Buat peran,” dan pilih “Layanan AWS” di “Pilih jenis entitas tepercaya.” Pilih “Redshift” untuk layanan, pilih “Redshift - Customizable,” dan kemudian pilih “Next.” Pilih kebijakan “Salin” yang Anda buat sebelumnya, dan pilih “Berikutnya.” Masukkan “Nama peran,” dan pilih “Buat peran.”	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kaitkan peran IAM dengan cluster Amazon Redshift.	Buka konsol Amazon Redshift, dan pilih “Kelola peran IAM.” Pilih “Peran yang tersedia” dari menu tarik-turun dan pilih peran yang Anda buat sebelumnya. Pilih “Terapkan perubahan.” Ketika “Status” untuk peran IAM pada “Kelola peran IAM” ditampilkan sebagai “In-sync”, Anda dapat menjalankan perintah “COPY”.	DBA

Verifikasi data sumber dan informasi objek sebelum memulai migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi baris di tabel Amazon Redshift sumber.	Gunakan skrip di bagian “Informasi tambahan” untuk memverifikasi dan mencatat jumlah baris di tabel Amazon Redshift sumber. Ingatlah untuk membagi data secara merata untuk skrip UNLOAD dan COPY. Ini akan meningkatkan efisiensi bongkar muat dan pemuatan data, karena kuantitas data yang dicakup oleh setiap skrip akan seimbang.	DBA
Verifikasi jumlah objek database di cluster Amazon Redshift sumber.	Gunakan skrip di bagian “Informasi tambahan” untuk memverifikasi dan mencatat jumlah database, pengguna,	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	skema, tabel, tampilan, dan fungsi yang ditentukan pengguna (UDFs) di kluster Amazon Redshift sumber Anda.	
Verifikasi hasil pernyataan SQL sebelum migrasi.	Beberapa pernyataan SQL untuk validasi data harus diurutkan menurut situasi bisnis dan data yang sebenarnya. Ini untuk memverifikasi data yang diimpor untuk memastikannya konsisten dan ditampilkan dengan benar.	DBA

Migrasikan data dan objek ke Wilayah target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan skrip DDL Amazon Redshift.	Hasilkan skrip Data Definitio n Language (DDL) dengan menggunakan tautan dari bagian “SQL statement to query Amazon Redshift” di bagian “Informasi tambahan”. Skrip DDL ini harus mencakup “buat pengguna,” “buat skema,” “hak istimewa skema untuk pengguna,” “buat tabel/ tampilan,” “hak istimewa pada objek ke pengguna,” dan “buat fungsi” kueri.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat objek di cluster Amazon Redshift untuk Wilayah target.	Jalankan skrip DDL dengan menggunakan AWS Command Line Interface (AWS CLI) Interface (AWS CLI) di Wilayah AWS di Tiongkok. Skrip ini akan membuat objek di cluster Amazon Redshift untuk Wilayah target.	DBA
Bongkar sumber data cluster Amazon Redshift ke bucket S3.	Jalankan perintah UNLOAD untuk membongkar data dari cluster Amazon Redshift di Wilayah sumber ke bucket S3.	DBA, Pengembang
Transfer sumber data bucket Region S3 ke bucket Region S3 target.	Transfer data dari bucket Region S3 sumber Anda ke bucket S3 target. Karena perintah “$ aws s3 sync” tidak dapat digunakan, pastikan Anda menggunakan proses yang diuraikan dalam artikel “Mentransfer data Amazon S3 dari Wilayah AWS ke Wilayah AWS di Tiongkok” di bagian “Sumber daya terkait”.	Developer
Muat data ke cluster Amazon Redshift target.	Di alat psql untuk Wilayah target Anda, jalankan perintah COPY untuk memuat data dari bucket S3 ke cluster Amazon Redshift target.	DBA

Verifikasi data di wilayah sumber dan target setelah migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi dan bandingkan jumlah baris dalam tabel sumber dan target.	Verifikasi dan bandingkan jumlah baris tabel di wilayah sumber dan target untuk memastikan semua dimigrasikan.	DBA
Verifikasi dan bandingkan jumlah objek basis data sumber dan target.	Verifikasi dan bandingkan semua objek database di daerah sumber dan target untuk memastikan semua dimigrasikan.	DBA
Verifikasi dan bandingkan hasil skrip SQL di daerah sumber dan target.	Jalankan skrip SQL yang disiapkan sebelum migrasi. Verifikasi dan bandingkan data untuk memastikan bahwa hasil SQL benar.	DBA
Setel ulang kata sandi semua pengguna di cluster Amazon Redshift target.	Setelah migrasi selesai dan semua data diverifikasi, Anda harus mengatur ulang semua kata sandi pengguna untuk klaster Amazon Redshift di Wilayah AWS di Tiongkok.	DBA

Sumber daya terkait

- Mentransfer data Amazon S3 dari Wilayah AWS ke Wilayah AWS di Tiongkok
- Membuat bucket S3
- Menyetel ulang kata sandi pengguna Amazon Redshift
- dokumentasi psql

Informasi tambahan

Kebijakan IAM untuk membongkar data

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::s3_bucket_name"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject", "s3:DeleteObject"],
      "Resource": ["arn:aws:s3:::s3_bucket_name/*"]
    }
  ]
}
```

Kebijakan IAM untuk menyalin data

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::s3_bucket_name"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::s3_bucket_name/*"]
    }
  ]
}
```

Pernyataan SQL untuk menanyakan Amazon Redshift

```
##Database

select * from pg_database where datdba>1;
```

```
##User

select * from pg_user where usesysid>1;

##Schema

SELECT n.nspname AS "Name",

  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"

FROM pg_catalog.pg_namespace n

WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'

ORDER BY 1;

##Table

select count(*) from pg_tables where schemaname not in
 ('pg_catalog','information_schema');

select schemaname,count(*) from pg_tables where schemaname not in
 ('pg_catalog','information_schema') group by schemaname order by 1;

##View

SELECT

    n.nspname AS schemaname,c.relname AS
 viewname,pg_catalog.pg_get_userbyid(c.relowner) as "Owner"

FROM

    pg_catalog.pg_class AS c

INNER JOIN

    pg_catalog.pg_namespace AS n

    ON c.relnamespace = n.oid

WHERE relkind = 'v' and n.nspname not in ('information_schema','pg_catalog');
```

```
##UDF

SELECT

   n.nspname AS schemaname,

   p.proname AS proname,

   pg_catalog.pg_get_userbyid(p.proowner) as "Owner"

FROM pg_proc p

LEFT JOIN pg_namespace n on n.oid = p.pronamespace

WHERE p.proowner != 1;
```

Skrip SQL untuk menghasilkan pernyataan DDL

- Get_schema_priv_by_user skrip
- Skrip Hasilkan_TBL_DDL
- Hasilkan_View_DDL
- Generate_User_Grant_Revoke_DDL
- Hasilkan_UDF_DDL

Transportasi database PostgreSQL antara dua instans Amazon RDS DB menggunakan pg_transport

Raunak Rishabh dan Jitender Kumar, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah untuk memigrasikan database yang sangat besar antara dua Amazon Relational Database Service (Amazon RDS) untuk instance PostgreSQL DB dengan menggunakan ekstensi pg_transport. Ekstensi ini memberikan mekanisme transportasi fisik untuk memindahkan setiap basis data. Dengan streaming file database dengan pemrosesan minimal, ini menyediakan metode yang sangat cepat untuk memigrasikan database besar antara instans DB dengan waktu henti minimal. Ekstensi ini menggunakan model tarik di mana instans DB target mengimpor database dari instance DB sumber.

Prasyarat dan batasan

Prasyarat

- Kedua instans DB harus menjalankan versi utama PostgreSQL yang sama.
- Database tidak boleh ada pada target. Jika tidak, transportasi gagal.
- Tidak ada ekstensi selain pg_transport yang harus diaktifkan di database sumber.
- Semua objek database sumber harus dalam default pg_default tablespace.
- Grup keamanan instans DB sumber harus mengizinkan lalu lintas dari instans DB target.
- Instal klien PostgreSQL seperti psql PgAdminatau untuk bekerja dengan instans Amazon RDS PostgreSQL DB. Anda dapat menginstal klien baik di sistem lokal Anda atau menggunakan instans Amazon Elastic Compute Cloud (Amazon EC2). Dalam pola ini, kita menggunakan psql pada sebuah EC2 instance.

Batasan

- Anda tidak dapat mengangkut database antara berbagai versi utama Amazon RDS for PostgreSQL.
- Hak akses dan kepemilikan dari database sumber tidak ditransfer ke database target.
- Anda tidak dapat mengangkut database pada replika baca atau pada instance induk dari replika baca.

- Anda tidak dapat menggunakan tipe data reg dalam tabel database apa pun yang Anda rencanakan untuk diangkut dengan metode ini.
- Anda dapat menjalankan hingga 32 total transportasi (termasuk impor dan ekspor) secara bersamaan pada instans DB.
- Anda tidak dapat mengganti nama atau include/exclude tabel. Semuanya bermigrasi apa adanya.

Perhatian

- Buat cadangan sebelum menghapus ekstensi, karena menghapus ekstensi juga menghapus objek dependen dan beberapa data yang penting untuk pengoperasian database.
- Pertimbangkan kelas instance dan proses yang berjalan pada database lain pada instance sumber saat Anda menentukan jumlah pekerja dan `work_mem` nilai untuk pg_transport.
- Ketika transport dimulai, semua koneksi pada database sumber berakhir dan database dimasukkan ke dalam mode read-only.

> **Note**
>
> Ketika transport berjalan pada satu database, itu tidak mempengaruhi database lain di server yang sama.

Versi produk

- Amazon RDS untuk PostgreSQL 10.10 dan yang lebih baru, dan Amazon RDS for PostgreSQL 11.5 dan yang lebih baru. Untuk informasi versi terbaru, lihat Mengangkut Database PostgreSQL Antara Instans DB dalam dokumentasi Amazon RDS.

Arsitektur

Alat

- pg_transport menyediakan mekanisme transportasi fisik untuk memindahkan setiap database. Dengan streaming file database dengan pemrosesan minimal, transportasi fisik memindahkan data jauh lebih cepat daripada proses dump dan load tradisional dan membutuhkan waktu henti minimal. Basis data PostgreSQL yang dapat diangkut menggunakan model tarik yaitu instans DB

tujuan mengimpor basis data dari instans DB sumber. Anda menginstal ekstensi ini pada instans DB Anda ketika Anda menyiapkan sumber dan lingkungan target, seperti yang dijelaskan dalam pola ini.

- psql memungkinkan Anda untuk terhubung ke, dan bekerja dengan, instans PostgreSQL DB Anda. Untuk menginstal psql pada sistem Anda, lihat halaman PostgreSQL Downloads.

Epik

Buat grup parameter target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup parameter untuk sistem target.	Tentukan nama grup yang mengidentifikasinya sebagai kelompok parameter target; misalnya,`pgtarget-param-group`. Untuk petunjuk, lihat dokumentasi Amazon RDS.	DBA
Ubah parameter untuk grup parameter.	Atur parameter berikut: 1. Tambahkan `pg_transport` ke `shared_preload_libraries` parameter. `shared_preload_libraries = pg_stat_statements, pg_transport` 2. Atur parameter `pg_transport.num_workers`. Pilih jumlah pekerja yang ingin Anda gunakan untuk menjalankan transportasi. Nilai yang Anda tetapkan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menentukan jumlah `transport.send_file` pekerja yang akan dibuat di sumber. 3. Tingkatkan nilai `max_worker_processes` menjadi lebih dari tiga kali nilai`pg_transport.num_workers`. Misalnya, jika Anda menetapkan nilai `pg_transport.num_workers` ke 4, `max_worker_processes` nilainya harus setidaknya 13. Jika ini gagal, pg_transport merekomendasikan nilai minimum. 4. Setel `pg_transport.timing` ke 1. Pengaturan ini memungkinkan pelaporan informasi waktu selama transportasi. 5. Atur parameter `pg_transport.work_mem`. Parameter ini menentukan memori maksimum untuk mengalokasikan untuk setiap pekerja. Nilai default adalah 128 MB.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang parameter ini, lihat dokumentasi Amazon RDS.	

Buat grup parameter sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup parameter untuk sistem sumber.	Tentukan nama grup yang mengidentifikasinya sebagai grup parameter sumber; misalnya,`pgsource-param-group` . Untuk petunjuk, lihat dokumentasi Amazon RDS.	DBA
Ubah parameter untuk grup parameter.	Atur parameter berikut: 1. Tambahkan `pg_transport` ke `shared_preload_libraries` parameter. `shared_preload_libraries = pg_stat_statements, pg_transport` 2. Atur parameter `pg_transport.num_workers` . Nilai parameter ini yang ditentukan dalam target menentukan jumlah `transport.send_file` pekerja yang akan digunakan. Jika Anda	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memiliki impor yang berjalan pada instance ini, tingkatkan nilai ini, tetapi pertimbangkan jumlah pekerja yang sudah berjalan. 3. Tingkatkan nilai `max_worker_processes` menjadi lebih dari tiga kali nilai `pg_transport.num_workers` pada target. Misalnya, jika Anda menetapkan nilai `pg_transport.num_workers` ke 4 pada target, `max_worker_processes` nilai pada sumber harus setidaknya 13. Jika ini gagal, pg_transport merekomendasikan nilai minimum. 4. Atur parameter `pg_transport.work_mem` . Parameter ini menentukan memori maksimum untuk mengalokasikan untuk setiap pekerja. Nilai default adalah 128 MB. Untuk informasi selengkapnya tentang parameter ini, lihat dokumentasi Amazon RDS.	

Siapkan lingkungan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon RDS for PostgreSQL DB baru untuk mengangkut database sumber Anda.	Tentukan kelas instance dan versi PostgreSQL berdasarkan kebutuhan bisnis Anda.	DBA, Administrator sistem, Arsitek basis data
Ubah grup keamanan target untuk memungkinkan koneksi pada port instans DB dari EC2 instance.	Secara default, port untuk instance PostgreSQL adalah 5432. Jika Anda menggunakan port lain, koneksi ke port itu harus terbuka untuk EC2 instance tersebut.	DBA, Administrator sistem
Ubah instance, dan tetapkan grup parameter target baru.	Misalnya, `pgtarget-param-group` .	DBA
Mulai ulang instans DB Amazon RDS target.	Parameter `shared_preload_libraries` dan `max_worker_processes` parameter statis dan memerlukan reboot dari instance.	DBA, Administrator sistem
Connect ke database dari EC2 instance menggunakan psql.	Gunakan perintah: `psql -h <rds_end_point> -p PORT -U username -d database -W`	DBA
Buat ekstensi pg_transport.	Jalankan kueri berikut sebagai pengguna dengan `rds_superuser` peran: `create extension pg_transport;`	DBA

Siapkan lingkungan sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah grup keamanan sumber untuk mengizinkan koneksi pada port instans DB dari instans Amazon dan menargetkan EC2 instans DB	Secara default, port untuk instance PostgreSQL adalah 5432. Jika Anda menggunakan port lain, koneksi ke port itu harus terbuka untuk EC2 instance tersebut.	DBA, Administrator sistem
Ubah instance dan tetapkan grup parameter sumber baru.	Misalnya, `pgsource-param-group` .	DBA
Mulai ulang instans DB Amazon RDS sumber.	Parameter `shared_preload_libraries` dan `max_worker_processes` parameter statis dan memerlukan reboot dari instance.	DBA
Connect ke database dari EC2 instance menggunakan psql.	Gunakan perintah: `psql -h <rds_end_point> -p PORT -U username -d database -W`	DBA
Buat ekstensi pg_transport dan hapus semua ekstensi lain dari database yang akan diangkut.	Transport akan gagal jika ada ekstensi selain pg_transport yang diinstal pada database sumber. Perintah ini harus dijalankan oleh pengguna dengan `rds_superuser` peran.	DBA

Lakukan transportasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan lari kering.	Gunakan `transport.import_from_server` fungsi untuk melakukan dry run terlebih dahulu: `SELECT transport.import_from_server( 'source-db-instance-endpoint', source-db-instance-port, 'source-db-instance-user', 'source-user-password', 'source-database-name', 'destination-user-password', 'true');` Parameter terakhir dari fungsi ini (diatur ke`true`) mendefinisikan dry run. Fungsi ini menampilkan kesalahan apa pun yang akan Anda lihat ketika Anda menjalankan transportasi utama. Selesaikan kesalahan sebelum Anda menjalankan transportasi utama.	DBA
Jika dry run berhasil, memulai transportasi database.	Jalankan `transport.import_from_server` fungsi untuk melakukan transportasi. Ini terhubung ke sumber dan mengimpor data.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`SELECT transport .import_from_serve r( 'source-db-instanc e-endpoint', source- db-instance-port, 'source-db-instance- user', 'source-user- password', 'source- database-name', 'destination-user- password', false);` Parameter terakhir dari fungsi ini (set to`false`) menunjukkan bahwa ini bukan dry run.	
Lakukan langkah pasca-tra nsportasi.	Setelah transportasi database selesai: • Validasi data di lingkungan target. • Tambahkan semua peran dan izin ke target. • Aktifkan semua ekstensi yang diperlukan di target dan sumber, jika diperlukan. • Kembalikan nilai `max_worker_process es` parameter.	DBA

Sumber daya terkait

- Dokumentasi Amazon RDS
- dokumentasi pg_transport
- Migrasi database menggunakan RDS PostgreSQL Transportable Databases (posting blog)

- Unduhan PostgreSQL
- utilitas psql
- Membuat Grup Parameter DB
- Ubah Parameter dalam Grup Parameter DB
- Unduhan PostgreSQL

Platform Ulang

Topik

- Ekspor database Microsoft SQL Server ke Amazon S3 dengan menggunakan AWS DMS
- Migrasikan nilai Oracle CLOB ke baris individual di PostgreSQL di AWS
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle dengan menggunakan Impor Oracle Data Pump langsung melalui tautan database
- Migrasi Oracle ke PeopleSoft Amazon RDS Kustom
- Migrasikan fungsionalitas Oracle ROWID ke PostgreSQL di AWS
- Migrasikan kode kesalahan Oracle Database ke database yang kompatibel dengan Amazon Aurora PostgreSQL
- Migrasikan SAP ASE di Amazon EC2 ke Amazon Aurora PostgreSQL yang kompatibel dengan menggunakan AWS SCT dan AWS DMS
- Migrasikan sertifikat SSL Windows ke Application Load Balancer menggunakan ACM
- Memigrasikan antrian pesan dari Microsoft Azure Service Bus ke Amazon SQS
- Migrasikan database Oracle JD Edwards EnterpriseOne ke AWS dengan menggunakan Oracle Data Pump dan AWS DMS
- Migrasikan PeopleSoft database Oracle ke AWS dengan menggunakan AWS DMS
- Memigrasikan database MySQL lokal ke Amazon RDS for MySQL
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server
- Migrasi data dari Microsoft Azure Blob ke Amazon S3 dengan menggunakan Rclone
- Migrasi dari Couchbase Server ke Couchbase Capella di AWS
- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2
- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2 dengan Auto Scaling
- Migrasi aplikasi.NET dari Microsoft Azure App Service ke AWS Elastic Beanstalk
- Bermigrasi dari Oracle WebLogic ke Apache Tomcat (ToMee) di Amazon ECS
- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS for Oracle menggunakan AWS DMS
- Memigrasi database Oracle lokal ke Amazon OpenSearch Service menggunakan Logstash
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle menggunakan Oracle Data Pump

- Bermigrasi dari PostgreSQL di Amazon ke Amazon RDS untuk PostgreSQL menggunakan pglogical EC2
- Memigrasikan database PostgreSQL lokal ke Aurora PostgreSQL
- Memigrasi database Microsoft SQL Server lokal ke Microsoft SQL Server di Amazon yang menjalankan Linux EC2
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan server tertaut
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan metode pencadangan dan pemulihan asli
- Migrasikan database Microsoft SQL Server ke Aurora MySQL dengan menggunakan AWS DMS dan AWS SCT
- Memigrasi database MariaDB lokal ke Amazon RDS for MariaDB menggunakan alat bawaan
- Memigrasikan database MySQL lokal ke Aurora MySQL
- Migrasikan database MySQL lokal ke Aurora MySQL menggunakan Percona, Amazon EFS, dan Amazon S3 XtraBackup
- Migrasikan aplikasi Java lokal ke AWS menggunakan AWS App2Container
- Migrasi sistem file bersama dalam migrasi AWS besar
- Migrasikan database Oracle ke Amazon RDS for Oracle dengan menggunakan adaptor file datar Oracle GoldenGate
- Ubah aplikasi Python dan Perl untuk mendukung migrasi database dari Microsoft SQL Server ke Amazon Aurora PostgreSQL Edisi yang kompatibel

Ekspor database Microsoft SQL Server ke Amazon S3 dengan menggunakan AWS DMS

Sweta Krishna, Amazon Web Services

Ringkasan

Organizations sering perlu menyalin database ke Amazon Simple Storage Service (Amazon S3) untuk migrasi database, backup dan restore, pengarsipan data, dan analisis data. Pola ini menjelaskan bagaimana Anda dapat mengekspor database Microsoft SQL Server ke Amazon S3. Basis data sumber dapat di-host di tempat atau di Amazon Elastic Compute Cloud (Amazon EC2) atau Amazon Relational Database Service (Amazon RDS) untuk Microsoft SQL Server di Amazon Web Services (AWS) Cloud.

Data diekspor menggunakan AWS Database Migration Service (AWS DMS). Secara default, AWS DMS menulis data full load dan change data capture (CDC) dalam format comma-separated value (.csv). Untuk penyimpanan yang lebih ringkas dan opsi kueri yang lebih cepat, pola ini menggunakan opsi format Apache Parquet (.parquet).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Peran AWS Identity and Access Management (IAM) untuk akun dengan akses tulis, hapus, dan tag ke bucket S3 target, dan AWS DMS (`dms.amazonaws.com`) ditambahkan sebagai entitas tepercaya ke peran IAM ini
- Database Microsoft SQL Server lokal (atau Microsoft SQL Server pada EC2 instans atau database Amazon RDS for SQL Server)
- Konektivitas jaringan antara virtual private cloud (VPC) di AWS dan jaringan lokal yang disediakan oleh AWS Direct Connect atau jaringan pribadi virtual (VPN)

Batasan

- Bucket S3 berkemampuan VPC (gateway VPC) saat ini tidak didukung dalam versi AWS DMS lebih awal dari 3.4.7.
- Perubahan pada struktur tabel sumber selama beban penuh tidak didukung.

- Mode AWS DMS full large binary object (LOB) tidak didukung.

Versi produk

- Microsoft SQL Server versi 2005 atau yang lebih baru untuk edisi Enterprise, Standard, Workgroup, dan Developer.
- Support untuk Microsoft SQL Server versi 2019 sebagai sumber tersedia di AWS DMS versi 3.3.2 dan yang lebih baru.

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server lokal (atau Microsoft SQL Server pada EC2 instans atau database Amazon RDS for SQL Server)

Tumpukan teknologi target

- AWS Direct Connect
- AWS DMS
- Amazon S3

Arsitektur target

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- AWS Direct Connect menautkan jaringan internal Anda ke lokasi Direct Connect melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke layanan AWS publik sambil melewati penyedia layanan internet di jalur jaringan Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi database.	Validasi versi basis data sumber dan pastikan itu didukung oleh AWS DMS. Untuk informasi tentang versi database SQL Server yang didukung, lihat Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS.	DBA
Buat VPC dan grup keamanan.	Di akun AWS Anda, buat VPC dan grup keamanan. Untuk informasi selengkapnya, lihat dokumentasi Amazon VPC.	Administrator Sistem
Buat pengguna untuk tugas AWS DMS.	Buat pengguna AWS DMS di database sumber dan berikan izin BACA. Pengguna ini akan digunakan oleh AWS DMS.	DBA
Uji konektivitas DB.	Uji konektivitas ke instans SQL Server DB dari pengguna AWS DMS.	DBA
Buat ember S3.	Buat bucket S3 target. Bucket ini akan menyimpan data tabel yang dimigrasi.	Administrator sistem
Buat kebijakan dan peran IAM.	1. Untuk membuat kebijakan IAM dengan izin bucket, gunakan kode di bagian Informasi tambahan.	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buat peran untuk AWS DMS, dan lampirkan kebijakan ke peran tersebut.	

Migrasi data dengan menggunakan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans replikasi AWS DMS.	Masuk ke AWS Management Console, dan buka konsol AWS DMS. Di panel navigasi, pilih Instance replikasi, Buat instance replikasi. Untuk petunjuk, lihat langkah 1 dalam dokumentasi AWS DMS.	DBA
Buat titik akhir sumber dan target.	Buat titik akhir sumber dan target. Uji koneksi dari instance replikasi ke titik akhir sumber dan target. Untuk petunjuk, lihat langkah 2 dalam dokumentasi AWS DMS.	DBA
Buat tugas replikasi.	Buat tugas replikasi, dan pilih full load atau full load dengan change data capture (CDC) untuk memigrasikan data dari SQL Server ke bucket S3. Untuk petunjuk, lihat langkah 3 dalam dokumentasi AWS DMS.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai replikasi data.	Mulai tugas replikasi, dan pantau log untuk kesalahan apa pun.	DBA

Validasi data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi data yang dimigrasi.	Di konsol, navigasikan ke bucket S3 target Anda. Buka subfolder yang memiliki nama yang sama dengan database sumber. Konfirmasikan bahwa folder berisi semua tabel yang dimigrasi dari database sumber.	DBA

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan dan hapus sumber daya AWS sementara.	Matikan sumber daya AWS sementara yang Anda buat untuk migrasi data, seperti instans replikasi AWS DMS, dan hapus setelah Anda memvalidasi ekspor.	DBA

Sumber daya terkait

- Panduan Pengguna AWS Database Migration Service
- Menggunakan database Microsoft SQL Server sebagai sumber AWS DMS
- Menggunakan Amazon S3 sebagai target AWS Database Migration Service

- Menggunakan bucket S3 sebagai target AWS DMS (AWS re:post)

Informasi tambahan

Gunakan kode berikut untuk menambahkan kebijakan IAM dengan izin bucket S3 untuk peran AWS DMS. Ganti `bucketname` dengan nama bucket Anda.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "s3:PutObject",
                 "s3:DeleteObject"
             ],
             "Resource": [
                 "arn:aws:s3:::bucketname*"
             ]
         },
         {
             "Effect": "Allow",
             "Action": [
                 "s3:ListBucket"
             ],
             "Resource": [
                 "arn:aws:s3:::bucketname*"
             ]
         }
     ]
 }
```

Migrasikan nilai Oracle CLOB ke baris individual di PostgreSQL di AWS

Sai Krishna Namburu dan Sindhusha Paturu, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara membagi nilai objek besar karakter Oracle (CLOB) menjadi baris individual di Amazon Aurora PostgreSQL Compatible Edition dan Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL. PostgreSQL tidak mendukung tipe data CLOB.

Tabel dengan partisi interval diidentifikasi dalam database Oracle sumber, dan nama tabel, jenis partisi, interval partisi, dan metadata lainnya ditangkap dan dimuat ke dalam database target. Anda dapat memuat data CLOB yang berukuran kurang dari 1 GB ke dalam tabel target sebagai teks menggunakan AWS Database Migration Service (AWS DMS), atau Anda dapat mengekspor data dalam format CSV, memuatnya ke bucket Amazon Simple Storage Service (Amazon S3), dan memigrasikannya ke database PostgreSQL target Anda.

Setelah migrasi, Anda dapat menggunakan kode PostgreSQL kustom yang disediakan dengan pola ini untuk membagi data CLOB menjadi baris individual berdasarkan pengidentifikasi `CHR(10)` karakter baris baru () dan mengisi tabel target.

Prasyarat dan batasan

Prasyarat

- Sebuah tabel database Oracle yang memiliki partisi interval dan catatan dengan tipe data CLOB.
- Aurora PostgreSQL kompatibel atau Amazon RDS untuk database PostgreSQL yang memiliki struktur tabel yang mirip dengan tabel sumber (kolom dan tipe data yang sama).

Batasan

- Nilai CLOB tidak boleh melebihi 1 GB.
- Setiap baris dalam tabel target harus memiliki pengidentifikasi karakter baris baru.

Versi produk

- Oracle 12c
- Aurora Postgres 11.6

Arsitektur

Diagram berikut menunjukkan tabel Oracle sumber dengan data CLOB, dan tabel PostgreSQL setara di Aurora PostgreSQL versi kompatibel 11.6.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di AWS Cloud.
- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat lainnya

Anda dapat menggunakan alat klien berikut untuk menghubungkan, mengakses, dan mengelola Aurora PostgreSQL yang kompatibel dan Amazon RDS for PostgreSQL database. (Alat ini tidak digunakan dalam pola ini.)

- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.
- DBeaveradalah alat database open-source untuk pengembang dan administrator database. Anda dapat menggunakan alat ini untuk memanipulasi, memantau, menganalisis, mengelola, dan memigrasi data Anda.

Praktik terbaik

Untuk praktik terbaik untuk memigrasikan database Anda dari Oracle ke PostgreSQL, lihat PostgreSQL PostgreSQL AWS Praktik terbaik untuk memigrasikan database Oracle ke Amazon RDS PostgreSQL atau Amazon Aurora PostgreSQL: Proses migrasi dan pertimbangan infrastruktur.

Untuk praktik terbaik untuk mengonfigurasi tugas AWS DMS untuk memigrasikan objek biner besar, lihat Memigrasi objek biner besar () LOBs dalam dokumentasi AWS DMS.

Epik

Identifikasi data CLOB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Analisis data CLOB.	Dalam database Oracle sumber, analisis data CLOB untuk melihat apakah itu berisi header kolom, sehingga Anda dapat menentukan metode untuk memuat data ke dalam tabel target. Untuk menganalisis data input, gunakan kueri berikut. `SELECT * FROM clobdata_or;`	Developer
Muat data CLOB ke database target.	Migrasikan tabel yang memiliki data CLOB ke tabel interim (staging) di database target Aurora atau Amazon RDS. Anda dapat menggunakan AWS DMS atau mengungga h data sebagai file CSV ke bucket Amazon S3. Untuk informasi tentang penggunaan AWS DMS untuk tugas ini, lihat Menggunak an database Oracle sebagai sumber dan Menggunakan database PostgreSQL sebagai	Insinyur migrasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	target dalam dokumentasi AWS DMS. Untuk informasi tentang penggunaan Amazon S3 untuk tugas ini, lihat Menggunakan Amazon S3 sebagai target dalam dokumentasi AWS DMS.	

Validasi tabel PostgreSQL target.	Validasi data target, termasuk header, terhadap data sumber dengan menggunakan kueri berikut dalam database target.
	
	`SELECT * FROM clobdata_pg;`
	`SELECT * FROM clobdatatarget;`
	
	Bandingkan hasil dengan hasil kueri dari database sumber (dari langkah pertama).
		Developer

Pisahkan data CLOB menjadi baris terpisah.	Jalankan kode PostgreSQL kustom yang disediakan di bagian Informasi tambahan untuk membagi data CLOB dan memasukkannya ke dalam baris terpisah dalam tabel PostgreSQL target.	Developer

Validasi data.

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi data dalam tabel target.	Validasi data yang dimasukkan ke dalam tabel target dengan menggunakan kueri berikut.
	`SELECT * FROM clobdata_pg;`
	`SELECT * FROM clobdatatarget;`
		Developer

Sumber daya terkait

- Jenis data CLOB (dokumentasi Oracle)
- Tipe data (dokumentasi PostgreSQL)

Informasi tambahan

Fungsi PostgreSQL untuk memisahkan data CLOB

```
do
$$
declare
totalstr varchar;
str1 varchar;
str2 varchar;
pos1 integer := 1;
pos2 integer ;
len integer;


begin
        select rawdata||chr(10) into totalstr from clobdata_pg;
        len :=  length(totalstr) ;
        raise notice 'Total length : %',len;
```

```
        raise notice 'totalstr : %',totalstr;
        raise notice 'Before while loop';


        while pos1 < len  loop


                  select position (chr(10) in totalstr) into pos2;
                 raise notice '1st position of new line : %',pos2;


                str1 := substring (totalstr,pos1,pos2-1);
                raise notice 'str1 : %',str1;


                   insert into clobdatatarget(data) values (str1);
                   totalstr := substring(totalstr,pos2+1,len);
                   raise notice 'new totalstr :%',totalstr;
                len :=  length(totalstr) ;


        end loop;
end
$$
LANGUAGE 'plpgsql' ;
```

Contoh input dan output

Anda dapat menggunakan contoh berikut untuk mencoba kode PostgreSQL sebelum memigrasikan data.

Buat database Oracle dengan tiga baris input.

```
CREATE TABLE clobdata_or (
id INTEGER GENERATED ALWAYS AS IDENTITY,
rawdata clob  );


insert into clobdata_or(rawdata) values (to_clob('test line 1') || chr(10) ||
 to_clob('test line 2') || chr(10) || to_clob('test line 3') || chr(10));
COMMIT;
```

```
SELECT * FROM clobdata_or;
```

Ini menampilkan output berikut.

id	data mentah
1	garis uji 1 garis uji 2 garis uji 3

Muat data sumber ke dalam tabel pementasan PostgreSQL () untuk diproses. `clobdata_pg`

```
SELECT * FROM clobdata_pg;

CREATE TEMP TABLE clobdatatarget (id1 SERIAL,data VARCHAR );

<Run the code in the additional information section.>

SELECT * FROM clobdatatarget;
```

Ini menampilkan output berikut.

id1	data
1	garis uji 1
2	garis uji 2
3	garis uji 3

Memigrasikan database Oracle lokal ke Amazon RDS for Oracle dengan menggunakan Impor Oracle Data Pump langsung melalui tautan database

Rizwan Wangde, Amazon Web Services

Ringkasan

Banyak pola mencakup migrasi database Oracle lokal ke Amazon Relational Database Service (Amazon RDS) untuk Oracle dengan menggunakan Oracle Data Pump, utilitas Oracle asli yang merupakan cara yang disukai untuk memigrasikan beban kerja Oracle yang besar. Pola-pola ini biasanya melibatkan mengekspor skema aplikasi atau tabel ke dalam file dump, mentransfer file dump ke direktori database di Amazon RDS for Oracle, dan kemudian mengimpor skema aplikasi dan data dari file dump.

Dengan menggunakan pendekatan itu, migrasi dapat memakan waktu lebih lama tergantung pada ukuran data dan waktu yang diperlukan untuk mentransfer file dump ke instans Amazon RDS. Selain itu, file dump berada pada volume Amazon Elastic Block Store (Amazon EBS) instans Amazon RDS, yang harus cukup besar untuk database dan file dump. Ketika file dump dihapus setelah impor, ruang kosong tidak dapat diambil, sehingga Anda terus membayar untuk ruang yang tidak digunakan.

Pola ini mengurangi masalah tersebut dengan melakukan impor langsung pada instans Amazon RDS dengan menggunakan Oracle Data Pump API (`DBMS_DATAPUMP`) melalui tautan database. Pola ini memulai jalur ekspor dan impor simultan antara basis data sumber dan target. Pola ini tidak memerlukan ukuran volume EBS untuk file dump karena tidak ada file dump yang dibuat atau disimpan pada volume. Pendekatan ini menghemat biaya bulanan ruang disk yang tidak digunakan.

Prasyarat dan batasan

Prasyarat

- Akun Amazon Web Services (AWS) yang aktif.
- Virtual Private Cloud (VPC) dikonfigurasi dengan subnet pribadi di setidaknya dua Availability Zone, untuk menyediakan infrastruktur jaringan untuk instans Amazon RDS.
- Database Oracle di pusat data lokal, atau dikelola sendiri di Amazon Elastic Compute Cloud (Amazon). EC2
- Instans Amazon RDS for Oracle yang ada di satu Availability Zone. Menggunakan Availability Zone tunggal meningkatkan kinerja penulisan selama migrasi. Penerapan multi-AZ dapat diaktifkan 24-48 jam sebelum cutover.

Solusi ini juga dapat menggunakan Amazon RDS Custom untuk Oracle sebagai target.

- AWS Direct Connect (direkomendasikan untuk database berukuran besar).
- Konektivitas jaringan dan aturan firewall di tempat yang dikonfigurasi untuk mengizinkan koneksi masuk dari instans Amazon RDS ke database Oracle lokal.

Batasan

- Batas ukuran database di Amazon RDS for Oracle adalah 64 tebibytes (TiB) per Desember 2022.
- Ukuran maksimum satu file pada instans Amazon RDS for Oracle DB adalah 16 TiB. Ini penting untuk diketahui karena Anda mungkin perlu menyebarkan tabel di beberapa ruang tabel.

Versi produk

- Database sumber: Oracle Database versi 10g Rilis 1 dan yang lebih baru.
- Basis data target: Untuk daftar terbaru versi dan edisi yang didukung di Amazon RDS, lihat Amazon RDS for Oracle dalam dokumentasi AWS.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle yang dikelola sendiri di tempat atau di cloud

Tumpukan teknologi target

- Amazon RDS for Oracle atau Amazon RDS Kustom untuk Oracle

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk bermigrasi dari database Oracle lokal ke Amazon RDS for Oracle dalam lingkungan Single-AZ. Arah panah menggambarkan aliran data dalam arsitektur. Diagram tidak menunjukkan komponen apa yang memulai koneksi.

1. Instans Amazon RDS for Oracle terhubung ke database Oracle sumber lokal untuk melakukan migrasi muatan penuh melalui tautan database.

2. AWS Database Migration Service (AWS DMS) menghubungkan ke database Oracle sumber lokal untuk melakukan replikasi berkelanjutan dengan menggunakan change data capture (CDC).
3. Perubahan CDC diterapkan ke database Amazon RDS for Oracle.

Alat

Layanan AWS

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke dalam AWS Cloud atau di antara kombinasi pengaturan cloud dan lokal. Pola ini menggunakan CDC dan data Replikasi hanya mengubah pengaturan.
- AWS Direct Connectmenghubungkan jaringan internal Anda ke Direct Connect lokasi melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke publik Layanan AWS sambil melewati penyedia layanan internet di jalur jaringan Anda.
- Amazon Relational Database Service membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional Oracle di AWS Cloud.

Alat lainnya

- Oracle Data Pump membantu Anda memindahkan data dan metadata dari satu database ke database lainnya dengan kecepatan tinggi.
- Alat klien seperti Oracle Instant Client atau SQL Developer digunakan untuk menghubungkan dan menjalankan query SQL pada database.

Praktik terbaik

Meskipun Direct Connect menggunakan koneksi jaringan pribadi khusus antara jaringan lokal dan AWS, pertimbangkan opsi berikut untuk keamanan tambahan dan enkripsi data untuk data dalam perjalanan:

- Jaringan pribadi virtual (VPN) yang menggunakan AWS Site-to-Site VPN atau koneksi IPsec VPN dari jaringan lokal ke AWS jaringan
- Enkripsi Jaringan Asli Oracle Database yang dikonfigurasi pada database Oracle lokal
- Enkripsi menggunakan TLS

Epik

Siapkan database Oracle sumber lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan konektivitas jaringan dari database target ke database sumber.	Konfigurasikan jaringan lokal dan firewall untuk mengizinkan koneksi masuk dari instans Amazon RDS target ke database Oracle sumber lokal.	Administrator jaringan, Insinyur keamanan

Buat pengguna database dengan hak istimewa yang sesuai.	Buat pengguna database di database Oracle sumber lokal dengan hak istimewa untuk memigrasikan data antara sumber dan target menggunakan Oracle Data Pump:
	`GRANT CONNECT to <migration_user>;`
	`GRANT DATAPUMP_EXP_FULL_DATABASE to <migration_user>;`
	`GRANT SELECT ANY TABLE to <migration_user>;`
		DBA

Siapkan database sumber lokal untuk migrasi AWS DMS CDC.	(Opsional) Siapkan database Oracle sumber lokal untuk migrasi AWS DMS CDC setelah menyelesaikan Oracle Data Pump Full Load: 1. Konfigurasikan hak istimewa tambahan yang diperlukan untuk mengelola FLASHBACK selama migrasi Oracle Data Pump:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` GRANT FLASHBACK ANY TABLE to <migratio n_user>; GRANT FLASHBACK ARCHIVE ADMINISTER to <migration_user>; ``` 2. Untuk mengonfigurasi hak istimewa akun pengguna yang diperlukan pada sumber Oracle yang dikelola sendiri AWS DMS, lihat dokumentasinya.AWS DMS 3. Untuk menyiapkan database sumber Oracle yang dikelola sendiri untuk CDC menggunakan AWS DMS, lihat dokumenta si.AWS DMS	
Instal dan konfigurasikan SQL Developer.	Instal dan konfigurasikan SQL Developer untuk menghubun gkan dan menjalankan kueri SQL pada basis data sumber dan target.	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan skrip untuk membuat ruang tabel.	Gunakan contoh query SQL berikut untuk menghasilkan script pada database sumber: `SELECT 'CREATE TABLESPACE ' tablespace_name ' DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE UNLIMITED;' from dba_tablespaces where tablespace_name not in ('SYSTEM', 'SYSAUX','TEMP','UNDOTBS1') order by 1;` Script akan diterapkan pada database target.	DBA
Hasilkan skrip untuk membuat pengguna, profil, peran, dan hak istimewa.	Untuk menghasilkan skrip untuk membuat pengguna database, profil, peran, dan hak istimewa, gunakan skrip dari dokumen Dukungan Oracle Cara Mengekstrak DDL untuk Pengguna termasuk Hak Istimewa dan Peran Menggunakan dbms_metadata.get_ddl (Doc ID 2739952.1) (Diperlukan akun Oracle). Script akan diterapkan pada database target.	DBA

Siapkan target Amazon RDS for Oracle instance

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat tautan database ke database sumber dan verifikasi konektivitas.	Untuk membuat link database ke database sumber lokal, Anda dapat menggunakan perintah contoh berikut:
	`CREATE DATABASE LINK link2src CONNECT TO <migration_user_account> IDENTIFIED BY <password> USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>) (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';`
	Untuk memverifikasi konektivitas, jalankan perintah SQL berikut:
	`select * from dual@link2src;`
	Konektivitas berhasil jika responsnyaX.
		DBA

Jalankan skrip untuk menyiapkan instance target.	Jalankan skrip yang dibuat sebelumnya untuk menyiapkan target Amazon RDS for Oracle instance:	DBA, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Ruang Meja 2. Profil 3. Peran Ini membantu memastikan bahwa migrasi Oracle Data Pump dapat membuat skema dan objeknya.	

Lakukan migrasi beban penuh dengan menggunakan Oracle Data Pump Import melalui tautan database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan skema yang diperlukan.	Untuk memigrasikan skema yang diperlukan dari database lokal sumber ke instans Amazon RDS target, gunakan kode di bagian Informasi tambahan: • Untuk memigrasikan skema tunggal, jalankan Kode 1 dari bagian Informasi tambahan. • Untuk memigrasikan beberapa skema, jalankan Kode 2 dari bagian Informasi tambahan. Untuk menyetel kinerja migrasi, Anda dapat menyesuaikan jumlah proses	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	paralel dengan menjalankan perintah berikut: `DBMS_DATAPUMP.SET_ PARALLEL (handle => v_hdnl, degree => 4);`	
Kumpulkan statistik skema untuk meningkatkan kinerja.	Perintah Gather Schema Statistics mengembalikan statistik pengoptimal kueri Oracle yang dikumpulkan untuk objek database. Dengan menggunakan informasi ini, pengoptimal dapat memilih rencana eksekusi terbaik untuk kueri apa pun terhadap objek ini: `EXECUTE DBMS_STAT S.GATHER_SCHEMA_ST ATS(ownname => '<schema_name>');`	DBA

Lakukan migrasi beban penuh dan replikasi CDC dengan menggunakan Oracle Data Pump dan AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tangkap SCN pada sumber database Oracle lokal.	Tangkap nomor perubahan sistem (SCN) pada sumber database Oracle lokal. Anda akan menggunakan SCN untuk impor beban penuh dan sebagai titik awal untuk replikasi CDC.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk menghasilkan SCN saat ini pada database sumber, jalankan pernyataan SQL berikut: `SELECT current_scn FROM V$DATABASE;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan migrasi penuh skema.	Untuk memigrasikan skema (`FULL LOAD`) yang diperlukan dari database lokal sumber ke instans Amazon RDS target, lakukan hal berikut: • Untuk memigrasikan skema tunggal, jalankan Kode 3 dari bagian Informasi tambahan. • Untuk memigrasikan beberapa skema, jalankan Kode 4 dari bagian Informasi tambahan. Dalam kode, ganti `<CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>` dengan SCN yang Anda ambil dari database sumber: `DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdnl, name => 'FLASHBACK_SCN', value => <CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>);` Untuk menyetel kinerja migrasi, Anda dapat menyesuaikan jumlah proses paralel:	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);`	
Nonaktifkan pemicu di bawah skema migrasi.	Sebelum Anda memulai tugas AWS DMS khusus CDC, nonaktifkan skema di `TRIGGERS` bawah migrasi.	DBA
Kumpulkan statistik skema untuk meningkatkan kinerja.	Perintah Gather Schema Statistics mengembalikan statistik pengoptimal kueri Oracle yang dikumpulkan untuk objek database: `EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS(ownname => '<schema_name>');` Dengan menggunakan informasi ini, pengoptimal dapat memilih rencana eksekusi terbaik untuk kueri apa pun terhadap objek ini.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan AWS DMS untuk melakukan replikasi berkelanj utan dari sumber ke target.	Gunakan AWS DMS untuk melakukan replikasi berkelanj utan dari database Oracle sumber ke target Amazon RDS for Oracle instance. Untuk informasi selengkap nya, lihat Membuat tugas untuk replikasi berkelanjutan menggunakan AWS DMS dan posting blog Cara bekerja dengan dukungan CDC asli di. AWS DMS	DBA, insinyur Migrasi

Potong ke Amazon RDS for Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan Multi-AZ pada instans 48 jam sebelum cutover.	Jika ini adalah instans produksi, sebaiknya aktifkan penerapan Multi-AZ di instans Amazon RDS untuk memberikan manfaat ketersediaan tinggi (HA) dan pemulihan bencana (DR).	DBA, insinyur Migrasi
Hentikan tugas AWS DMS khusus CDC (jika CDC dihidupkan).	1. Pastikan latensi sumber dan latensi target pada CloudWatch metrik Amazon AWS DMS tugas menunjukkan 0 detik. 2. Hentikan tugas AWS DMS khusus CDC.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan pemicu.	Aktifkan `TRIGGERS` yang Anda nonaktifkan sebelum tugas CDC dibuat.	DBA

Sumber daya terkait

AWS

- Mempersiapkan database sumber yang dikelola sendiri Oracle untuk CDC menggunakan AWS DMS
- Membuat tugas untuk replikasi berkelanjutan menggunakan AWS DMS
- Penerapan multi-AZ untuk ketersediaan tinggi
- Cara bekerja dengan dukungan CDC asli di AWS DMS (posting blog)

Dokumentasi Oracle

- DBMS_DATAPUMP

Informasi tambahan

Kode 1: Migrasi beban penuh saja, skema aplikasi tunggal

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''<schema_name>'')');  --
 To migrate one selected schema
    DBMS_DATAPUMP.METADATA_FILTER (hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')'); --
 To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
```

```
/
```

Kode 2: Migrasi beban penuh saja, beberapa skema aplikasi

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'SCHEMA_LIST',
 '''<SCHEMA_1>'',''<SCHEMA_2>'', ''<SCHEMA_3>'''); -- To migrate multiple schemas
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')');
 -- To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Kode 3: Migrasi beban penuh sebelum tugas khusus CDC, skema aplikasi tunggal

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''<schema_name>'')');  --
 To migrate one selected schema
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')');
 -- To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdnl, name => 'FLASHBACK_SCN', value =>
 <CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>); -- SCN required for AWS DMS CDC only task.
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Kode 4: Migrasi beban penuh sebelum tugas khusus CDC, beberapa skema aplikasi

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN (operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE (handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'SCHEMA_LIST',
 '''<SCHEMA_1>'',''<SCHEMA_2>'', ''<SCHEMA_3>'''); -- To migrate multiple schemas
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')');
 -- To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdnl, name => 'FLASHBACK_SCN', value =>
 <CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>); -- SCN required for AWS DMS CDC only task.
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Skenario di mana pendekatan migrasi campuran dapat bekerja lebih baik

Dalam skenario langka di mana database sumber berisi tabel dengan jutaan baris dan kolom LOBSEGMENT berukuran sangat besar, pola ini akan memperlambat migrasi. Oracle bermigrasi LOBSEGMENTs melalui tautan jaringan satu per satu. Ini mengekstrak satu baris (bersama dengan data kolom LOB) dari tabel sumber, dan menyisipkan baris ke dalam tabel target, mengulangi proses sampai semua baris dimigrasikan. Oracle Data Pump melalui tautan database tidak mendukung beban massal atau mekanisme pemuatan jalur langsung untuk LOBSEGMENTs.

Dalam situasi ini, kami merekomendasikan yang berikut:

- Lewati tabel yang diidentifikasi selama migrasi Oracle Data Pump dengan menambahkan filter metadata berikut:

  ```
  dbms_datapump.metadata_filter(handle =>h1, name=>'NAME_EXPR', value => 'NOT IN
   (''TABLE_1'',''TABLE_2'')');
  ```

- Gunakan AWS DMS tugas (migrasi muatan penuh, dengan replikasi CDC jika diperlukan) untuk memigrasikan tabel yang diidentifikasi. AWS DMS akan mengekstrak beberapa baris dari database Oracle sumber, dan memasukkannya dalam batch ke instance Amazon RDS target, yang meningkatkan kinerja.

Migrasi Oracle ke PeopleSoft Amazon RDS Kustom

Gaurav Gupta, Amazon Web Services

Ringkasan

Oracle PeopleSoft adalah solusi perencanaan sumber daya perusahaan (ERP) untuk proses di seluruh perusahaan. PeopleSoft memiliki arsitektur tiga tingkat: klien, aplikasi, dan database. PeopleSoft dapat dijalankan di Amazon Relational Database Service (Amazon RDS). Sekarang, Anda juga dapat menjalankan PeopleSoft Amazon RDS Custom, yang menyediakan akses ke sistem operasi yang mendasarinya.

Amazon RDS Custom for Oracle adalah layanan database terkelola untuk aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi dan lingkungan database yang mendasarinya. Saat memigrasikan database Oracle ke Amazon RDS Custom, Amazon Web Services (AWS) dapat mengelola tugas pencadangan dan ketersediaan tinggi, sementara Anda dapat fokus pada pemeliharaan PeopleSoft aplikasi dan fungsionalitas Anda. Untuk faktor kunci yang perlu dipertimbangkan untuk migrasi, lihat Strategi migrasi database Oracle di AWS Prescriptive Guidance.

Pola ini berfokus pada langkah-langkah untuk memigrasikan PeopleSoft database di Amazon Elastic Compute Cloud (Amazon EC2) ke Amazon RDS Custom dengan menggunakan cadangan Oracle Recovery Manager (RMAN). Ini menggunakan sistem file bersama Amazon Elastic File System (Amazon EFS) antara EC2 instance dan Amazon RDS Custom, meskipun Anda juga dapat menggunakan Amazon FSx atau drive bersama apa pun. Pola ini menggunakan cadangan penuh RMAN (kadang-kadang disebut sebagai cadangan level 0).

Prasyarat dan batasan

Prasyarat

- Database sumber Oracle versi 19C yang berjalan di Amazon EC2 dengan Oracle Linux 7, Oracle Linux 8, Red Hat Enterprise Linux (RHEL) 7, atau RHEL 8. Dalam contoh untuk pola ini, nama database sumber adalah`FSDM092`, tetapi ini bukan persyaratan.

> **Note**
>
> Anda juga dapat menggunakan pola ini dengan database sumber Oracle lokal. Anda harus memiliki konektivitas jaringan yang sesuai antara jaringan lokal dan virtual private cloud (VPC).

- Sebuah contoh demo PeopleSoft 9.2.
- Sebuah tingkat PeopleSoft aplikasi tunggal. Namun, Anda dapat menyesuaikan pola ini untuk bekerja dengan beberapa tingkatan aplikasi.
- Amazon RDS Kustom dikonfigurasi dengan setidaknya 8 GB ruang swap.

Batasan

Pola ini tidak mendukung konfigurasi berikut:

- Mengatur `ARCHIVE_LAG_TARGET` parameter database ke nilai di luar rentang 60-7200
- Menonaktifkan mode log instans DB () `NOARCHIVELOG`
- Mematikan atribut instans yang dioptimalkan Amazon Elastic Block Store (Amazon EBS) EC2
- Memodifikasi volume EBS asli yang dilampirkan ke instance EC2
- Menambahkan volume EBS baru atau mengubah jenis volume dari gp2 ke gp3
- Mengubah format ekstensi untuk `LOG_ARCHIVE_FORMAT` parameter (membutuhkan`*.arc`)
- Multiplexing atau mengubah lokasi dan nama file kontrol (harus) `/rdsdbdata/db/*DBNAME*/controlfile/control-01.ctl`

Untuk informasi tambahan tentang konfigurasi ini dan konfigurasi lain yang tidak didukung, lihat dokumentasi Amazon RDS.

Versi produk

Untuk versi Oracle Database dan kelas instans yang didukung oleh Amazon RDS Custom, lihat Persyaratan dan batasan untuk Amazon RDS Custom for Oracle.

Arsitektur

Tumpukan teknologi target

- Penyeimbang Beban Aplikasi
- Amazon EFS
- Amazon RDS Custom for Oracle
- AWS Secrets Manager
- Amazon Simple Storage Service (Amazon S3)

Arsitektur target

Diagram arsitektur berikut mewakili PeopleSoft sistem yang berjalan di Availability Zone tunggal di AWS. Tingkat aplikasi diakses melalui Application Load Balancer. Baik aplikasi maupun database berada dalam subnet pribadi, dan Amazon RDS Custom dan Amazon EC2 database instance menggunakan sistem file bersama Amazon EFS untuk menyimpan dan mengakses file cadangan RMAN. Amazon S3 digunakan untuk membuat mesin RDS Oracle khusus dan untuk menyimpan metadata redo log.

Alat

Alat

Layanan AWS

- Amazon RDS Custom for Oracle adalah layanan database terkelola untuk aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi dan lingkungan database yang mendasarinya. Ini mengotomatiskan tugas-tugas administrasi database, seperti backup dan ketersediaan tinggi.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di AWS Cloud. Pola ini menggunakan sistem file bersama Amazon EFS untuk menyimpan dan mengakses file cadangan RMAN.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Dalam pola ini, Anda mengambil kata sandi pengguna database dari Secrets Manager untuk membuat `RDSADMIN` dan `ADMIN` pengguna dan untuk mengubah `sys` dan `system` kata sandi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone. Pola ini menggunakan Application Load Balancer.

Alat-alat lainnya

- Oracle Recovery Manager (RMAN) menyediakan dukungan cadangan dan pemulihan untuk database Oracle. Pola ini menggunakan RMAN untuk melakukan backup panas dari database Oracle sumber di Amazon EC2 yang dipulihkan di Amazon RDS Custom.

Praktik terbaik

- Untuk parameter inisialisasi database, sesuaikan pfile standar yang disediakan oleh instans Amazon RDS Custom DB PeopleSoft alih-alih menggunakan spfile dari database sumber Oracle. Ini karena spasi putih dan komentar menyebabkan masalah saat membuat replika baca di Amazon RDS Custom. Untuk informasi selengkapnya tentang parameter inisialisasi database, lihat Oracle Support Note 1100831.1 (memerlukan akun Oracle Support).
- Amazon RDS Custom menggunakan manajemen memori otomatis Oracle secara default. Jika Anda ingin menggunakan kernel Hugemem, Anda dapat mengonfigurasi Amazon RDS Custom untuk menggunakan manajemen memori bersama otomatis sebagai gantinya.
- Biarkan `memory_max_target` parameter diaktifkan secara default. Kerangka kerja menggunakan ini di latar belakang untuk membuat replika baca.
- Aktifkan Oracle Flashback Database. Fitur ini berguna saat mengembalikan siaga dalam skenario pengujian failover (bukan switchover).

Epik

Siapkan instans DB dan sistem file

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance DB.	Di konsol Amazon RDS, buat Amazon RDS Custom untuk instans Oracle DB dengan nama DB yang disebut FSDMO92 (atau nama database sumber Anda). Untuk petunjuknya, lihat Bekerja dengan Amazon RDS Custom dalam dokumenta si AWS dan Amazon RDS	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Custom for Oracle — New Control Capabilities in Database Environment postingan blog. Ini memastikan bahwa nama database diatur ke nama yang sama dengan database sumber. (Jika tetap kosong, nama EC2 instance dan database akan disetel ke`ORCL`.)	

Lakukan pencadangan penuh RMAN dari basis data Amazon EC2 sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skrip cadangan.	Buat skrip cadangan RMAN untuk mencadangkan database ke sistem file Amazon EFS yang Anda pasang (`/efs`dalam contoh berikut). Anda dapat menggunakan kode contoh atau menjalankan salah satu skrip RMAN yang ada. `#!/bin/bash` `Dt=`date +'%Y%m%d-%H%M'` `BACKUP_LOG="rman-${ORACLE_SID}-$Dt"` `export TAGDATE=`date +%Y%m%d%H%M`;` `LOGPATH=/u01/scripts/logs`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`rman target / >> $LOGPATH/rman-${ORACLE_SID}-$Dt << EOF`
	`SQL "ALTER SYSTEM SWITCH LOGFILE";`
	`SQL "ALTER SESSION SET NLS_DATE_FORMAT="DD.MM.YYYY HH24:MI:SS"";`
	`RUN`
	`{`
	`ALLOCATE CHANNEL ch11 TYPE DISK MAXPIECESIZE 5G;`
	`ALLOCATE CHANNEL ch12 TYPE DISK MAXPIECESIZE 5G;`
	`BACKUP AS COMPRESSED BACKUPSET FULL DATABASE FORMAT '/efs/rman_backup/FSCM/%d_%T_%s_%p_FULL' ;`
	`SQL "ALTER SYSTEM ARCHIVE LOG CURRENT";`
	`BACKUP FORMAT '/efs/rman_backup/FSCM/%d_%T_%s_%p_ARCHIVE' ARCHIVELOG ALL DELETE ALL INPUT ;`
	`BACKUP CURRENT CONTROLFILE FORMAT '/efs/rman_backup/FSCM/%d_%T_%s_%p_CONTROL';`
	`}`
	`EXIT;`
	`EOF`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan skrip cadangan.	Untuk menjalankan skrip cadangan RMAN, masuk sebagai Oracle Home User, dan jalankan skrip.
	`$  chmod a+x rman_backup.sh`
	`$ ./rman_backup.sh &`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa kesalahan dan catat nama file cadangan.	Periksa file log RMAN untuk kesalahan. Jika semuanya terlihat baik-baik saja, daftarkan cadangan file kontrol dengan menjalankan perintah berikut.
	`RMAN> list backup of controlfile;`
	`using target database control file instead of recovery catalog`
	Perhatikan nama file output.
	`List of Backup Sets ==================`
	`BS Key Type LV Size Device Type Elapsed Time Completion Time ------- ---- -- ---------- --------- ---------- ---------- 12 Full 21.58M DISK 00:00:01 13-JUL-22 BP Key: 12 Status: AVAILABLE Compressed: NO Tag: TAG20220713T150155 Piece Name: /efs/rman_backup/FSCM/FSDMO92_20220713_12_1_CONTROL Control File Included: Ckp SCN: 165591599`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`85898 Ckp time: 13-JUL-22` Anda akan menggunakan file kontrol cadangan `/efs/rman_backup/FSCM/FSDMO92_20220713_12_1_CONTROL` saat memulihkan database di Amazon RDS Custom.	

Matikan tingkat aplikasi sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Matikan aplikasi.	Untuk mematikan tingkat aplikasi sumber, gunakan `psadmin` utilitas atau utilitas baris `psadmin` perintah.
	
	1. Untuk mematikan server web, jalankan perintah berikut.
	
	`psadmin -w shutdown -d "webserver domain name"`
	
	2. Untuk mematikan server aplikasi, jalankan perintah berikut.
	
	`psadmin -c shutdown -d "application server domain name"`
		DBA, Administrator PeopleSoft

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Untuk mematikan penjadwal proses, jalankan perintah berikut. `psadmin -p stop -d "process scheduler domain name"`	

Konfigurasikan basis data Kustom Amazon RDS target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal paket nfs-utils rpm.	Untuk menginstal `nfs-utils rpm` paket, jalankan perintah berikut. `$ yum install -y nfs-utils`	DBA
Pasang penyimpanan EFS.	Dapatkan perintah pemasangan Amazon EFS dari halaman konsol Amazon EFS. Pasang sistem file EFS di instans Amazon RDS dengan menggunakan klien Network File System (NFS). `sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-xxxxxxxxxx.efs.eu-west-1.amazonaws.com:/ /efs`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-xxxxxxxxxx.efs.eu-west-1.amazonaws.com:/ /efs`	

Jatuhkan database starter dan buat direktori untuk menyimpan file database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jeda mode otomatisasi.	Anda harus menjeda mode otomatisasi pada instans Amazon RDS Custom DB sebelum melanjutkan ke langkah berikutnya, untuk memastikan bahwa otomatisasi tidak mengganggu aktivitas pemulihan RMAN. Anda dapat menjeda otomatisasi dengan menggunakan konsol AWS atau perintah AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) (pastikan Anda telah mengonfigurasi AWS CLI terlebih dahulu). `aws rds modify-db-instance \`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--db-instance-id entifier peoplesoft-fscm-92 \ --automation-mode all-paused \ --resume-full-au tomation-mode-minute 360 \ --region eu-west-1` Saat Anda menentukan durasi jeda, pastikan Anda menyisakan cukup waktu untuk pemulihan RMAN. Ini tergantung pada ukuran database sumber, jadi modifikasi nilai 360 yang sesuai. Juga, pastikan bahwa total waktu otomatisasi yang dijeda tidak tumpang tindih dengan jendela cadangan atau pemeliharaan database.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Membuat dan memodifikasi file parameter untuk PeopleSoft	Untuk membuat dan memodifikasi pfile PeopleSoft, gunakan pfile standar yang dibuat dengan instans Amazon RDS Custom DB. Tambahkan parameter yang Anda butuhkan PeopleSoft.
	
	1. Beralih ke `rds user rdsdb` dengan menjalankan perintah berikut.
	
	`$ sudo su - rdsdb`
	
	2. Masuk ke SQL* Plus pada database starter, dan buat pfile dengan menjalankan perintah berikut.
	
	`SQL> create pfile from spfile;`
	
	Ini menciptakan pfile di`$ORACLE_HOME/dbs` .
	3. Buat cadangan dari pfile ini.
	4. Edit pfile untuk menambah atau memperbarui PeopleSoft parameter.
	
	`*._gby_hash_aggregation_enabled=false`
	
	`*._unnest_subquery=false`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`*.nls_language='AMERICAN'`
	`*.nls_length_semantics='CHAR'`
	`*.nls_territory='AMERICA'`
	`*.open_cursors=1000`
	`*.db_files=1200`
	`*.undo_tablespace='UNDOTBS1'`
	PeopleSoft parameter terkait dapat ditemukan di Oracle Support Note 1100831.1.
	5. Hapus referensi spfile dari pfile.
	`*.spfile='/rdsdbbin/oracle/dbs/spfileFSDM092.ora'`	

Jatuhkan basis data starter.	Untuk menghapus database Amazon RDS Custom yang ada, gunakan kode berikut.
	`$ sqlplus / as sysdba`
	`SQL> shutdown immediate;`
	`SQL> startup mount exclusive restrict;`
	`SQL> drop database;`
	`SQL> exit`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan database Amazon RDS Custom dari cadangan.	Kembalikan database dengan menggunakan skrip berikut. Skrip pertama-tama akan mengembalikan file kontrol dan kemudian mengembalikan seluruh database dari bagian cadangan yang disimpan di mount EFS. `#!/bin/bash` `Dt=`date +'%Y%m%d-%H` `%M'` `BACKUP_LOG="rman-${O` `RACLE_SID}-$Dt"` `export TAGDATE=`date +` `%Y%m%d%H%M`;` `LOGPATH=/rdsdbdata/scr` `ipts/logs` `rman target / >>` `$LOGPATH/rman-${OR` `ACLE_SID}-$Dt << EOF` `restore controlfi` `le from "/efs/rma` `n_backup/FSCM/FSDM` `O92_20220713_12_1_` `CONTROL";` `alter database mount;` `run` `{` `set newname for database` `to '/rdsdbdata/db/FSD` `MO92_A/datafile/%f_` `%b';` `SET NEWNAME FOR TEMPFILE` `1 TO '/rdsdbdata/db/` `FSDMO92_A/datafile/%f` `_%b';` `RESTORE DATABASE;` `SWITCH DATAFILE ALL;` `SWITCH TEMPFILE ALL;`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	RECOVER DATABASE; } EOF sqlplus / as sysdba >> $LOGPATH/rman-${OR ACLE_SID}-$Dt<<-EOF ALTER DATABASE RENAME FILE '/u01/psoft/db/ oradata/FSDMO92/redo0 1.log' TO '/rdsdbda ta/db/FSDMO92_A/on linelog/redo01.log'; ALTER DATABASE RENAME FILE '/u01/psoft/db/ oradata/FSDMO92/redo0 2.log' TO '/rdsdbda ta/db/FSDMO92_A/on linelog/redo02.log'; ALTER DATABASE RENAME FILE '/u01/psoft/db/ oradata/FSDMO92/redo0 3.log' TO '/rdsdbda ta/db/FSDMO92_A/on linelog/redo03.log'; alter database clear unarchived logfile group 1; alter database clear unarchived logfile group 2; alter database clear unarchived logfile group 3; alter database open resetlogs; EXIT EOF	

Ambil kata sandi dari Secrets Manager, buat pengguna, dan ubah kata sandi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ambil kata sandi dari Secrets Manager.	Anda dapat melakukan langkah ini dengan menggunakan konsol AWS atau AWS CLI. Langkah-l angkah berikut menunjukkan instruksi untuk konsol. 1. Masuk ke AWS Management Console dan buka konsol Amazon RDS. 2. Di panel navigasi, pilih Database, lalu pilih database Amazon RDS. 3. Pilih tab Konfigurasi, dan catat ID sumber daya untuk instance. Ini akan dalam format `db-<ID>` (misalnya ,`db-73GJNHLGDNZNDOX NWXSECUW6LE`). 4. Buka konsol Secrets Manager. 5. Pilih rahasia yang memiliki nama yang sama dengan`do-not-delete-custom-<resource_id>` , di mana `resource-id` mengacu pada ID sumber daya yang Anda catat di langkah 3. 6. Pilih Ambil nilai rahasia.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Kata sandi ini akan sama untuk`sys`,`system`,`rdsadmin` dan `admin` pengguna.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat pengguna RDSADMIN.	`RDSADMIN`adalah pengguna database untuk memantau dan mengatur instans Amazon RDS Custom DB. Karena basis data starter dijatuhkan dan database target dipulihkan dari sumber menggunakan RMAN, Anda harus membuat ulang pengguna ini setelah operasi pemulihan untuk memastikan bahwa pemantauan Kustom Amazon RDS berfungsi seperti yang diharapkan. Anda juga harus membuat profil dan tablespace terpisah untuk `RDSADMIN` pengguna.
	1. Masukkan perintah berikut pada prompt SQL.
	`SQL> set echo on feedback on serverout on`
	`SQL> @?/rdbms/admin/utlpwdmg.sql`
	`SQL> ALTER PROFILE DEFAULT LIMIT FAILED_LOGIN_ ATTEMPTS UNLIMITED PASSWORD_LIFE_TIME UNLIMITED PASSWORD_VERIFY_F UNCTION NULL;`
	2. Buat profil`RDSADMIN`.
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>SQL> set echo on
	 feedback on serverout
	 on
	SQL> alter session set
	 "_oracle_script"=t
	rue;
	SQL> CREATE PROFILE
	 RDSADMIN
	LIMIT
	COMPOSITE_LIMIT
	 UNLIMITED
	SESSIONS_PER_USER
	 UNLIMITED
	CPU_PER_SESSION
	 UNLIMITED
	CPU_PER_CALL
	 UNLIMITED
	LOGICAL_READS_PER
	_SESSION UNLIMITED
	LOGICAL_READS_PER
	_CALL UNLIMITED
	IDLE_TIME UNLIMITED
	CONNECT_TIME
	 UNLIMITED
	PRIVATE_SGA
	 UNLIMITED
	FAILED_LOGIN_ATTE
	MPTS 10
	PASSWORD_LIFE_TIME
	 UNLIMITED
	PASSWORD_REUSE_TIME
	 UNLIMITED
	PASSWORD_REUSE_MAX
	 UNLIMITED
	PASSWORD_VERIFY_F
	UNCTION NULL
	PASSWORD_LOCK_TIME
	 86400/86400
	PASSWORD_GRACE_TIME
	 604800/86400;</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Buat RDSADMIN tablespace.
	
	`SQL> CREATE BIGFILE TABLESPACE rdsadmin '/rdsdbdata/db/FSDMO92_A/datafile/rdsadmin.dbf' DATAFILE SIZE 7M AUTOEXTEND ON NEXT 1m LOGGING ONLINE PERMANENT BLOCKSIZE 8192 EXTENT MANAGEMENT LOCAL AUTOALLOCATE DEFAULT NOCOMPRESS SEGMENT SPACE MANAGEMENT AUTO;`
	
	4. Buat RDSADMIN pengguna. Ganti RDSADMIN kata sandi dengan kata sandi yang Anda peroleh sebelumnya dari Secrets Manager.
	
	`SQL> CREATE USER rdsadmin IDENTIFIED BY xxxxxxxxxx DEFAULT TABLESPACE rdsadmin TEMPORARY TABLESPACE TEMP profile rdsadmin ;`
	
	5. Berikan hak istimewa kepada RDSADMIN.
	
	`SQL> GRANT "CONNECT" TO RDSADMIN WITH ADMIN OPTION;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	SQL> GRANT "RESOURCE " TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT "DBA" TO RDSADMIN; SQL> GRANT "SELECT_C ATALOG_ROLE" TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT ALTER SYSTEM TO RDSADMIN; SQL> GRANT UNLIMITED TABLESPACE TO RDSADMIN; SQL> GRANT SELECT ANY TABLE TO RDSADMIN; SQL> GRANT ALTER DATABASE TO RDSADMIN; SQL> GRANT ADMINISTER DATABASE TRIGGER TO RDSADMIN; SQL> GRANT ANY OBJECT PRIVILEGE TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT INHERIT ANY PRIVILEGES TO RDSADMIN; SQL> ALTER USER RDSADMIN DEFAULT ROLE ALL; 6. Set the SYS, SYSTEM, and DBSNMP user profiles to RDSADMIN. SQL> set echo on feedback on serverout on	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`SQL> alter user SYS profile RDSADMIN;`
	`SQL> alter user SYSTEM profile RDSADMIN;`
	`SQL> alter user DBSNMP profile RDSADMIN;`	

Buat pengguna master.	Karena basis data starter dijatuhkan dan database target dipulihkan dari sumber dengan menggunakan RMAN, Anda harus membuat ulang pengguna master. Dalam contoh ini, nama pengguna master adalah`admin`.
	`SQL> create user admin identified by <password>;`
	`SQL> grant dba to admin`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ubah kata sandi sistem.	Ubah kata sandi sistem dengan menggunakan kata sandi yang Anda ambil dari Secrets Manager.
	`SQL> alter user sys identified by xxxxxxxxxx;`
	`SQL> alter user system identified by xxxxxxxxxx;`
	Jika Anda tidak mengubah kata sandi ini, Amazon RDS Custom menampilkan pesan kesalahan, “Pemantauan basis data pengguna atau kredensyal pengguna telah berubah.”
		DBA

Konfigurasikan entri TNS untuk Amazon RDS Custom dan PeopleSoft

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan file tnsnames.	Untuk terhubung ke database dari tingkat aplikasi, konfigurasikan `tnsnames.ora` file sehingga Anda dapat terhubung ke database dari tingkat aplikasi. Dalam contoh berikut, Anda dapat melihat bahwa ada tautan lunak ke `tnsnames.ora` file, tetapi file tersebut kosong secara default.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>$ cd /rdsdbbin/oracle/network/admin
	$ ls -ltr
	-rw-r--r-- 1 rdsdb database 1536 Feb 14 2018 shrept.lst
	lrwxrwxrwx 1 rdsdb database 30 Apr 5 13:19 listener.ora -> /rdsdbdata/config/listener.ora
	lrwxrwxrwx 1 rdsdb database 28 Apr 5 13:19 sqlnet.ora -> /rdsdbdata/config/sqlnet.ora
	lrwxrwxrwx 1 rdsdb database 30 Apr 5 13:19 tnsnames.ora -> /rdsdbdata/config/tnsnames.ora</pre>
	1. Buat `tsnames.ora` entri. Karena cara otomatisasi Amazon RDS mem-parsing file, Anda harus memastikan bahwa entri tidak berisi spasi putih, komentar, atau baris tambahan. Jika tidak, Anda mungkin mengalami masalah saat menggunakan beberapa APIs, seperti create-db-instance-read - replica.
	2. Ganti port, host, dan SID sesuai dengan kebutuhan PeopleSoft database Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Gunakan kode berikut sebagai contoh.
	`$ vi tnsnames.ora`
	`FSDMO92=(DESCRIP TION = (ADDRESS_ LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = x.x.x.x)(PORT = 1521))) (CONNECT_ DATA = (SERVER = DEDICATED) (SID = FSDMO92)))`
	3. Untuk mengonfirmasi bahwa PeopleSoft database dapat dijangkau, jalankan perintah berikut.
	`$ tnsping FSDMO92`
	`TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 14-JUL-2022 10:16:45`
	`Copyright (c) 1997, 2021, Oracle. All rights reserved.`
	`Used parameter files: /rdsdbbin/oracle/network/admin/sqlnet.ora`
	`Used TNSNAMES adapter to resolve the alias`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = x.x.x.x)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SID = FSDMO92)))` `OK (0 msec)`	

Buat softlink spfile

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat softlink spfile.	1. Untuk membuat spfile di lokasi`/rdsdbdata/admin/FSDMO92/pfile` , jalankan perintah berikut.
	`SQL> create spfile='/rdsdbdata/admin/FSDMO92/pfile/spfileFSDMO92.ora' from pfile;`
	2. Arahkan ke`$ORACLE_HOME/dbs` , dan buat tautan lunak untuk spfile.
	`ln -s '/rdsdbdata/admin/FSDMO92/pfile/spfileFSDMO92.ora' spfileFSDMO92.ora`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Setelah file ini dibuat, Anda dapat mematikan dan memulai database dengan menggunakan spfile.	

Lakukan langkah pasca-migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi skema, koneksi, dan tugas pemeliharaan.	Untuk menyelesaikan migrasi, lakukan tugas-tugas berikut. • Kumpulkan statistik skema. • Pastikan bahwa tingkat PeopleSoft aplikasi dapat terhubung ke database Amazon RDS Custom yang baru. • Siapkan jadwal pencadangan dan pemeliharaan Anda.	DBA

Sumber daya terkait

- Bekerja dengan Amazon RDS Custom
- Amazon RDS Kustom untuk Oracle - Kemampuan Kontrol Baru di Lingkungan Database (posting blog)
- Mengintegrasikan Amazon RDS Kustom untuk Oracle dengan Amazon EFS (posting blog)
- Mengonfigurasi Amazon RDS sebagai Oracle Database PeopleSoft (whitepaper AWS)

Migrasikan fungsionalitas Oracle ROWID ke PostgreSQL di AWS

Rakesh Raghav dan Ramesh Pathuri, Amazon Web Services

Ringkasan

Pola ini menjelaskan opsi untuk memigrasikan fungsionalitas `ROWID` pseudocolumn di Oracle Database ke database PostgreSQL di Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL, Amazon Aurora PostgreSQL Compatible Edition, atau Amazon Elastic Compute Cloud (Amazon). EC2

Dalam database Oracle, `ROWID` pseudocolumn adalah alamat fisik dari baris dalam tabel. Pseudocolumn ini digunakan untuk mengidentifikasi baris secara unik bahkan jika kunci utama tidak ada di atas meja. PostgreSQL memiliki pseudocolumn serupa yang `ctid` disebut, tetapi tidak dapat digunakan sebagai. `ROWID` Seperti yang dijelaskan dalam dokumentasi PostgreSQL`ctid`, mungkin berubah jika diperbarui atau setelah setiap proses. `VACUUM`

Ada tiga cara Anda dapat membuat fungsionalitas `ROWID` pseudocolumn di PostgreSQL:

- Gunakan kolom kunci primer bukan `ROWID` untuk mengidentifikasi baris dalam tabel.
- Gunakan primary/unique kunci logis (yang mungkin merupakan kunci komposit) dalam tabel.
- Tambahkan kolom dengan nilai yang dihasilkan secara otomatis dan jadikan itu primary/unique kunci untuk meniru`ROWID`.

Pola ini memandu Anda melalui ketiga implementasi dan menjelaskan kelebihan dan kekurangan dari setiap opsi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Keahlian pengkodean prosedural Language/PostgreSQL (PL/PGSQL)
- Sumber Database Oracle
- Kluster Amazon RDS for PostgreSQL atau Aurora PostgreSQL yang kompatibel, atau instance untuk meng-host database PostgreSQL EC2

Batasan

- Pola ini menyediakan solusi untuk fungsionalitas tersebut. `ROWID` PostgreSQL tidak menyediakan yang setara dengan di Oracle Database. `ROWID`

Versi produk

- PostgreSQL 11.9 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Oracle Database

Tumpukan teknologi target

- Kompatibel dengan Aurora PostgreSQL, Amazon RDS for PostgreSQL, atau instance dengan database PostgreSQL EC2

Opsi implementasi

Ada tiga opsi untuk mengatasi kurangnya `ROWID` dukungan di PostgreSQL, tergantung pada apakah tabel Anda memiliki kunci primer atau indeks unik, kunci primer logis, atau atribut identitas. Pilihan Anda bergantung pada jadwal proyek Anda, fase migrasi Anda saat ini, dan dependensi pada aplikasi dan kode database.

Opsi	Deskripsi	Keuntungan	Kekurangan
Kunci primer atau indeks unik	Jika tabel Oracle Anda memiliki kunci utama, Anda dapat menggunakan atribut kunci ini untuk mengidentifikasi baris secara unik.	• Tidak ada ketergantungan pada fitur database berpemilik. • Dampak minimal pada kinerja, karena bidang kunci primer diindeks.	• Memerlukan perubahan pada aplikasi dan kode basis data yang bergantung pada `ROWID` untuk beralih ke bidang kunci utama.

primary/unique Kunci logis	Jika tabel Oracle Anda memiliki kunci primer logis, Anda dapat menggunakan atribut kunci ini untuk mengidentifikasi baris secara unik. Kunci primer logis terdiri dari atribut atau sekumpulan atribut yang secara unik dapat mengidentifikasi baris, tetapi tidak diberlakukan pada database melalui kendala.	• Tidak ada ketergantungan pada fitur database berpemilik.	• Memerlukan perubahan pada aplikasi dan kode basis data yang bergantung pada `ROWID` untuk beralih ke bidang kunci utama. • Dampak signifikan pada kinerja jika atribut kunci primer logis tidak diindeks. Namun, Anda dapat menambahkan indeks unik untuk mencegah masalah kinerja.

Atribut identitas	jika tabel Oracle Anda tidak memiliki kunci utama, Anda dapat membuat bidang tambahan sebagai`GENERATED ALWAYS AS IDENTITY`. Atribut ini menghasilkan nilai unik setiap kali data dimasukkan ke dalam tabel, sehingga dapat digunakan untuk mengident ifikasi secara unik baris untuk operasi Data Manipulation Language (DML/ Data Manipulation Language).	• Tidak ada ketergant ungan pada fitur database berpemili k. • Database PostgreSQL mengisi atribut dan mempertahankan keunikannya.	• Memerlukan perubahan pada aplikasi dan kode database yang bergantung pada `ROWID` untuk beralih ke atribut identitas. • Dampak signifika n pada kinerja jika bidang tambahan tidak diindeks. Namun, Anda dapat menambahk an indeks untuk mencegah masalah kinerja.

Alat

- Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional PostgreSQL di AWS Cloud.
- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda. Dalam pola ini, Anda dapat menggunakan AWS CLI untuk menjalankan perintah SQL melalui pgAdmin.
- pgAdmin adalah alat manajemen sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database.

- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.

Epik

Identifikasi tabel sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi tabel Oracle yang menggunakan `ROWID` atribut.	Gunakan AWS Schema Conversion Tool (AWS SCT) untuk mengidentifikasi tabel Oracle yang memiliki fungsionalitas. `ROWID` Untuk informasi selengkapnya, lihat dokumentasi AWS SCT. — atau — Di Oracle, gunakan `DBA_TAB_COLUMNS` tampilan untuk mengidentifikasi tabel yang memiliki `ROWID` atribut. Bidang ini dapat digunakan untuk menyimpan karakter 10-byte alfanumerik. Tentukan penggunaan dan konversik an ini ke `VARCHAR` bidang jika sesuai.	DBA atau pengembang
Identifikasi kode yang mereferensikan tabel ini.	Gunakan AWS SCT untuk membuat laporan penilaian migrasi guna mengident ifikasi prosedur yang terpengaruh`ROWID`. Untuk informasi selengkapnya, lihat dokumentasi AWS SCT.	DBA atau pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	— atau — Dalam database sumber Oracle, gunakan bidang teks `dba_source` tabel untuk mengidentifikasi objek yang menggunakan `ROWID` fungsionalitas.	

Tentukan penggunaan kunci utama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi tabel yang tidak memiliki kunci utama.	Dalam database sumber Oracle, gunakan `DBA_CONSTRAINTS` untuk mengidentifikasi tabel yang tidak memiliki kunci utama. Informasi ini akan membantu Anda menentukan strategi untuk setiap tabel. Contoh: `select dt.* from dba_tables dt where not exists (select 1 from all_constraints ct where ct.owner = Dt.owner and ct.table_name = Dt.table_name and ct.constraint_type = 'P' )`	DBA atau pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`and dt.owner = '{schema}'`	

Identifikasi dan terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan perubahan untuk tabel yang memiliki kunci primer yang ditentukan atau logis.	Buat perubahan kode aplikasi dan database yang ditampilkan di bagian Informasi tambahan untuk menggunakan kunci primer unik atau kunci primer logis untuk mengidentifikasi baris dalam tabel Anda.	DBA atau pengembang
Tambahkan bidang tambahan ke tabel yang tidak memiliki kunci primer yang ditentukan atau logis.	Tambahkan atribut tipe`GENERATED ALWAYS AS IDENTITY`. Buat perubahan kode aplikasi dan database yang ditampilkan di bagian Informasi tambahan.	DBA atau pengembang
Tambahkan indeks jika perlu.	Tambahkan indeks ke bidang tambahan atau kunci primer logis untuk meningkatkan kinerja SQL.	DBA atau pengembang

Sumber daya terkait

- PostgreSQL CTID (dokumentasi PostgreSQL)
- Kolom yang dihasilkan (dokumentasi PostgreSQL)
- ROWID Pseudocolumn (dokumentasi Oracle)

Informasi tambahan

Bagian berikut memberikan contoh kode Oracle dan PostgreSQL untuk menggambarkan tiga pendekatan.

Skenario 1: Menggunakan kunci unik primer

Dalam contoh berikut, Anda membuat tabel `testrowid_s1` dengan `emp_id` sebagai kunci utama.

Kode Oracle:

```
create table testrowid_s1 (emp_id integer, name varchar2(10), CONSTRAINT testrowid_pk
 PRIMARY KEY (emp_id));
INSERT INTO testrowid_s1(emp_id,name) values (1,'empname1');
INSERT INTO testrowid_s1(emp_id,name) values (2,'empname2');
INSERT INTO testrowid_s1(emp_id,name) values (3,'empname3');
INSERT INTO testrowid_s1(emp_id,name) values (4,'empname4');
commit;

SELECT rowid,emp_id,name FROM testrowid_s1;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3pAAAAAAAMOAAA          1 empname1
AAAF3pAAAAAAAMOAAB          2 empname2
AAAF3pAAAAAAAMOAAC          3 empname3
AAAF3pAAAAAAAMOAAD          4 empname4

UPDATE testrowid_s1 SET name = 'Ramesh' WHERE rowid = 'AAAF3pAAAAAAAMOAAB' ;
commit;

SELECT rowid,emp_id,name FROM testrowid_s1;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3pAAAAAAAMOAAA          1 empname1
AAAF3pAAAAAAAMOAAB          2 Ramesh
AAAF3pAAAAAAAMOAAC          3 empname3
AAAF3pAAAAAAAMOAAD          4 empname4
```

Kode PostgreSQL:

```
CREATE TABLE public.testrowid_s1
(
    emp_id integer,
```

```
    name character varying,
    primary key (emp_id)
);

insert into public.testrowid_s1 (emp_id,name) values
(1,'empname1'),(2,'empname2'),(3,'empname3'),(4,'empname4');

select emp_id,name from testrowid_s1;
 emp_id |   name
--------+----------
      1 | empname1
      2 | empname2
      3 | empname3
      4 | empname4

update testrowid_s1 set name = 'Ramesh' where emp_id = 2 ;

select emp_id,name from testrowid_s1;
 emp_id |   name
--------+----------
      1 | empname1
      3 | empname3
      4 | empname4
      2 | Ramesh
```

Skenario 2: Menggunakan kunci primer logis

Dalam contoh berikut, Anda membuat tabel `testrowid_s2` dengan `emp_id` sebagai kunci utama logis.

Kode Oracle:

```
create table testrowid_s2 (emp_id integer, name varchar2(10) );
INSERT INTO testrowid_s2(emp_id,name) values (1,'empname1');
INSERT INTO testrowid_s2(emp_id,name) values (2,'empname2');
INSERT INTO testrowid_s2(emp_id,name) values (3,'empname3');
INSERT INTO testrowid_s2(emp_id,name) values (4,'empname4');
commit;

SELECT rowid,emp_id,name FROM testrowid_s2;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3rAAAAAAAMeAAA          1 empname1
```

```
AAAF3rAAAAAAAMeAAB          2 empname2
AAAF3rAAAAAAAMeAAC          3 empname3
AAAF3rAAAAAAAMeAAD          4 empname4

UPDATE testrowid_s2 SET name = 'Ramesh' WHERE rowid = 'AAAF3rAAAAAAAMeAAB' ;
commit;

SELECT rowid,emp_id,name FROM testrowid_s2;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3rAAAAAAAMeAAA          1 empname1
AAAF3rAAAAAAAMeAAB          2 Ramesh
AAAF3rAAAAAAAMeAAC          3 empname3
AAAF3rAAAAAAAMeAAD          4 empname4
```

Kode PostgreSQL:

```
CREATE TABLE public.testrowid_s2
(
    emp_id integer,
    name character varying
);

insert into public.testrowid_s2 (emp_id,name) values
(1,'empname1'),(2,'empname2'),(3,'empname3'),(4,'empname4');

select emp_id,name from testrowid_s2;
 emp_id |   name
--------+----------
      1 | empname1
      2 | empname2
      3 | empname3
      4 | empname4

update testrowid_s2 set name = 'Ramesh' where emp_id = 2 ;

select emp_id,name from testrowid_s2;
 emp_id |   name
--------+----------
      1 | empname1
      3 | empname3
      4 | empname4
      2 | Ramesh
```

Skenario 3: Menggunakan atribut identitas

Dalam contoh berikut, Anda membuat tabel `testrowid_s3` tanpa kunci utama dan dengan menggunakan atribut identitas.

Kode Oracle:

```
create table testrowid_s3 (name varchar2(10));
INSERT INTO testrowid_s3(name) values ('empname1');
INSERT INTO testrowid_s3(name) values ('empname2');
INSERT INTO testrowid_s3(name) values ('empname3');
INSERT INTO testrowid_s3(name) values ('empname4');
commit;

SELECT rowid,name FROM testrowid_s3;
ROWID              NAME
------------------ ----------
AAAF3sAAAAAAAMmAAA empname1
AAAF3sAAAAAAAMmAAB empname2
AAAF3sAAAAAAAMmAAC empname3
AAAF3sAAAAAAAMmAAD empname4

UPDATE testrowid_s3 SET name = 'Ramesh' WHERE rowid = 'AAAF3sAAAAAAAMmAAB' ;
commit;

SELECT rowid,name FROM testrowid_s3;
ROWID              NAME
------------------ ----------
AAAF3sAAAAAAAMmAAA empname1
AAAF3sAAAAAAAMmAAB Ramesh
AAAF3sAAAAAAAMmAAC empname3
AAAF3sAAAAAAAMmAAD empname4
```

Kode PostgreSQL:

```
CREATE TABLE public.testrowid_s3
(
    rowid_seq bigint generated always as identity,
    name character varying
);

insert into public.testrowid_s3 (name) values
('empname1'),('empname2'),('empname3'),('empname4');
```

```
select rowid_seq,name from testrowid_s3;
 rowid_seq |   name
-----------+----------
         1 | empname1
         2 | empname2
         3 | empname3
         4 | empname4

update testrowid_s3 set name = 'Ramesh' where rowid_seq = 2 ;

select rowid_seq,name from testrowid_s3;
 rowid_seq |   name
-----------+----------
         1 | empname1
         3 | empname3
         4 | empname4
         2 | Ramesh
```

Migrasikan kode kesalahan Oracle Database ke database yang kompatibel dengan Amazon Aurora PostgreSQL

Sai Parthasaradhi dan Veeranjaneyulu Grandhi, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara memigrasikan kode kesalahan Oracle Database ke database Amazon Aurora PostgreSQL Compatible Edition dengan menggunakan tabel metadata yang telah ditentukan sebelumnya.

Kode kesalahan Oracle Database tidak selalu memiliki kode kesalahan PostgreSQL yang sesuai. Perbedaan kode kesalahan ini dapat menyulitkan untuk mengkonfigurasi logika pemrosesan prosedur atau fungsi dalam arsitektur PostgreSQL target.

Anda dapat menyederhanakan proses dengan menyimpan kode kesalahan basis data sumber dan target yang berarti bagi PL/pgSQL program Anda dalam tabel metadata. Kemudian, konfigurasikan tabel untuk menandai kode kesalahan Oracle Database yang valid dan petakan ke ekuivalen PostgreSQL mereka sebelum melanjutkan dengan logika proses yang tersisa. Jika kode kesalahan Oracle Database tidak ada dalam tabel metadata, proses keluar dengan pengecualian. Kemudian, Anda dapat meninjau detail kesalahan secara manual dan menambahkan kode kesalahan baru ke tabel jika program Anda memerlukannya.

Dengan menggunakan konfigurasi ini, database Amazon Aurora PostgreSQL yang kompatibel dapat menangani kesalahan dengan cara yang sama seperti database Oracle sumber Anda.

> **Note**
>
> Mengkonfigurasi database PostgreSQL untuk menangani kode kesalahan Oracle Database dengan benar biasanya memerlukan perubahan pada database dan kode aplikasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sumber Oracle Database dengan layanan instance dan listener aktif dan berjalan
- Cluster yang kompatibel dengan Amazon Aurora PostgreSQL yang aktif dan berjalan

- Keakraban dengan Oracle Database
- Keakraban dengan database PostgreSQL

Arsitektur

Diagram berikut menunjukkan contoh alur kerja database yang kompatibel dengan Amazon Aurora PostgreSQL untuk validasi dan penanganan kode kesalahan data:

Diagram menunjukkan alur kerja berikut:

1. Sebuah tabel berisi kode kesalahan Oracle Database dan klasifikasi dan kode kesalahan PostgreSQL yang setara dan klasifikasi. Tabel ini mencakup kolom valid_error yang mengklasifikasikan jika kode kesalahan tertentu yang telah ditentukan valid atau tidak.
2. Ketika sebuah PL/pgSQL fungsi (func_processdata) melempar pengecualian, ia memanggil fungsi kedua PL/pgSQL (error_validation).
3. Fungsi error_validation menerima kode kesalahan Oracle Database sebagai argumen masukan. Kemudian, fungsi memeriksa kode kesalahan yang masuk terhadap tabel untuk melihat apakah kesalahan termasuk dalam tabel.
4. Jika kode kesalahan Oracle Database disertakan dalam tabel, maka fungsi error_validation mengembalikan nilai TRUE dan logika proses berlanjut. Jika kode kesalahan tidak disertakan dalam tabel, maka fungsi mengembalikan nilai FALSE, dan logika proses keluar dengan pengecualian.
5. Ketika fungsi mengembalikan nilai FALSE, maka rincian kesalahan secara manual ditinjau oleh lead fungsional aplikasi untuk menentukan validitasnya.
6. Kode kesalahan baru kemudian ditambahkan secara manual ke tabel atau tidak. Jika kode kesalahan valid dan ditambahkan ke tabel, maka fungsi error_validation mengembalikan nilai TRUE saat berikutnya pengecualian terjadi. Jika kode kesalahan tidak valid, dan proses harus gagal ketika pengecualian terjadi, maka kode kesalahan tidak ditambahkan ke tabel.

Tumpukan teknologi

- Amazon Aurora PostgreSQL
- pgAdmin
- Pengembang Oracle SQL

Alat

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- pgAdmin adalah alat administrasi dan pengembangan sumber terbuka untuk PostgreSQL. Ini menyediakan antarmuka grafis yang menyederhanakan pembuatan, pemeliharaan, dan penggunaan objek database.
- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi gratis yang menyederhanakan pengembangan dan pengelolaan Oracle Database baik dalam penerapan tradisional maupun cloud.

Epik

Migrasikan kode kesalahan Oracle Database ke database yang kompatibel dengan Amazon Aurora PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel di database Amazon Aurora PostgreSQL yang kompatibel.	Jalankan perintah PostgreSQL CREATE TABLE berikut: `( source_error_code numeric NOT NULL, target_error_code character varying NOT NULL, valid_error character varying(1) NOT NULL );`	Pengembang PostgreSQL, Oracle, untuk PostgreSQL RDS/Aurora

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan kode kesalahan PostgreSQL dan kode kesalahan Oracle Database yang sesuai ke tabel.	Jalankan perintah PostgreSQL INSERT untuk menambahkan nilai kode kesalahan yang diperlukan ke tabel error_codes.
	
	Kode kesalahan PostgreSQL harus menggunakan karakter yang bervariasi tipe data (nilai SQLSTATE). Kode kesalahan Oracle harus menggunakan tipe data numerik (nilai SQLCODE).
	
	Contoh Sisipkan pernyataan:
	
	`insert into error_codes values (-1817,'22007','Y');`
	`insert into error_codes values (-1816,'22007','Y');`
	`insert into error_codes values (-3114,'08006','N');`
	
	Note
	Jika Anda menangkap pengecualian konektivitas basis data Java (JDBC) khusus Oracle, Anda harus mengganti pengecualian tersebut dengan pengecualian lintas basis data generik
		Pengembang PostgreSQL, Oracle, untuk PostgreSQL RDS/Aurora

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	atau beralih ke pengecualian khusus PostgreSQL.	
Buat PL/pgSQL fungsi untuk memvalidasi kode kesalahan.	Buat PL/pgSQL fungsi dengan menjalankan perintah PostgreSQL CREATE FUNCTION. Pastikan bahwa fungsi melakukan hal berikut: • Menerima kode kesalahan Oracle yang dilemparkan oleh sebuah program. • Memeriksa apakah kode kesalahan hadir dalam tabel error_codes. • Mengembalikan nilai TRUE atau FALSE, berdasarkan jika kode kesalahan hadir dalam tabel metadata atau tidak.	Pengembang PostgreSQL, Oracle, untuk PostgreSQL RDS/Aurora

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau kode kesalahan baru secara manual saat direkam oleh PL/pgSQL fungsi.	Tinjau kode kesalahan baru secara manual. Jika kode kesalahan baru valid untuk kasus penggunaan Anda, tambahkan ke tabel error_codes dengan menjalankan perintah PostgreSQL INSERT. -atau- Jika kode kesalahan baru tidak valid untuk kasus penggunaan Anda, jangan tambahkan ke tabel. Logika proses akan terus gagal dan keluar dengan pengecualian ketika kesalahan terjadi.	Pengembang PostgreSQL, Oracle, untuk PostgreSQL RDS/Aurora

Sumber daya terkait

Lampiran A. PostgreSQL Kode Kesalahan (dokumentasi PostgreSQL)

Pesan kesalahan database (dokumentasi Oracle Database)

Migrasikan SAP ASE di Amazon EC2 ke Amazon Aurora PostgreSQL yang kompatibel dengan menggunakan AWS SCT dan AWS DMS

Amit Kumar dan Ankit Gupta, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database SAP Adaptive Server Enterprise (SAP ASE) yang di-host di instans Amazon Elastic Compute Cloud (Amazon) ke EC2 Amazon Aurora PostgreSQL Compatible Edition dengan menggunakan AWS Schema Conversion Tool (AWS SCT) dan AWS Database Migration Service (AWS DMS). Pola ini berfokus pada konversi bahasa definisi data (DDL) untuk objek yang disimpan dan migrasi data.

Kompatibel dengan Aurora PostgreSQL mendukung beban kerja pemrosesan transaksi online (OLTP). Layanan terkelola ini menyediakan konfigurasi yang secara otomatis menskalakan sesuai permintaan. Ini dapat secara otomatis memulai, mematikan, meningkatkan, atau menurunkan basis data Anda berdasarkan kebutuhan aplikasi Anda. Anda dapat menjalankan database Anda di cloud tanpa mengelola instance database apa pun. Kompatibel dengan Aurora PostgreSQL menyediakan opsi hemat biaya untuk beban kerja yang jarang, intermiten, atau tidak dapat diprediksi.

Proses migrasi terdiri dari dua fase utama:

- Mengonversi skema database dengan menggunakan AWS SCT
- Migrasi data dengan menggunakan AWS DMS

Instruksi terperinci untuk kedua fase disediakan di bagian Epik. Untuk informasi tentang masalah pemecahan masalah yang spesifik untuk menggunakan AWS DMS dengan database SAP ASE, lihat Memecahkan masalah dengan SAP ASE dalam dokumentasi AWS DMS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber SAP ASE pada sebuah EC2 instance dengan server, database, dan layanan listener yang aktif dan berjalan
- Basis data target yang kompatibel dengan Aurora PostgreSQL

Batasan

- Nomor port untuk koneksi harus 5432.
- Fitur huge_pages aktif secara default tetapi dapat dimodifikasi.
- Point-in-time granularitas pemulihan (PITR) adalah 5 menit.
- Replikasi Lintas Wilayah saat ini tidak tersedia.
- Ukuran penyimpanan maksimum untuk database Aurora adalah 128 TiB.
- Anda dapat membuat hingga 15 replika baca.
- Batas ukuran tabel hanya dibatasi oleh ukuran volume cluster Aurora, sehingga ukuran tabel maksimum untuk cluster DB yang kompatibel dengan Aurora PostgreSQL adalah 32 TiB. Kami menyarankan Anda mengikuti praktik terbaik untuk desain meja, seperti mempartisi tabel besar.

Versi produk

- Database sumber: AWS DMS saat ini mendukung SAP ASE 15, 15.5, 15.7, dan 16.x. Lihat Panduan Pengguna AWS DMS untuk informasi terbaru tentang dukungan versi SAP ASE.
- Basis data target: PostgreSQL 9.4 dan yang lebih baru (untuk versi 9.x), 10.x, 11.x, 12.x, 13.x, dan 14.x. Lihat Panduan Pengguna AWS DMS untuk versi PostgreSQL terbaru yang didukung.
- Amazon Aurora 1.x atau lebih baru. Untuk informasi terbaru, lihat Rilis dan versi mesin yang kompatibel dengan Aurora PostgreSQL dalam dokumentasi Aurora.

Arsitektur

Tumpukan teknologi sumber

- Database SAP ASE berjalan di Amazon EC2

Tumpukan teknologi target

- Database yang kompatibel dengan Aurora PostgreSQL

Arsitektur migrasi

Alat

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- AWS Schema Conversion Tool (AWS SCT) mendukung migrasi database heterogen dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target.
- AWS DMS mendukung beberapa basis data sumber dan target yang berbeda. Untuk informasi selengkapnya, lihat Sumber untuk Migrasi Data dan Target untuk Migrasi Data dalam dokumentasi AWS DMS. Untuk versi dan dukungan fitur yang paling komprehensif, kami sarankan Anda menggunakan AWS DMS versi terbaru.

Epik

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan akses jaringan dalam EC2 contoh sumber.	Siapkan grup keamanan dalam EC2 instance yang menghosting basis data SAP ASE sumber Anda. Untuk petunjuk, lihat Grup EC2 keamanan Amazon untuk instans Linux dalam EC2 dokumentasi Amazon.	Administrator sistem
Buat cluster DB yang kompatibel dengan Aurora PostgreSQL target Anda.	Instal, konfigurasikan, dan luncurkan cluster yang kompatibel dengan Aurora PostgreSQL untuk database target Anda. Untuk informasi selengkap nya, lihat Membuat klaster DB	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon Aurora di dokumentasi Aurora.	
Siapkan otorisasi untuk cluster DB target.	Siapkan grup keamanan dan firewall untuk database target. Untuk petunjuknya, lihat Membuat klaster DB Amazon Aurora di dokumentasi Aurora.	DBA, Administrator sistem

Konversikan skema database Anda dengan AWS SCT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan AWS SCT.	Luncurkan AWS SCT dengan mengikuti petunjuk dalam dokumentasi AWS SCT. AWS SCT menyediakan antarmuka pengguna berbasis proyek untuk secara otomatis mengonversi skema database database sumber SAP ASE Anda ke dalam format yang kompatibel dengan instans DB yang kompatibel dengan Aurora PostgreSQL target Anda.	DBA
Buat titik akhir AWS SCT.	Buat titik akhir untuk sumber SAP ASE dan target database PostgreSQL. Untuk petunjuk, lihat dokumentasi AWS SCT.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat laporan penilaian.	Buat laporan penilaian migrasi database untuk mengevalu asi migrasi dan mendeteksi objek dan fungsi yang tidak kompatibel. Untuk petunjuk, lihat dokumentasi AWS SCT.	DBA
Konversi skema.	Konversikan skema database dengan mengikuti petunjuk dalam dokumentasi AWS SCT.	DBA
Validasi objek database.	Jika AWS SCT tidak dapat mengonversi objek database, AWS SCT akan mengidentifikasi namanya dan detail lainnya. Anda harus mengonversi benda-benda ini secara manual. Untuk mengidentifikasi ketidakcocokan ini, ikuti petunjuk di postingan blog AWS Validasi objek database setelah bermigrasi dari SAP ASE ke Amazon RDS for PostgreSQL atau Amazon Aurora PostgreSQL.	DBA

Menganalisis migrasi AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target.	Periksa versi database SAP ASE untuk kompatibilitas dengan AWS DMS. Untuk informasi selengkapnya, lihat Sumber untuk AWS DMS dan Target untuk AWS DMS dalam dokumentasi AWS DMS.	DBA
Identifikasi persyaratan untuk jenis dan kapasitas penyimpanan.	Pilih kapasitas penyimpanan yang sesuai untuk basis data target berdasarkan ukuran basis data sumber Anda.	DBA, Administrator sistem
Pilih jenis instans, kapasitas, dan fitur lain dari instance replikasi.	Pilih jenis instans, kapasitas, fitur penyimpanan, dan fitur jaringan yang memenuhi kebutuhan Anda. Untuk panduan, lihat Memilih instans replikasi AWS DMS yang tepat untuk migrasi Anda dalam dokumentasi AWS DMS.	DBA, Administrator sistem
Identifikasi persyaratan keamanan akses jaringan.	Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target. Ikuti panduan dalam Menyiapkan jaringan untuk	DBA, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	instance replikasi dalam dokumentasi AWS DMS.	

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan membuat tugas migrasi di AWS DMS.	Untuk memigrasikan data, buat tugas dan ikuti instruksi dalam dokumentasi AWS DMS. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif.	DBA
Validasi data.	Untuk memvalidasi bahwa data Anda dimigrasikan secara akurat dari database sumber ke database target, ikuti pedoman validasi data yang disediakan dalam dokumentasi AWS DMS.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi strategi migrasi aplikasi.	Pilih salah satu dari tujuh strategi (7R) untuk memigrasikan aplikasi ke cloud.	DBA, Pemilik aplikasi, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.	Selesaikan tugas database yang diidentifikasi oleh tim aplikasi, termasuk memperbarui detail koneksi DNS untuk database target dan memperbarui kueri dinamis.	DBA, Pemilik aplikasi, Administrator sistem

Potong ke database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.	Alihkan koneksi dari database sumber ke database target. Untuk informasi selengkap nya, lihat bagian Cut over dari strategi Migrasi untuk database relasional.	DBA, Pemilik aplikasi, Administrator sistem

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.	Hentikan semua tugas migrasi, instance replikasi, titik akhir, dan sumber daya AWS SCT dan AWS DMS lainnya. Untuk informasi selengkapnya, lihat dokumentasi AWS DMS.	DBA, Administrator sistem
Tinjau dan validasi dokumen proyek.	Validasi semua langkah dalam dokumentasi proyek untuk	DBA, Pemilik aplikasi, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memastikan bahwa semua tugas telah berhasil diselesai kan.	
Tutup proyek.	Tutup proyek migrasi dan berikan umpan balik apa pun.	DBA, Pemilik aplikasi, Administrator sistem

Sumber daya terkait

Referensi

- Aktifkan koneksi terenkripsi untuk instans PostgreSQL DB di Amazon RDS (AWS Prescriptive Guidance)
- Transportasi database PostgreSQL antara dua instans Amazon RDS DB menggunakan pg_transport (AWS Prescriptive Guidance)
- Harga Amazon Aurora
- Praktik terbaik dengan Amazon Aurora PostgreSQL Edisi yang kompatibel (dokumentasi Amazon Aurora)
- Dokumentasi AWS SCT
- Dokumentasi AWS DMS
- Menggunakan Database SAP ASE sebagai Sumber AWS DMS

Tutorial dan video

- Memulai AWS Database Migration Service
- AWS Database Migration Service (video)

Migrasikan sertifikat SSL Windows ke Application Load Balancer menggunakan ACM

Chandra Sekhar Yaratha dan Igor Kovalchuk, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk menggunakan AWS Certificate Manager (ACM) untuk memigrasikan sertifikat Secure Sockets Layer (SSL) yang ada dari situs web yang dihosting di server lokal atau instans Amazon Elastic Compute Cloud (Amazon EC2) di Microsoft Internet Information Services (IIS). Sertifikat SSL kemudian dapat digunakan dengan Elastic Load Balancing di AWS.

SSL melindungi data Anda, menegaskan identitas Anda, memberikan peringkat mesin pencari yang lebih baik, membantu memenuhi persyaratan Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS), dan meningkatkan kepercayaan pelanggan. Pengembang dan tim TI yang mengelola beban kerja ini ingin aplikasi web dan infrastruktur mereka, termasuk server IIS dan Windows Server, untuk tetap mematuhi kebijakan dasar mereka.

Pola ini mencakup ekspor sertifikat SSL yang ada secara manual dari Microsoft IIS, mengonversinya dari format Personal Information Exchange (PFX) ke format Private Enhanced Mail (PEM) yang didukung ACM, dan kemudian mengimpornya ke ACM di akun AWS Anda. Ini juga menjelaskan cara membuat Application Load Balancer untuk aplikasi Anda dan mengonfigurasi Application Load Balancer untuk menggunakan sertifikat impor Anda. Koneksi HTTPS kemudian dihentikan pada Application Load Balancer, dan Anda tidak memerlukan overhead konfigurasi lebih lanjut di server web. Untuk informasi selengkapnya, lihat Membuat pendengar HTTPS untuk Application Load Balancer Anda.

Server Windows menggunakan file.pfx atau.p12 untuk berisi file kunci publik (sertifikat SSL) dan file kunci pribadi uniknya. Otoritas Sertifikat (CA) memberi Anda file kunci publik Anda. Anda menggunakan server Anda untuk menghasilkan file kunci pribadi terkait tempat permintaan penandatanganan sertifikat (CSR) dibuat.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Virtual private cloud (VPC) di AWS dengan setidaknya satu subnet pribadi dan satu subnet publik di setiap Availability Zone yang digunakan oleh target Anda
- IIS versi 8.0 atau yang lebih baru berjalan di Windows Server 2012 atau yang lebih baru

- Aplikasi web yang berjalan di IIS
- Akses administrator ke server IIS

Arsitektur

Tumpukan teknologi sumber

- Implementasi server web IIS dengan SSL untuk memastikan bahwa data ditransmisikan dengan aman dalam koneksi terenkripsi (HTTPS)

Arsitektur sumber

Tumpukan teknologi target

- Sertifikat ACM di akun AWS Anda
- Application Load Balancer yang dikonfigurasi untuk menggunakan sertifikat yang diimpor
- Instans Windows Server di subnet pribadi

Arsitektur target

Alat

- AWS Certificate Manager (ACM) membantu Anda membuat, menyimpan, dan memperbarui sertifikat dan kunci SSL/TLS X.509 publik dan pribadi yang melindungi situs web dan aplikasi AWS Anda.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh EC2 instans, kontainer, dan alamat IP di satu atau beberapa Availability Zone.

Praktik terbaik

- Menerapkan pengalihan lalu lintas dari HTTP ke HTTPS.
- Konfigurasikan grup keamanan untuk Application Load Balancer Anda dengan benar untuk mengizinkan lalu lintas masuk hanya ke port tertentu.

- Luncurkan EC2 instans Anda di Availability Zone yang berbeda untuk memastikan ketersediaan tinggi.
- Konfigurasikan domain aplikasi Anda untuk menunjuk ke nama DNS Application Load Balancer, bukan alamat IP-nya.
- Pastikan Application Load Balancer memiliki pemeriksaan kesehatan lapisan aplikasi yang dikonfigurasi.
- Konfigurasikan ambang batas untuk pemeriksaan kesehatan.
- Gunakan Amazon CloudWatch untuk memantau Application Load Balancer.

Epik

Ekspor file.pfx

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekspor file.pfx dari Windows Server.	Untuk mengekspor sertifikat SSL sebagai file.pfx dari manajer IIS lokal di Windows Server: 1. Pilih Manajer Mulai, Administratif, Layanan Informasi Internet (IIS). 2. Pilih nama server, dan di bawah Keamanan, klik dua kali Sertifikat Server. 3. Pilih sertifikat yang ingin Anda ekspor, lalu pilih Ekspor. 4. Di kotak Ekspor Sertifikat, pilih lokasi, jalur, dan nama untuk file.pfx Anda. 5. Tentukan dan konfirmasikan kata sandi untuk file.pfx Anda.	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Anda memerlukan kata sandi ini saat menginstal file.pfx. 6. Pilih OK. File.pfx Anda sekarang harus disimpan ke lokasi dan jalur yang Anda tentukan.	

Ubah sertifikat yang dikodekan PFX ke format PEM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh dan instal toolkit OpenSSL.	1. Unduh dan instal Win32/Win64 OpenSSL dari situs web Shining Light Productions. 2. Tambahkan lokasi binari OpenSSL ke variabel `PATH` sistem Anda, sehingga binari dapat tersedia untuk penggunaan baris perintah.	Administrator sistem
Ubah sertifikat yang dikodekan PFX ke format PEM.	Langkah-langkah berikut mengonversi file sertifikat yang ditandatangani PFX yang dikodekan menjadi tiga file dalam format PEM:	Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `cert-file.pem` berisi SSL/TLS sertifikat untuk sumber daya. • `privatekey.pem` berisi kunci pribadi sertifikat tanpa perlindungan kata sandi. • `ca-chain.pem` berisi sertifikat root CA. Untuk mengonversi sertifikat yang dikodekan PFX: 1. Jalankan Windows PowerShell. 2. Gunakan perintah berikut untuk mengekstrak kunci pribadi sertifikat dari file PFX. Masukkan kata sandi sertifikat saat diminta. `openssl pkcs12 -in <filename>.pfx -nocerts -out withpw-privatekey.pem` Perintah menghasilkan file kunci pribadi yang dikodekan PEM bernama. `privatekey.pem` Masukkan frasa sandi untuk melindungi file kunci pribadi saat diminta. 3. Jalankan perintah berikut untuk menghapus frasa	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	sandi. Saat diminta, berikan frasa sandi yang Anda buat di langkah 2.
	`openssl rsa -in withpw-privatekey.pem -out privatekey.pem`
	Jika perintah berhasil, itu menampilkan pesan “menulis kunci RSA.”
	4. Gunakan perintah berikut untuk mentransfer sertifikat dari file PFX ke file PEM.
	`openssl pkcs12 -in <file_name>.pfx -clcerts -nokeys -out cert-file.pem`
	Ini membuat file sertifikat yang dikodekan PEM bernama. `cert-file.pem` Jika perintah berhasil, itu menampilkan pesan “MAC diverifikasi OK.”
	5. Buat file rantai CA dari file PFX. Perintah berikut membuat file rantai CA bernama`ca-chain.pem` .
	`openssl pkcs12 -in <file_name>.pfx -`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`cacerts -nokeys -chain -out ca-chain.pem` Jika perintah berhasil, itu menampilkan pesan “MAC diverifikasi OK.”	

Impor sertifikat ke ACM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersiaplah untuk mengimpor sertifikat.	Di konsol ACM, pilih Impor sertifikat.	Administrator awan
Berikan badan sertifikat.	Untuk badan Sertifikat, tempel sertifikat yang disandikan PEM yang ingin Anda impor. Untuk informasi selengkapnya tentang perintah dan langkah yang dijelaskan dalam tugas ini dan tugas lain dalam epik ini, lihat Mengimpor sertifikat dalam dokumentasi ACM.	Administrator awan
Berikan kunci pribadi sertifikat.	Untuk kunci privat Sertifikat, tempel kunci privat tak terenkripsi yang disandikan PEM yang cocok dengan kunci publik sertifikat.	Administrator awan
Berikan rantai sertifikat.	Untuk rantai Sertifikat, tempel rantai sertifikat yang dikodekan PEM,	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang disimpan dalam file. `CertificateChain.pem`	
Impor sertifikat.	Pilih Tinjau dan impor. Konfirmasikan bahwa informasi tentang sertifikat Anda sudah benar, lalu pilih Impor.	Administrator awan

Membuat Application Load Balancer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan penyeimbang beban dan pendengar.	Ikuti petunjuk dalam dokumentasi Elastic Load Balancing untuk mengonfig urasi grup target, mendaftar kan target, dan membuat Application Load Balancer dan pendengar. Tambahkan pendengar kedua (HTTPS) untuk port 443.	Administrator awan

Pemecahan masalah

Isu	Solusi
Windows PowerShell tidak mengenali perintah OpenSSL bahkan setelah Anda menambahk annya ke jalur sistem.	Periksa `$env:path` untuk memastikan bahwa itu termasuk lokasi binari OpenSSL. Jika tidak, jalankan perintah berikut di PowerShell:

Isu	Solusi
	`$env:path = $env:path + ";C:\OpenSSL-Win64\bin"`

Sumber daya terkait

Mengimpor sertifikat ke ACM

- Konsol ACM
- Sertifikat dan format kunci untuk mengimpor
- Mengimpor sertifikat
- Panduan Pengguna AWS Certificate Manager

Membuat Application Load Balancer

- Buat Application Load Balancer
- Panduan Pengguna Application Load Balancer

Memigrasikan antrian pesan dari Microsoft Azure Service Bus ke Amazon SQS

Nisha Gambhir, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan aplikasi web atau konsol .NET Framework atau .NET Core dari menggunakan platform pesan antrian Microsoft Azure Service Bus ke Amazon Simple Queue Service (Amazon SQS).

Aplikasi menggunakan layanan pesan untuk mengirim data ke, dan menerima data dari, aplikasi lain. Layanan ini membantu membangun layanan mikro terpisah, sangat skalabel, sistem terdistribusi, dan aplikasi tanpa server di cloud.

Antrian Azure Service Bus adalah bagian dari infrastruktur pesan Azure yang lebih luas yang mendukung antrian dan pengiriman pesan. publish/subscribe

Amazon SQS adalah layanan antrian pesan yang dikelola sepenuhnya yang memungkinkan Anda memisahkan dan menskalakan layanan mikro, sistem terdistribusi, dan aplikasi tanpa server. Amazon SQS menghilangkan kompleksitas dan overhead yang terkait dengan pengelolaan dan pengoperasian middleware yang berorientasi pesan, dan memungkinkan pengembang untuk fokus pada membedakan pekerjaan. Menggunakan Amazon SQS, Anda dapat mengirim, menyimpan, dan menerima pesan antar komponen perangkat lunak pada volume berapa pun, tanpa kehilangan pesan atau mengharuskan layanan lain tersedia.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Aplikasi web atau konsol .NET Framework atau .NET Core yang menggunakan antrian Azure Service Bus (kode contoh terlampir)

Versi produk

- .NET Framework 3.5 atau yang lebih baru, atau.NET Core 1.0.1, 2.0.0, atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Aplikasi web atau konsol .NET (Core atau Framework) yang menggunakan antrian Azure Service Bus untuk mengirim pesan

Tumpukan teknologi target

- Amazon SQS

Alat

Alat

- Microsoft Visual Studio

Kode

Untuk membuat kebijakan AWS Identity and Access management (IAM) untuk Amazon SQS:

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

2. Pada panel navigasi yang ada di sebelah kiri, pilih Kebijakan, lalu pilih Buat kebijakan.

3. Pilih tab JSON, dan tempel kode berikut:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": [
            "sqs:DeleteMessage",
            "sqs:GetQueueUrl",
            "sqs:ChangeMessageVisibility",
            "sqs:SendMessageBatch",
            "sqs:ReceiveMessage",
            "sqs:SendMessage",
            "sqs:GetQueueAttributes",
            "sqs:ListQueueTags",
            "sqs:ListDeadLetterSourceQueues",
```

```
            "sqs:DeleteMessageBatch",
            "sqs:PurgeQueue",
            "sqs:DeleteQueue",
            "sqs:CreateQueue",
            "sqs:ChangeMessageVisibilityBatch",
            "sqs:SetQueueAttributes"
         ],
         "Resource": "arn:aws:sqs:*:<AccountId>:*"
      },
      {
         "Sid": "VisualEditor1",
         "Effect": "Allow",
         "Action": "sqs:ListQueues",
         "Resource": "*"
      }
   ]
}
```

4. Pilih Kebijakan tinjauan, ketik nama, lalu pilih Buat kebijakan.

5. Lampirkan kebijakan yang baru dibuat ke peran IAM Anda yang ada atau buat peran baru.

Epik

Siapkan Amazon SQS di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM untuk Amazon SQS.	Buat kebijakan IAM yang akan menyediakan akses ke Amazon SQS. Lihat bagian Kode untuk kebijakan sampel.	Insinyur sistem
Buat profil AWS.	Buat profil baru dengan menjalankan AWS Tools untuk PowerShell perintah Set-AWSCredential. Perintah ini menyimpan kunci akses dan kunci rahasia Anda dalam file kredensi default Anda di bawah nama profil yang Anda	Insinyur sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tentukan. Tautkan kebijakan Amazon SQS yang Anda buat sebelumnya dengan akun ini. Simpan ID kunci akses AWS dan kunci akses rahasia. Ini akan diperlukan pada langkah selanjutnya.	
Buat antrian SQS.	Anda dapat membuat antrian standar atau antrian masuk pertama, keluar pertama (FIFO). Untuk petunjuk, lihat tautan di bagian Referensi.	Insinyur sistem

Merevisi kode aplikasi.NET Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS Toolkit for Visual Studio.	Toolkit ini adalah ekstensi untuk Microsoft Visual Studio dan memudahkan Anda untuk membangun dan menyebark an aplikasi.NET di AWS. Untuk petunjuk pemasangan dan penggunaan, lihat tautan di bagian Referensi.	Pengembang aplikasi
Instal AWSSDK paket.SQS. NuGet	Anda dapat menginstal AWSSDK .SQS dengan memilih “Manage NuGet Package” di Visual Studio atau dengan menjalankan perintah “ AWSSDKInstall-Package .SQS”.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AWSCredentials objek di aplikasi.NET Anda.	Contoh aplikasi dalam lampiran menunjukkan cara membuat AWSCredentials objek Basic, yang mewarisi dari AWSCredentials. Anda dapat menggunakan ID kunci akses dan kunci akses rahasia dari sebelumnya, atau biarkan objek memilih ini dari folder.aws sebagai bagian dari profil pengguna saat dijalankan.	Pengembang aplikasi
Buat objek klien SQS.	Buat objek klien SQS (AmazonSQSClient) untuk.NET Framework. Ini adalah bagian dari namespace Amazon.sqs. Objek ini diperlukan alih-alih IQueue Klien, yang merupakan bagian dari Microsoft.Azure. ServiceBus namespace.	Pengembang aplikasi
Panggil SendMessageAsync metode untuk mengirim pesan ke antrian SQS.	Ubah kode yang mengirim pesan ke antrian untuk menggunakan. amazonSqs Client SendMessageAsync metode. Untuk detailnya, lihat contoh kode terlampir.	Pengembang aplikasi
Panggil ReceiveMessageAsync metode untuk menerima pesan dari antrian SQS.	Ubah kode yang menerima pesan untuk menggunakan file amazonSqsClient. ReceiveMessageAsync metode. Untuk detailnya, lihat contoh kode terlampir.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Panggil DeleteMessageAsync metode untuk menghapus pesan dari antrian SQS.	Untuk menghapus pesan, ubah kode dari QueueClient. CompleteAsync metode untuk amazonSqsClient. DeleteMes sageAsync metode. Untuk detailnya, lihat contoh kode terlampir.	Pengembang aplikasi

Sumber daya terkait

- Panduan Pengembang AWS SDK for .NET
- Pesan Menggunakan Amazon SQS
- Membuat dan Menggunakan Antrian Amazon SQS dengan AWS SDK for .NET
- Kirim Pesan Amazon SQS
- Menerima Pesan dari Antrian Amazon SQS
- Menghapus Pesan dari Antrian Amazon SQS
- AWS Toolkit for Visual Studio

Informasi tambahan

Pola ini mencakup dua contoh aplikasi (lihat bagian lampiran):

- AzureSbTestApptermasuk kode yang menggunakan antrian Azure Service Bus.
- AmazonSqsTestAppmenggunakan Amazon SQS. Ini adalah aplikasi konsol yang menggunakan.NET Core 2.2 dan termasuk contoh untuk mengirim dan menerima pesan.

Catatan:

- QueueClient adalah objek IQueue Klien, yang merupakan bagian dari Microsoft.Azure. ServiceBus namespace (termasuk dalam Microsoft.Azure. ServiceBus NuGet paket).
- amazonSqsClient adalah objek AmazonSQSClient, yang merupakan bagian dari namespace Amazon.sqs (termasuk dalam paket.SQS). AWSSDK NuGet

- Tergantung di mana kode berjalan, katakanlah jika sedang berjalan EC2, peran perlu memiliki izin untuk menulis ke dalam SQS Queue.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Migrasikan database Oracle JD Edwards EnterpriseOne ke AWS dengan menggunakan Oracle Data Pump dan AWS DMS

Thanigaivel Thirumalai, Amazon Web Services

Ringkasan

Anda dapat memigrasi dan menjalankan database JD Edwards di Amazon EnterpriseOne Relational Database Service (Amazon RDS). Saat memigrasikan database ke Amazon RDS, AWS dapat menangani tugas pencadangan dan penyiapan ketersediaan tinggi, sehingga Anda dapat berkonsentrasi pada pemeliharaan EnterpriseOne aplikasi dan fungsinya. Untuk mengetahui daftar lengkap faktor kunci yang perlu dipertimbangkan selama proses migrasi, lihat Strategi migrasi database Oracle di AWS Prescriptive Guidance.

Ada beberapa cara untuk memigrasikan EnterpriseOne database, termasuk:

- Menggunakan Oracle Universal Batch Engine (UBE) R98403 untuk pembuatan skema dan tabel, dan menggunakan AWS Database Migration Service (AWS DMS) untuk migrasi
- Menggunakan alat asli DB untuk pembuatan skema dan tabel dan menggunakan AWS DMS untuk migrasi
- Menggunakan alat asli DB untuk migrasi data yang ada (beban penuh) dan menggunakan AWS DMS untuk tugas pengambilan data perubahan (CDC)

Pola ini mencakup opsi ketiga. Ini menjelaskan cara memigrasikan EnterpriseOne database lokal Anda ke Amazon RDS for Oracle dengan menggunakan Oracle Data Pump dengan AWS DMS dan fitur CDC-nya.

Oracle JD Edwards EnterpriseOne adalah solusi perencanaan sumber daya perusahaan (ERP) untuk organisasi yang memproduksi, membangun, mendistribusikan, melayani, atau mengelola produk atau aset fisik. JD Edwards EnterpriseOne mendukung berbagai perangkat keras, sistem operasi, dan platform basis data.

Saat Anda memigrasikan aplikasi ERP penting seperti JD Edwards EnterpriseOne, meminimalkan waktu henti adalah kuncinya. AWS DMS meminimalkan waktu henti dengan mendukung pemuatan penuh dan replikasi berkelanjutan dari database sumber ke database target. AWS DMS juga menyediakan pemantauan dan pencatatan waktu nyata untuk migrasi, yang dapat membantu Anda mengidentifikasi dan menyelesaikan masalah apa pun yang dapat menyebabkan waktu henti.

Saat Anda mereplikasi perubahan dengan AWS DMS, Anda harus menentukan waktu atau nomor perubahan sistem (SCN) sebagai titik awal untuk membaca perubahan dari log database. Sangat penting untuk menjaga agar log ini dapat diakses di server untuk jangka waktu yang ditentukan (kami sarankan 15 hari) untuk memastikan bahwa AWS DMS memiliki akses ke perubahan ini.

Prasyarat dan batasan

Prasyarat

- Basis data Amazon RDS for Oracle yang disediakan di lingkungan AWS Cloud Anda sebagai basis data target. Untuk petunjuk, lihat dokumentasi Amazon RDS.
- EnterpriseOne Database yang berjalan di tempat atau di instans Amazon Elastic Compute Cloud (Amazon EC2) di AWS.

> **Note**
>
> Pola ini dirancang untuk bermigrasi dari tempat ke AWS, tetapi diuji dengan menggunakan EnterpriseOne database pada sebuah EC2 instance. Jika berencana untuk bermigrasi dari lingkungan lokal, Anda harus mengonfigurasi konektivitas jaringan yang sesuai.

- Rincian skema. Identifikasi skema database Oracle mana (misalnya, DV92 0) yang Anda rencanakan untuk migrasi. EnterpriseOne Sebelum Anda memulai proses migrasi, kumpulkan rincian berikut tentang skema:
  - Ukuran skema
  - Jumlah objek per jenis objek
  - Jumlah objek yang tidak valid

Batasan

- Anda harus membuat skema apa pun yang Anda inginkan pada target Amazon RDS for Oracle Database – AWS DMS tidak membuat ini untuk Anda. (Bagian Epics menjelaskan cara menggunakan Pompa Data untuk mengekspor dan mengimpor skema.) Nama skema harus sudah ada untuk database Oracle target. Tabel dari skema sumber diimpor ke pengguna atau skema, dan AWS DMS menggunakan administrator atau akun sistem untuk terhubung ke instans target. Untuk memigrasikan beberapa skema, Anda dapat membuat beberapa tugas replikasi. Anda juga dapat memigrasikan data ke skema yang berbeda pada instance target. Untuk melakukan ini, gunakan aturan transformasi skema pada pemetaan tabel AWS DMS.

- Pola ini telah diuji dengan dataset demo. Kami menyarankan Anda memvalidasi kompatibilitas untuk dataset dan kustomisasi Anda.
- Pola ini menggunakan EnterpriseOne database yang berjalan di Microsoft Windows. Namun, Anda dapat menggunakan proses yang sama dengan sistem operasi lain yang didukung oleh AWS DMS.

Arsitektur

Diagram berikut menunjukkan sistem yang berjalan EnterpriseOne pada database Oracle sebagai database sumber, dan database Amazon RDS for Oracle sebagai database target. Data diekspor dari database Oracle sumber dan diimpor ke database Amazon RDS for Oracle target dengan menggunakan Oracle Data Pump, dan direplikasi untuk pembaruan CDC dengan menggunakan AWS DMS.

1. Oracle Data Pump mengekstrak data dari database sumber, dan data dikirim ke target database Amazon RDS for Oracle.
2. Data CDC dikirim dari database sumber ke titik akhir sumber di AWS DMS.
3. Dari titik akhir sumber, data dikirim ke instans replikasi AWS DMS, tempat tugas replikasi dilakukan.
4. Setelah tugas replikasi selesai, data dikirim ke titik akhir target di AWS DMS.
5. Dari titik akhir target, data dikirim ke instans database Amazon RDS for Oracle.

Alat

Layanan AWS

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di AWS Cloud.

Layanan lainnya

- Oracle Data Pump membantu Anda memindahkan data dan metadata dari satu database ke database lainnya dengan kecepatan tinggi.

Praktik terbaik

Migrating LOBs

Jika database sumber Anda berisi objek biner besar (LOBs) yang perlu dimigrasikan ke database target, AWS DMS menyediakan opsi berikut:

- Mode LOB penuh — AWS DMS memigrasikan semua LOBs dari sumber ke database target terlepas dari ukurannya. Meskipun migrasi lebih lambat dari mode lainnya, keuntungannya adalah data tidak terpotong. Untuk kinerja yang lebih baik, Anda dapat membuat tugas terpisah pada instance replikasi baru untuk memigrasikan tabel LOBs yang memiliki ukuran lebih besar dari beberapa megabyte.
- Mode LOB terbatas - Anda menentukan ukuran maksimum data kolom LOB, yang memungkinkan AWS DMS mengalokasikan sumber daya terlebih dahulu dan menerapkan secara massal. LOBs Jika ukuran kolom LOB melebihi ukuran yang ditentukan dalam tugas, AWS DMS memotong data dan mengirimkan peringatan ke file log AWS DMS. Anda dapat meningkatkan kinerja dengan menggunakan mode LOB terbatas jika ukuran data LOB Anda berada dalam ukuran LOB terbatas.
- Mode LOB sebaris - Anda dapat bermigrasi LOBs tanpa memotong data atau memperlambat kinerja tugas Anda dengan mereplikasi baik kecil maupun besar. LOBs Pertama, tentukan nilai untuk `InlineLobMaxSize` parameter, yang hanya tersedia ketika mode LOB penuh diatur ke`true`. Tugas AWS DMS mentransfer LOBs inline kecil, yang lebih efisien. Kemudian, AWS DMS memigrasikan yang besar LOBs dengan melakukan pencarian dari tabel sumber. Namun, mode LOB inline hanya berfungsi selama fase beban penuh.

Menghasilkan nilai urutan

Selama proses AWS DMS CDC, nomor urutan inkremental tidak direplikasi dari database sumber. Untuk menghindari perbedaan dalam nilai urutan, Anda harus menghasilkan nilai urutan terbaru dari sumber untuk semua urutan, dan menerapkannya ke database Amazon RDS for Oracle target.

AWS Secrets Manager

Untuk membantu mengelola kredensil Anda, kami sarankan Anda mengikuti petunjuk di posting blog Kelola kredenal titik akhir AWS DMS Anda dengan AWS Secrets Manager.

Performa

- Instans replikasi - Untuk panduan memilih ukuran instans terbaik, lihat Memilih ukuran terbaik untuk instans replikasi dalam dokumentasi AWS DMS.

- Opsi konektivitas - Untuk menghindari masalah latensi, kami sarankan Anda memilih opsi konektivitas yang tepat. AWS Direct Connect menyediakan jalur terpendek ke sumber daya AWS, karena ini adalah koneksi khusus antara pusat data perusahaan Anda dan AWS. Saat transit, lalu lintas jaringan Anda tetap berada di jaringan global AWS dan tidak pernah melalui internet. Ini mengurangi kemungkinan mencapai kemacetan atau peningkatan latensi yang tidak terduga jika dibandingkan dengan menggunakan VPN atau internet publik.
- Bandwidth jaringan - Untuk mengoptimalkan kinerja, verifikasi bahwa throughput jaringan Anda cepat. Jika Anda menggunakan terowongan VPN antara database sumber lokal dan AWS DMS, pastikan bandwidth cukup untuk beban kerja Anda.
- Paralelisme tugas - Anda dapat mempercepat replikasi data dengan memuat beberapa tabel secara paralel selama beban penuh. Pola ini menggunakan titik akhir RDBMS, jadi opsi ini hanya berlaku untuk proses beban penuh. Paralelisme tugas dikendalikan oleh `MaxFullLoadSubTasks` parameter, yang menentukan berapa banyak sub-tugas beban penuh yang dijalankan secara paralel. Secara default, parameter ini diatur ke 8, yang berarti bahwa delapan tabel (jika dipilih dalam pemetaan tabel) dimuat bersama selama mode penuh. Anda dapat menyesuaikan parameter ini di bagian pengaturan tugas beban penuh dari skrip JSON untuk tugas tersebut.
- Paralelisme tabel - AWS DMS juga memungkinkan Anda memuat satu tabel besar dengan menggunakan beberapa thread paralel. Ini sangat berguna untuk tabel sumber Oracle yang memiliki miliaran catatan serta beberapa partisi dan subpartisi. Jika tabel sumber tidak dipartisi, Anda dapat menggunakan batas kolom untuk beban paralel.
- Membagi beban - Saat Anda membagi beban di beberapa tugas atau instans AWS DMS, ingat batas transaksi saat Anda menangkap perubahan.

Epik

Gunakan Oracle Data Pump untuk mengekspor skema EnterpriseOne

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan SCN.	Ketika database sumber aktif dan digunakan oleh Enterpris eOne aplikasi, memulai ekspor data dengan Oracle Data Pump. Anda harus terlebih dahulu menghasilkan nomor perubahan sistem (SCN)	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dari database sumber untuk konsistensi data selama ekspor dengan Oracle Data Pump dan sebagai titik awal untuk CDC di AWS DMS.
	
	Untuk menghasilkan SCN saat ini dari database sumber Anda, gunakan pernyataan SQL berikut:
	
	`SQL> select current_scn from v$database;`
	
	`CURRENT_SCN`
	`-----------`
	`30009727`
	
	Simpan SCN yang dihasilkan. Anda akan menggunakan SCN saat mengekspor data dan membuat tugas replikasi AWS DMS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file parameter.	Untuk membuat file parameter untuk mengekspor skema, Anda dapat menggunakan kode berikut.
	`directory=DMS_DATA_PUMP_DIR logfile=export_dms.log dumpfile=export_dms_data.dmp schemas=<schema name> flashback_scn=<SCN from previous command>`
	Note
	Anda juga dapat menentukan sendiri `DATA_PUMP_DIR` dengan menggunakan perintah berikut, berdasarkan kebutuhan Anda.
	`SQL> CREATE OR REPLACE DIRECTORY DMS_DATA_PUMP_DIR AS '<Directory for dump>'; Directory created.`
	`SQL> GRANT READ, WRITE ON DIRECTORY DMS_DATA_PUMP_DIR TO SYSTEM; Grant succeeded.`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ekspor skema.	Untuk melakukan ekspor, gunakan `expdp` utilitas sebagai berikut:
	
	`C:\Users\Administrator>expdp system/********@<DB Name> PARFILE='<Path to PAR file create above>'`
	
	`Export: Release 19.0.0.0.0 - Production on *** *** ** **:**:** **** Version 19.3.0.0.0`
	
	`Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.`
	
	`Connected to: Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production Starting "SYSTEM"."SYS_EXPORT_SCHEMA_02": system/******@<DB Name>PARFILE='E:\exp_dms_datapump.par' Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS Processing object type SCHEMA_EXPORT/TABL`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	E/STATISTICS/TABLE _STATISTICS Processing object type SCHEMA_EXPORT/STAT ISTICS/MARKER Processing object type SCHEMA_EXPORT/USER Processing object type SCHEMA_EXPORT/ROLE _GRANT Processing object type SCHEMA_EXPORT/DEFA ULT_ROLE Processing object type SCHEMA_EXPORT/TABL ESPACE_QUOTA Processing object type SCHEMA_EXPORT/PRE_ SCHEMA/PROCACT_SCHEMA Processing object type SCHEMA_EXPORT/TABLE/ TABLE Processing object type SCHEMA_EXPORT/TABL E/GRANT/OWNER_GRANT/ OBJECT_GRANT Processing object type SCHEMA_EXPORT/TABLE/ INDEX/INDEX Processing object type SCHEMA_EXPORT/TABLE/ CONSTRAINT/CONSTRAINT . . exported "<Schema Name>"."<Table Name>" 228.9 MB 496397 rows Master table "SYSTEM". "SYS_EXPORT_SCHEMA _02" successfully loaded/unloaded	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	``` ********** ********** ********** ********** **** Dump file set for SYSTEM.SYS_EXPORT_ SCHEMA_02 is: E:\DMSDUMP\EXPORT_ DMS_DATA.DMP Job "SYSTEM"."SYS_EXPO RT_SCHEMA_02" successfully completed at *** *** ** **:**:** **** elapsed 0 00:01:57 ```	

Gunakan Oracle Data Pump untuk mengimpor skema EnterpriseOne

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Transfer file dump ke instance target.	Untuk mentransfer file Anda dengan menggunakan `DBMS_FILE_TRANSFER` utilitas, Anda perlu membuat link database dari database sumber ke Amazon RDS for Oracle instance. Setelah tautan dibuat, Anda dapat menggunakan utilitas untuk mentransfer file Pompa Data langsung ke instance Amazon RDS. Atau, Anda dapat mentransfer file Pompa Data ke Amazon Simple Storage Service (Amazon S3) dan kemudian	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	mengimpornya ke instans Amazon RDS for Oracle. Untuk informasi selengkapnya tentang opsi ini, lihat bagian Informasi tambahan.
	
	Untuk membuat tautan database ORARDSDB yang terhubung ke pengguna master Amazon RDS di instans DB target, jalankan perintah berikut pada database sumber:
	
	`sqlplus / as sysdba`
	
	`SQL*Plus: Release 19.0.0.0.0 on *** *** ** **:**:** **** Version 19.3.0.0.0`
	
	`Copyright (c) 1982, 2019, Oracle. All rights reserved.`
	
	`Connected to: Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 Version 19.3.0.0.0`
	
	`SQL> create database link orardsdb connect to admin identifie d by "******" using '(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = orcl.**** **.us-east-1.rds.a`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`mazonaws.com)(PORT = 1521))(CONNECT_DATA = (SERVER = DEDICATED ) (SERVICE_NAME = orcl)))';`
	
	`Database link created.`
	
	`SQL>`	

Uji tautan basis data.	Uji tautan database untuk memastikan bahwa Anda dapat terhubung ke database target Amazon RDS for Oracle dengan menggunakan. `sqlplus`
	
	`SQL> select name from v$database@orardsdb;`
	
	`NAME`
	`---------`
	`ORCL`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Transfer file dump ke database target.	Untuk menyalin file dump ke database Amazon RDS for Oracle, Anda dapat menggunakan direktori `DATA_PUMP_DIR` default atau Anda dapat membuat direktori Anda sendiri dengan menggunakan kode berikut, yang harus dijalankan pada instance Amazon RDS target:
	`exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'DMS_TARGET_PUMP_DIR');`
	`PL/SQL procedure successfully completed.`
	Script berikut menyalin file dump bernama `EXPORT_DMS_DATA.DMP` dari instance sumber ke target Amazon RDS for Oracle database dengan menggunakan link database bernama. `orardsdb` Anda harus menjalankan skrip pada instance database sumber.
	`BEGIN DBMS_FILE_TRANSFER.PUT_FILE(`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`source_directory_object => 'DMS_DATA_PUMP_DIR', source_file_name => 'EXPORT_DMS_DATA.DMP', destination_directory_object => 'DMS_TARGET_PUMP_DIR', destination_file_name => 'EXPORT_DMS_DATA.DMP', destination_database => 'orardsdb'); END;`
	
	`PL/SQL procedure successfully completed.`	

Buat daftar file dump dalam database target.	Setelah PL/SQL prosedur selesai, Anda dapat mencantumkan file dump data di database Amazon RDS for Oracle dengan menggunakan kode berikut: `select * from table (rdsadmin.rds_file_util.listdir(p_directory => 'DMS_TARGET_PUMP_DIR'));`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat pengguna khusus JDE di Instance target.	Buat profil dan peran JD Edwards dengan menggunakan perintah ini dalam instance target:
	`SQL> CREATE PROFILE "JDEPROFILE" LIMIT IDLE_TIME 15;`
	`Profile created.`
	
	`SQL> CREATE ROLE "JDE_ROLE";`
	`Role created.`
	
	`SQL> CREATE ROLE "JDEADMIN";`
	`CREATE ROLE "JDEUSER";`
	`Role created.`
	`Role created.`
	
	Berikan izin yang diperlukan untuk peran:
	`SQL> GRANT CREATE ANY SEQUENCE TO JDE_ROLE;`
	`GRANT DROP ANY SEQUENCE TO JDE_ROLE;`
	`GRANT CREATE ANY TRIGGER TO JDE_ROLE;`
	`GRANT DROP ANY TRIGGER TO JDE_ROLE;`
		DBA, JDE CNC

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat tablespaces dalam instance target.	Buat ruang tabel yang diperlukan dalam instance target dengan menggunakan perintah berikut untuk skema yang terlibat dalam migrasi ini:
	`SQL> CREATE TABLESPACE <Tablespace Name for Tables>;`
	`Tablespace created.`
	
	`SQL> CREATE TABLESPACE <Tablespace Name for Indexes>;`
	`Tablespace created.`
		DBA, JDE CNC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai impor pada database target.	Sebelum memulai proses impor, atur peran, skema, dan ruang tabel pada database Amazon RDS for Oracle target menggunakan file dump data. Untuk melakukan impor, akses database target dengan akun pengguna utama Amazon RDS, dan gunakan nama string koneksi dalam `tnsnames.ora` file, yang mencakup Amazon RDS for Oracle Database. `tns-entry` Jika perlu, Anda dapat menyertakan opsi remap untuk mengimpor file dump data ke ruang tabel yang berbeda atau di bawah nama skema yang berbeda. Untuk memulai impor, gunakan kode berikut: `impdp admin@orardsdb directory=DMS_TARGET_PUMP_DIR logfile=import.log dumpfile=EXPORT_DMS_DATA.DMP` Untuk memastikan impor berhasil, periksa file log impor untuk kesalahan apa pun, dan tinjau detail seperti jumlah objek, jumlah baris, dan objek yang tidak valid. Jika	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ada objek yang tidak valid, kompilasi ulang. Selain itu, bandingkan objek database sumber dan target untuk mengonfirmasi bahwa mereka cocok.	

Menyediakan instans replikasi AWS DMS dengan titik akhir sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh templat .	Unduh template AWS CloudFormation DMS_Instance.yaml untuk menyediakan instans replikasi AWS DMS serta titik akhir sumber dan targetnya.	Administrator awan, DBA
Mulai pembuatan tumpukan.	1. Masuk ke AWS Management Console, dan buka CloudFormation konsol AWS di https://console.aws.amazon.com/cloudformation. 2. Pilih Buat tumpukan. 3. Untuk Tentukan templat, pilih Unggah file templat. 4. Pilih Pilih file. 5. Pilih `DMS_instance.yaml` file. 6. Pilih Berikutnya.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan parameternya.	1. Untuk nama Stack, masukkan nama tumpukan Anda. 2. Untuk Parameter Instans AWS DMS, masukkan parameter berikut: • DMSInstanceJenis — Pilih instans yang diperlukan untuk instans replikasi AWS DMS, berdasarkan kebutuhan bisnis Anda. • DMSStorageUkuran — Masukkan ukuran penyimpanan untuk instans AWS DMS, berdasarkan ukuran migrasi Anda. 3. Untuk Source Oracle Database Configuration, masukkan parameter berikut: • SourceOracleEndpointID — Nama server database Oracle sumber • SourceOracleDatabaseName— Nama layanan database sumber atau ID sesi (SID) sebagaimana berlaku • SourceOracleUserName— Nama pengguna database	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber (defaultnya adalah`system`) • SourceOracleDBPass word— Kata sandi nama pengguna basis data sumber • SourceOracleDBPort— Port basis data sumber 4. Untuk Target RDS untuk Oracle Database Configura tion, masukkan parameter berikut: • Target RDSOracle EndpointID - Titik akhir basis data RDS target • Target RDSOracle DatabaseName - Nama basis data RDS target • Target RSOracle UserName - Nama pengguna RDS target • Target RDSOracle DBPassword - Kata sandi RDS target • TargetOracleDBPort— Target port database RDS 5. Untuk Konfigurasi VPC, Subnet dan Grup Keamanan, masukkan parameter berikut: • VPCID — VPC untuk contoh replikasi	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• VPCSecurityGroupId— Grup keamanan VPC untuk contoh replikasi • DMSSubnet1— Subnet untuk Availability Zone 1 • DMSSubnet2— Subnet untuk Availability Zone 2 6. Pilih Berikutnya.	
Buat tumpukan.	1. Pada halaman Configure stack options, untuk Tags, masukkan nilai opsional apa pun. 2. Pilih Berikutnya. 3. Pada halaman Ulasan, verifikasi detailnya, lalu pilih Kirim. Penyediaan harus selesai dalam waktu sekitar 5-10 menit. Ini selesai ketika halaman AWS CloudForm ation Stacks menampilkan CREATE_COMPLETE.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan titik akhir.	1. Buka konsol AWS DMS di https://console.aws.amazon.com/dms/v2/. 2. Untuk manajemen Sumber Daya, pilih Instance replikasi, lalu tinjau instance replikasi. 3. Untuk manajemen Sumber Daya, pilih Endpoint, lalu tinjau titik akhir.	Administrator awan, DBA
Uji konektivitas.	Setelah titik akhir sumber dan target menunjukkan status sebagai Aktif, uji konektivitas. Pilih Jalankan pengujian untuk setiap titik akhir (sumber dan target) untuk memastikan status ditampilkan sebagai berhasil.	Administrator awan, DBA

Buat tugas replikasi AWS DMS untuk replikasi langsung

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas replikasi.	Buat tugas replikasi AWS DMS dengan menggunakan langkah-langkah berikut: 1. Buka konsol AWS DMS di https://console.aws.amazon.com/dms/v2/. 2. Di panel navigasi, di bawah Migrasi Data, pilih Tugas migrasi database.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di kotak Konfigurasi tugas, untuk pengidentifikasi tugas, masukkan pengident ifikasi tugas Anda. 4. Untuk contoh Replikasi, pilih contoh replikasi DMS yang Anda buat. 5. Untuk titik akhir basis data Sumber, pilih titik akhir sumber Anda. 6. Untuk titik akhir database Target, pilih database Amazon RDS for Oracle target Anda. 7. Untuk jenis Migrasi, pilih Replikasi perubahan data saja. Jika Anda menerima pesan bahwa pencatatan tambahan perlu diaktifka n, ikuti petunjuk di bagian Pemecahan Masalah. 8. Dalam kotak Pengaturan tugas, pilih Tentukan nomor urutan log. 9. Untuk nomor perubahan Sistem, masukkan SCN database Oracle yang Anda hasilkan dari database Oracle sumber. 10. Pilih Aktifkan validasi. 11. Pilih Aktifkan CloudWatch Log.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Dengan mengaktifkan fitur ini, Anda dapat memvalidasi data dan CloudWatch log Amazon untuk meninjau log instans replikasi AWS DMS. 12. Di bawah aturan Seleksi, lengkapi yang berikut ini: • Untuk Skema, pilih Masukkan skema. • Untuk nama Skema, masukkan nama Skema JDE (misalnya: DV92 0). • Untuk nama Tabel, masukkan%. • Untuk Tindakan, pilih Sertakan. 13. Pilih Buat tugas. Setelah Anda membuat tugas, AWS DMS memigrasikan perubahan yang sedang berlangsung ke instans database Amazon RDS for Oracle dari SCN yang Anda berikan dalam mode mulai CDC. Anda juga dapat memverifikasi migrasi dengan meninjau CloudWatch log.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ulangi tugas replikasi.	Ulangi langkah sebelumnya untuk membuat tugas replikasi untuk skema JD Edwards lain yang merupakan bagian dari migrasi.	Administrator awan, DBA, administrator JDE CNC

Validasi skema database pada target Amazon RDS for Oracle database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi transfer data.	Setelah tugas AWS DMS dimulai, Anda dapat memeriksa tab Statistik tabel di halaman Tugas untuk melihat perubahan yang dibuat pada data. Anda dapat memantau status replikasi yang sedang berlangsung di konsol pada halaman tugas migrasi database. Untuk informasi selengkapnya, lihat validasi data AWS DMS.	Administrator awan, DBA

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan replikasi.	Hentikan prosedur replikasi dan hentikan layanan aplikasi sumber.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan aplikasi JD Edwards.	Luncurkan presentasi JD Edwards target dan aplikasi tingkat logika di AWS, dan arahkan ke database Amazon RDS for Oracle. Saat Anda mengakses aplikasi, Anda harus memperhatikan bahwa semua koneksi sekarang dibuat dengan database Amazon RDS for Oracle.	DBA, administrator JDE CNC
Matikan database sumber.	Setelah Anda mengonfirmasi bahwa tidak ada lagi koneksi, Anda dapat mematikan basis data sumber.	DBA

Pemecahan masalah

Isu	Solusi

Anda menerima pesan peringatan untuk mengaktifkan pencatatan tambahan di database sumber untuk replikasi yang sedang berlangsung	Masukkan perintah ini untuk mengaktifkan logging tambahan:
	
	`SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;`
	`SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;`
	`SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;`
	`SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS;`
	`SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;`

Isu	Solusi
	`SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;`
AWS DMS memiliki logging tambahan yang dimatikan.	Pencatatan tambahan dimatikan secara default di AWS DMS. Untuk menyalakannya untuk titik akhir Oracle sumber: 1. Masuk ke AWS Management Console dan buka konsol AWS DMS di https://console.aws.amazon.com/dms/v2/. 2. Pilih Titik akhir. 3. Pilih titik akhir sumber Oracle yang ingin Anda tambahkan supplemental logging. 4. Pilih Ubah. 5. Pilih Lanjutan, dan kemudian tambahkan kode berikut ke kotak teks Atribut koneksi tambahan: `addSupplementalLogging=Y` 6. Pilih Ubah.

Pencatatan tambahan tidak diaktifkan di tingkat CDB.	1. Masukkan perintah ini:
	`SQL> alter session set container = CDB$ROOT;`
	`Session altered.`
	2. Ulangi langkah-langkah untuk mengaktifkan logging tambahan.

Isu	Solusi
Anda menerima pesan galat: “Test Endpoint failed: Application-Status: 1020912, Applicati on-Message: LogMiner is not supported in Oracle PDB environment Endpoint initialization failed.”	Jika Anda menemukan pesan kesalahan ini, Anda dapat menggunakan Binary Reader sebagai gantinya LogMiner. Di bawah Pengaturan titik akhir, tambahkan baris ini ke atribut koneksi tambahan untuk basis data sumber Anda: `useLogMinerReader=N;useBfile=Y;`

Sumber daya terkait

- Memulai AWS Database Migration Service
- Praktik Terbaik untuk AWS Database Migration Service
- Migrasi Database Oracle ke AWS Cloud
- Referensi jenis sumber daya AWS Database Migration Service untuk AWS CloudFormation
- Kelola kredensil titik akhir AWS DMS Anda dengan AWS Secrets Manager
- Memecahkan masalah tugas migrasi di AWS Database Migration Service
- Praktik terbaik untuk AWS Database Migration Service

Informasi tambahan

Transfer file menggunakan Amazon S3

Untuk mentransfer file ke Amazon S3, Anda dapat menggunakan AWS CLI atau konsol Amazon S3. Setelah mentransfer file ke Amazon S3, Anda dapat menggunakan instans Amazon RDS for Oracle untuk mengimpor file Pompa Data dari Amazon S3.

Jika Anda memilih untuk mentransfer file dump menggunakan integrasi Amazon S3 sebagai metode alternatif, lakukan langkah-langkah berikut:

1. Buat ember S3.
2. Ekspor data dari database sumber menggunakan Oracle Data Pump.
3. Unggah file Pompa Data ke bucket S3.

4. Unduh file Pompa Data dari bucket S3 ke database Amazon RDS for Oracle target.
5. Lakukan impor menggunakan file Pompa Data.

> **Note**
>
> Untuk mentransfer file data besar antara instans S3 dan RDS, sebaiknya gunakan fitur Amazon S3 Transfer Acceleration.

Migrasikan PeopleSoft database Oracle ke AWS dengan menggunakan AWS DMS

sampath kathirvel, Amazon Web Services

Ringkasan

Oracle PeopleSoft adalah solusi perencanaan sumber daya perusahaan (ERP) untuk proses di seluruh perusahaan. PeopleSoft memiliki arsitektur tiga tingkat: klien, aplikasi, dan database. PeopleSoft dapat dijalankan di Amazon Relational Database Service (Amazon RDS).

Jika Anda memigrasikan database Oracle ke Amazon RDS, Amazon Web Services (AWS) dapat menangani tugas pencadangan dan ketersediaan tinggi, sehingga Anda bebas berkonsentrasi pada pemeliharaan PeopleSoft aplikasi dan fungsinya. Untuk mengetahui daftar lengkap faktor kunci yang perlu dipertimbangkan selama proses migrasi, lihat Strategi migrasi database Oracle di AWS Prescriptive Guidance.

Pola ini memberikan solusi untuk memigrasikan database Oracle lokal Anda ke Amazon RDS for Oracle menggunakan Oracle Data Pump dengan AWS Database Migration Service (AWS DMS) dan fitur change data capture (CDC).

Saat memigrasikan aplikasi ERP penting seperti Oracle PeopleSoft, meminimalkan waktu henti adalah kuncinya. AWS DMS meminimalkan waktu henti dengan mendukung beban penuh dan replikasi berkelanjutan. dari database sumber ke database target. AWS DMS juga menyediakan pemantauan dan pencatatan migrasi secara real-time, yang dapat membantu Anda mengidentifikasi dan menyelesaikan masalah apa pun yang dapat menyebabkan waktu henti.

Saat mereplikasi perubahan dengan AWS DMS, Anda harus menentukan waktu atau nomor perubahan sistem (SCN) sebagai titik awal AWS DMS untuk membaca perubahan dari log database. Sangat penting untuk menjaga agar log ini dapat diakses di server untuk jangka waktu yang ditentukan untuk memastikan bahwa AWS DMS memiliki akses ke perubahan ini.

Prasyarat dan batasan

Prasyarat

- Basis data Amazon RDS for Oracle yang disediakan di lingkungan AWS Cloud Anda sebagai basis data target.
- PeopleSoft Database Oracle yang berjalan di tempat atau di Amazon Elastic Compute Cloud EC2 (Amazon) di AWS Cloud.

> **Note**
>
> Pola ini dirancang untuk bermigrasi dari lokasi ke AWS, tetapi diuji dengan menggunakan Oracle Database pada instans Amazon EC2 . Untuk bermigrasi dari tempat, Anda perlu mengonfigurasi konektivitas jaringan yang sesuai.

- Rincian skema. Saat memigrasikan PeopleSoft aplikasi Oracle ke Amazon RDS for Oracle, perlu untuk mengidentifikasi skema database Oracle mana (misalnya,) untuk bermigrasi. `SYSADM` Sebelum memulai proses migrasi, kumpulkan rincian berikut tentang skema:
  - Size
  - Jumlah objek per jenis objek
  - Jumlah objek yang tidak valid.

  Informasi ini akan membantu proses migrasi.

Batasan

- Skenario ini telah diuji hanya dengan database PeopleSoft DEMO. Itu belum diuji dengan dataset besar.

Arsitektur

Diagram berikut menunjukkan instance yang menjalankan database Oracle sebagai database sumber dan database Amazon RDS for Oracle sebagai database target. Data diekspor dan diimpor dari database Oracle sumber ke database Amazon RDS for Oracle target menggunakan Oracle Data Pump dan direplikasi untuk perubahan CDC menggunakan AWS DMS.

1. Langkah awal melibatkan ekstraksi data dari database sumber dengan menggunakan Oracle Data Pump, diikuti dengan mengirimkannya ke target database Amazon RDS for Oracle.
2. Data dikirim dari database sumber ke titik akhir sumber di AWS DMS.
3. Dari titik akhir sumber, data dikirim ke instans replikasi AWS DMS, tempat tugas replikasi dilakukan.
4. Setelah tugas replikasi selesai, data dikirim ke titik akhir target di AWS DMS.
5. Dari titik akhir target, data dikirim ke instans database Amazon RDS for Oracle.

Alat

Layanan AWS

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di AWS Cloud.

Layanan lainnya

- Oracle Data Pump membantu Anda memindahkan data dan metadata dari satu database ke database lainnya dengan kecepatan tinggi.

Praktik terbaik

Migrating LOBs

Jika database sumber Anda berisi objek biner besar (LOBs) yang perlu dimigrasikan ke database target, AWS DMS menyediakan opsi berikut:

- Mode LOB penuh — AWS DMS memigrasikan semua LOBs dari sumber ke database target terlepas dari ukurannya. Meskipun migrasi lebih lambat, keuntungannya adalah data tidak terpotong. Untuk kinerja yang lebih baik, Anda dapat membuat tugas terpisah pada instance replikasi baru untuk memigrasikan tabel yang memiliki LOBs ukuran lebih besar dari beberapa megabyte.
- Mode LOB terbatas - Anda menentukan ukuran maksimum data kolom LOB, yang memungkinkan AWS DMS mengalokasikan sumber daya terlebih dahulu dan menerapkan secara massal. LOBs Jika ukuran kolom LOB melebihi ukuran yang ditentukan dalam tugas, AWS DMS memotong data dan mengirimkan peringatan ke file log AWS DMS. Anda dapat meningkatkan kinerja dengan menggunakan mode LOB Terbatas jika ukuran data LOB Anda berada dalam ukuran LOB Terbatas.
- Mode LOB sebaris - Anda dapat bermigrasi LOBs tanpa memotong data atau memperlambat kinerja tugas Anda dengan mereplikasi baik kecil maupun besar. LOBs Pertama, tentukan nilai untuk InlineLobMaxSize parameter, yang hanya tersedia ketika mode LOB Penuh diatur ke true. Tugas AWS DMS mentransfer LOBs inline kecil, yang lebih efisien. Kemudian, AWS DMS memigrasikan yang besar LOBs dengan melakukan pencarian dari tabel sumber. Namun, mode LOB Inline hanya berfungsi selama fase beban penuh.

Menghasilkan nilai urutan

Perlu diingat bahwa selama proses pengambilan data perubahan dengan AWS DMS, nomor urutan tambahan tidak direplikasi dari database sumber. Untuk menghindari perbedaan dalam nilai urutan, Anda harus menghasilkan nilai urutan terbaru dari sumber untuk semua urutan, dan menerapkannya ke database Amazon RDS for Oracle target.

Manajemen kredensi

Untuk membantu mengamankan sumber daya AWS Anda, sebaiknya ikuti praktik terbaik AWS Identity and Access Management (IAM).

Epik

Menyediakan instans replikasi AWS DMS dengan titik akhir sumber dan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh templat .	Unduh CloudFormation template AWS DMS_Instance.yaml untuk menyediakan instans replikasi AWS DMS serta titik akhir sumber dan targetnya.	Administrator awan, DBA
Mulai pembuatan tumpukan.	1. Di AWS Management Console, pilih CloudFormation. 2. Pilih Buat tumpukan. 3. Untuk Tentukan templat, pilih Unggah file templat. 4. Pilih Pilih file. 5. Pilih `DMS_instance.yaml` file. 6. Pilih Berikutnya.	Administrator awan, DBA
Tentukan parameternya.	1. Untuk nama Stack, masukkan nama tumpukan Anda.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di bawah Parameter Instans AWS DMS, masukkan parameter berikut: • DMSInstanceJenis — Pilih instans yang diperlukan untuk instans replikasi AWS DMS, berdasarkan kebutuhan bisnis Anda. • DMSStorageUkuran — Masukkan ukuran penyimpanan untuk instans AWS DMS, berdasarkan ukuran migrasi Anda. 3. Di bawah Source Oracle Database Configuration, masukkan parameter berikut: • SourceOracleEndpointID — Nama server database Oracle sumber • SourceOracleDatabaseName— Nama layanan database sumber atau ID sesi (SID) sebagaimana berlaku • SourceOracleUserName— Nama pengguna database sumber (defaultnya adalah sistem)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• SourceOracleDBPassword— Kata sandi nama pengguna basis data sumber • SourceOracleDBPort— Port basis data sumber 4. Di bawah Target RDS untuk Oracle Database Configuration, masukkan parameter berikut: • Target RDSOracle EndpointID - Titik akhir basis data RDS target • Target RDSOracle DatabaseName - Nama basis data RDS target • Target RSOracle UserName - Nama pengguna RDS target • Target RDSOracle DBPassword - Kata sandi RDS target • TargetOracleDBPort— Target port database RDS 5. Di bawah VPC, Subnet dan Konfigurasi Grup Keamanan, masukkan parameter berikut: • VPCID — VPC untuk contoh replikasi	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• VPCSecurityGroupId— Grup keamanan VPC untuk contoh replikasi • DMSSubnet1— Subnet untuk Availability Zone 1 • DMSSubnet2— Subnet untuk Availability Zone 2 6. Pilih Berikutnya.	
Buat tumpukan.	1. Pada halaman Configure stack options, untuk Tags, masukkan nilai opsional apa pun. 2. Pilih Berikutnya. 3. Pada halaman Ulasan, verifikasi detailnya, lalu pilih Kirim. Penyediaan harus selesai dalam waktu sekitar 5-10 menit. Ini selesai ketika halaman AWS CloudForm ation Stacks menampilkan CREATE_COMPLETE.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan titik akhir.	1. Dari AWS Managemen t Console, pilih Layanan Migrasi Database. 2. Di bawah Manajemen sumber daya, pilih Instance replikasi. 3. Di bawah Manajemen sumber daya, pilih Endpoint.	Administrator awan, DBA
Uji konektivitas.	Setelah titik akhir sumber dan target menunjukkan status sebagai Aktif, uji konektivi tas. Pilih Jalankan pengujian untuk setiap titik akhir (sumber dan target) untuk memastika n status ditampilkan sebagai berhasil.	Administrator awan, DBA

Ekspor PeopleSoft skema dari database Oracle lokal dengan menggunakan Oracle Data Pump

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan SCN.	Ketika database sumber aktif dan digunakan oleh aplikasi, memulai ekspor data dengan Oracle Data Pump. Anda harus terlebih dahulu menghasilkan nomor perubahan sistem (SCN) dari database sumber untuk konsistensi data selama ekspor dengan Oracle Data Pump dan sebagai titik awal	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	untuk mengubah pengambilan data di AWS DMS.
	
	Untuk menghasilkan SCN saat ini dari database sumber Anda, masukkan pernyataan SQL berikut.
	
	`SQL> select name from v$database;`
	`SQL> select name from v$database;`
	`NAME`
	`---------`
	`PSFTDMO`
	`SQL> SELECT current_scn FROM v$database;`
	`CURRENT_SCN`
	`-----------`
	`23792008`
	
	Simpan SCN yang dihasilkan untuk digunakan saat Anda mengekspor data dan untuk membuat tugas replikasi AWS DMS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file parameter.	Untuk membuat file parameter untuk mengekspor skema, Anda dapat menggunakan kode berikut.
	`$ cat exp_datapmp.par`
	`userid=system/*******`
	`directory=DATA_P`
	`UMP_DIR`
	`logfile=export_dms_`
	`sample_user.log`
	`dumpfile=export_dms_`
	`sample_data_%U.dmp`
	`schemas=SYSADM`
	`flashback_scn=237920`
	`08`
	Note
	Anda juga dapat menentukan sendiri `DATA_PUMP_DIR` dengan menggunakan perintah berikut, berdasarkan kebutuhan Anda.
	`SQL> CREATE OR REPLACE`
	`DIRECTORY DATA_PUMP`
	`_DIR AS '/opt/oracle/`
	`product/19c/dbhome_1/`
	`dmsdump/';`
	`Directory created.`
	`SQL> GRANT READ, WRITE`
	`ON DIRECTORY DATA_PUMP`
	`_DIR TO system;`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Grant succeeded. SQL> SQL> SELECT owner, directory_name, directory_path FROM dba_directories WHERE directory_name='DA TA_PUMP_DIR'; OWNER DIRECTORY_NAME DIRECTORY_PATH ---------- ---------- ---------- ---------- ---------- ---------- SYS DATA_PUMP_DIR /opt/ oracle/product/19c/dbh ome_1/dmsdump/	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekspor skema.	Untuk melakukan ekspor, gunakan `expdp` utilitas. `$ expdp parfile=e xp_datapmp.par .... Transferring the dump file with DBMS_FILE _TRANSFER to Target: . . exported "SYSADM". "PS_XML_TEMPLT_LNG" 6.320 KB 0 rows . . exported "SYSADM". "PS_XML_TEMPLT_LNK" 6.328 KB 0 rows . . exported "SYSADM". "PS_XML_XLATDEF_LNG" 6.320 KB 0 rows . . exported "SYSADM". "PS_XML_XLATITM_LNG" 7.171 KB 0 rows . . exported "SYSADM". "PS_XPQRYRUNCNTL" 7.601 KB 0 rows . . exported "SYSADM". "PS_XPQRYRUNPARM" 7.210 KB 0 rows . . exported "SYSADM". "PS_YE_AMOUNTS" 9.351 KB 0 rows . . exported "SYSADM". "PS_YE_DATA" 16.58 KB 0 rows . . exported "SYSADM". "PS_YE_EE" 6.75 KB 0 rows . . exported "SYSADM". "PS_YE_W2CP_AMOUNTS" 9.414 KB 0 rows`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	. . exported "SYSADM". "PS_YE_W2CP_DATA" 20.94 KB 0 rows . . exported "SYSADM". "PS_YE_W2C_AMOUNTS" 10.27 KB 0 rows . . exported "SYSADM". "PS_YE_W2C_DATA" 20.95 KB 0 rows . . exported "SYSADM". "PS_ZBD_JOBCODE_TBL" 14.60 KB 0 rows . . exported "SYSADM". "PTGRANTTBL" 5.468 KB 0 rows Master table "SYSTEM". "SYS_EXPORT_SCHEMA _01" successfully loaded/unloaded ** Dump file set for SYSTEM.SYS_EXPORT_ SCHEMA_01 is: /opt/oracle/pr oduct/19c/dbhome_1 /dmsdump/export_dm s_sample_data_01.dmp Job "SYSTEM"."SYS_EXPO RT_SCHEMA_01" successfully completed at Mon Dec 19 20:13:57 2022 elapsed 0 00:38:22	

Impor PeopleSoft skema ke database Amazon RDS for Oracle dengan menggunakan Oracle Data Pump

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Transfer file dump ke instance target.	Untuk mentransfer file Anda menggunakan`DBMS_FILE_TRANSFER`, Anda perlu membuat link database dari database sumber ke Amazon RDS for Oracle instance. Setelah tautan dibuat, Anda dapat menggunakan utilitas untuk mentransfer file Pompa Data langsung ke instance RDS.
	
	Atau, Anda dapat mentransfer file Pompa Data ke Amazon Simple Storage Service (Amazon S3) dan kemudian mengimpornya ke instans Amazon RDS for Oracle. Untuk informasi selengkapnya tentang opsi ini, lihat bagian Informasi tambahan.
	
	Untuk membuat tautan database `ORARDSDB` yang terhubung ke pengguna master Amazon RDS pada instans DB target, jalankan perintah berikut pada database sumber.
	
	`$sqlplus / as sysdba`
	`$ SQL> create database link orardsdb connect`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`to admin identified by "*****" using '(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = testpsft.*******.us-west-2.rds.amazonaws.com)(PORT = 1521))(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orcl)))';` `Database link created.`	

Uji tautan basis data.	Uji tautan database untuk memastikan bahwa Anda dapat terhubung menggunakan sqlplus ke database target Amazon RDS for Oracle.
	`SQL>`
	`SQL> select name from v$database@orardsdb;`
	`NAME`
	`---------`
	`ORCL`
	`SQL>`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Transfer file dump ke database target.	Untuk menyalin file dump ke Amazon RDS for Oracle database, Anda dapat menggunakan direktori `DATA_PUMP_DIR` default atau Anda dapat membuat direktori Anda sendiri menggunakan kode berikut.
	`exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'TARGET_PUMP_DIR') ;`
	Script berikut menyalin file dump bernama `export_dms_sample_data_01.dmp` dari instance sumber ke target Amazon RDS for Oracle database menggunakan link database bernama. `orardsdb`
	`$ sqlplus / as sysdba SQL> BEGIN DBMS_FILE_TRANSFER.PUT_FILE( source_directory_object => 'DATA_PUMP_DIR', source_file_name => 'export_dms_sample_data_01.dmp', destination_directory_object => 'TARGET_PUMP_DIR'',`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`destination_file_name => 'export_dms_sample_data_01.dmp', destination_database => 'orardsdb' ); END; / PL/SQL procedure successfully completed .`	
Buat daftar file dump dalam database target.	Setelah PL/SQL prosedur selesai, Anda dapat mencantumkan file dump data di database Amazon RDS for Oracle dengan menggunakan kode berikut. `SQL> select * from table (rdsadmin.rds_file_util.listdir(p_directory => 'TARGET_PUMP_DIR'));`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai impor pada database target.	Sebelum memulai proses impor, atur peran, skema, dan ruang tabel pada database Amazon RDS for Oracle target menggunakan file dump data. Untuk melakukan impor, akses database target dengan akun pengguna master Amazon RDS, dan gunakan nama string koneksi dalam `tnsnames.ora` file, yang mencakup Amazon RDS for Oracle Database. `tns-entry` Jika perlu, Anda dapat menyertakan opsi remap untuk mengimpor file dump data ke ruang tabel yang berbeda atau di bawah nama skema yang berbeda. Untuk memulai impor, gunakan kode berikut. `impdp admin@orardsdb directory=TARGET_PUMP_DIR logfile=import.log dumpfile=export_dms_sample_data_01.dmp` Untuk memastikan impor berhasil, periksa file log impor untuk kesalahan apa pun, dan tinjau detail seperti jumlah objek, jumlah baris,	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan objek yang tidak valid. Jika ada objek yang tidak valid, kompilasi ulang. Selain itu, bandingkan objek basis data sumber dan target untuk mengonfirmasi bahwa mereka cocok.	

Buat tugas replikasi AWS DMS menggunakan CDC untuk melakukan replikasi langsung

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas replikasi.	Buat tugas replikasi AWS DMS dengan menggunakan langkah-langkah berikut: 1. Di konsol AWS DMS, di bawah Konversi & migrasi, pilih tugas migrasi database. 2. Di bawah Konfigurasi tugas, untuk pengenal Tugas, masukkan pengenal tugas Anda. 3. Untuk contoh Replikasi, pilih contoh replikasi DMS yang Anda buat. 4. Untuk titik akhir basis data Sumber, pilih titik akhir sumber Anda. 5. Untuk titik akhir database Target, pilih database Amazon RDS for Oracle target Anda.	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Untuk jenis Migrasi, pilih Replikasi perubahan data saja. Jika Anda menerima pesan bahwa pencatatan tambahan perlu diaktifkan, ikuti petunjuk di bagian Informasi tambahan. 7. Di bawah Pengaturan tugas, pilih Tentukan nomor urutan log. 8. Untuk nomor perubahan Sistem, masukkan SCN database Oracle yang Anda hasilkan dari database Oracle sumber. 9. Pilih Aktifkan validasi. 10. Pilih Aktifkan CloudWatch Log. Dengan mengaktifkan fitur ini, Anda dapat memvalidasi data dan CloudWatch log Amazon untuk meninjau log instans replikasi AWS DMS. 11. Di bawah aturan Seleksi, lengkapi yang berikut ini: • Untuk Skema, pilih Masukkan skema. • Untuk nama Skema, masukkan SYSADM. • Untuk nama Tabel, masukkan%.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk Tindakan, pilih Sertakan. 12. Di bawah aturan Transformasi, lengkapi yang berikut ini: • Untuk Target, pilih Tabel. • Untuk nama Skema, pilih Masukkan skema. • Untuk nama Skema, masukkan SYSADM. • Untuk Tindakan, pilih Ganti nama menjadi. 13. Pilih Buat tugas. Setelah Anda membuat tugas, tugas tersebut akan memigrasikan CDC ke instance database Amazon RDS for Oracle dari SCN yang Anda berikan dalam mode mulai CDC. Anda juga dapat memverifikasi dengan meninjau CloudWatch log.	

Validasi skema database pada target Amazon RDS for Oracle database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi transfer data.	Setelah tugas AWS DMS dimulai, Anda dapat memeriksa tab Statistik tabel di halaman Tugas untuk	Administrator awan, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	melihat perubahan yang dibuat pada data. Anda dapat memantau status replikasi yang sedang berlangsung di konsol pada halaman tugas migrasi database. Untuk informasi selengkapnya, lihat validasi data AWS DMS.	

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan replikasi.	Hentikan prosedur replikasi dan hentikan layanan aplikasi sumber.	Administrator awan, DBA
Luncurkan tingkat PeopleSoft menengah.	Luncurkan aplikasi tingkat PeopleSoft menengah target di AWS, dan arahkan ke database Amazon RDS for Oracle yang baru saja dimigrasikan. Saat Anda mengakses aplikasi, Anda harus memperhatikan bahwa semua koneksi aplikasi sekarang dibuat dengan database Amazon RDS for Oracle.	DBA, administrator PeopleSoft
Matikan database sumber.	Setelah Anda mengonfirmasi bahwa tidak ada lagi koneksi	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ke database sumber, itu dapat dimatikan.	

Sumber daya terkait

- Memulai AWS Database Migration Service
- Praktik Terbaik untuk AWS Database Migration Service
- Migrasi Database Oracle ke AWS Cloud

Informasi tambahan

Transfer file menggunakan Amazon S3

Untuk mentransfer file ke Amazon S3, Anda dapat menggunakan AWS CLI atau konsol Amazon S3. Setelah mentransfer file ke Amazon S3, Anda dapat menggunakan instans Amazon RDS for Oracle untuk mengimpor file Pompa Data dari Amazon S3.

Jika Anda memilih untuk mentransfer file dump menggunakan integrasi Amazon S3 sebagai metode alternatif, lakukan langkah-langkah berikut:

1. Buat ember S3.
2. Ekspor data dari database sumber menggunakan Oracle Data Pump.
3. Unggah file Pompa Data ke bucket S3.
4. Unduh file Pompa Data dari bucket S3 ke database Amazon RDS for Oracle target.
5. Lakukan impor menggunakan file Pompa Data.

> **Note**
>
> Untuk mentransfer file data besar antara instans S3 dan RDS, disarankan untuk menggunakan fitur Amazon S3 Transfer Acceleration.

Aktifkan pencatatan tambahan

Jika Anda menerima pesan peringatan untuk mengaktifkan pencatatan tambahan di database sumber untuk replikasi yang sedang berlangsung, gunakan langkah-langkah berikut.

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;
```

Memigrasikan database MySQL lokal ke Amazon RDS for MySQL

Lorenzo Mota, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan database MySQL lokal ke Amazon Relational Database Service (Amazon RDS) untuk MySQL. Pola ini membahas penggunaan AWS Database Migration Service (AWS DMS) atau alat MySQL asli seperti mysqldump untuk migrasi database lengkap. Pola ini terutama untuk DBAs dan arsitek solusi. Ini dapat digunakan dalam proyek kecil atau besar sebagai prosedur pengujian (kami merekomendasikan setidaknya satu siklus pengujian) atau sebagai prosedur migrasi akhir.

Prasyarat dan batasan

Prasyarat

- AWS Akun yang aktif
- Database sumber MySQL di pusat data lokal

Batasan

- Batas ukuran database: 64 TB

Versi produk

- MySQL versi 5.5, 5.6, 5.7, 8.0. Untuk daftar terbaru versi yang didukung, lihat MySQL di Amazon AWS RDS dalam dokumentasi. Jika Anda menggunakan AWS DMS, lihat juga Menggunakan Database yang kompatibel dengan MySQL sebagai Target untuk versi AWS DMS MySQL yang saat ini didukung oleh. AWS DMS

Arsitektur

Tumpukan teknologi sumber

- Database MySQL lokal

Tumpukan teknologi target

- Instans Amazon RDS DB yang menjalankan MySQL

Arsitektur target

Diagram berikut menunjukkan target Amazon RDS untuk implementasi MySQL setelah migrasi.

AWS arsitektur migrasi data

Menggunakan AWS DMS:

Diagram berikut menunjukkan arsitektur migrasi data saat Anda gunakan AWS DMS untuk mengirim perubahan penuh dan bertahap hingga cutover. Koneksi jaringan dari tempat ke AWS tergantung pada kebutuhan Anda dan berada di luar cakupan untuk pola ini.

Menggunakan alat MySQL asli:

Diagram berikut menunjukkan arsitektur migrasi data saat Anda menggunakan alat MySQL asli. File dump ekspor disalin ke Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dan diimpor ke database Amazon RDS for MySQL sebelum cutover. AWS Koneksi jaringan dari tempat ke AWS tergantung pada kebutuhan Anda dan berada di luar cakupan untuk pola ini.

Catatan:

- Bergantung pada persyaratan downtime dan ukuran database, penggunaan AWS DMS atau alat change data capture (CDC) meminimalkan waktu cutover. AWS DMS dapat membantu mengurangi waktu cutover ke target baru seminimal mungkin (biasanya menit). Strategi offline dengan mysqldump bisa cukup jika ukuran database dan latensi jaringan memungkinkan jendela pendek. (Kami merekomendasikan pengujian untuk mendapatkan perkiraan waktu.)
- Biasanya strategi CDC seperti AWS DMS membutuhkan lebih banyak pemantauan dan kompleksitas daripada opsi offline.

Alat

- Layanan AWS: AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau di antara kombinasi pengaturan cloud dan lokal. Untuk

informasi tentang sumber MySQL dan basis data target yang didukung AWS DMS oleh, lihat Memigrasi Database yang kompatibel dengan MySQL ke. AWS Jika database sumber Anda tidak didukung oleh AWS DMS, Anda harus memilih metode lain untuk memigrasi data Anda.

- Alat MySQL asli: mysqldump
- Alat pihak ketiga: Percona XtraBackup

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi database.	Validasi versi basis data sumber dan target.	DBA
Identifikasi persyaratan perangkat keras.	Identifikasi persyaratan perangkat keras untuk server target.	DBA, Administrator sistem
Identifikasi persyaratan penyimpanan.	Identifikasi persyaratan penyimpanan (seperti jenis dan kapasitas penyimpanan) untuk basis data target.	DBA, Administrator sistem
Pilih jenis instans.	Pilih jenis instans target berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.	DBA, Administrator sistem
Identifikasi persyaratan akses jaringan.	Identifikasi persyaratan keamanan untuk akses jaringan untuk basis data sumber dan target.	DBA, Administrator sistem
Identifikasi objek yang tidak didukung.	Identifikasi objek yang tidak didukung (jika ada) dan tentukan upaya migrasi.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi dependensi.	Identifikasi dependensi apa pun pada database jarak jauh.	DBA
Tentukan strategi migrasi aplikasi.	Tentukan strategi untuk memigrasi aplikasi klien.	DBA, Pemilik aplikasi, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).	Konfigurasikan tabel rute, gateway internet, gateway NAT, dan subnet. Untuk informasi selengkapnya, lihat VPCs dan Amazon RDS dalam dokumentasi Amazon RDS.	Administrator sistem
Buat grup keamanan.	Konfigurasikan port dan rentang CIDR atau spesifik IPs tergantung pada kebutuhan Anda. Port default untuk MySQL adalah 3306. Untuk informasi selengkapnya, lihat Mengontrol akses dengan grup keamanan di dokumenta si Amazon RDS.	Administrator sistem
Konfigurasikan dan mulai Amazon RDS for MySQL DB instans.	Untuk petunjuknya, lihat Membuat instans Amazon RDS DB dalam dokumentasi Amazon RDS. Periksa versi yang didukung.	Administrator sistem

Migrasikan data - opsi 1 (menggunakan alat asli)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan alat MySQL asli atau alat pihak ketiga untuk memigrasikan objek dan data database.	Untuk petunjuk, lihat dokumentasi untuk alat MySQL seperti mysqldump dan Percona (untuk migrasi fisik). XtraBackup Untuk informasi selengkapnya tentang opsi, lihat posting blog Opsi migrasi untuk MySQL ke Amazon RDS for MySQL atau Amazon Aurora MySQL.	DBA

Migrasi data - opsi 2 (menggunakan AWS DMS)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan AWS DMS.	Untuk instruksi, lihat AWS DMS dokumentasi.	DBA

Lakukan tugas awal sebelum cutover

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbaiki perbedaan jumlah objek.	Kumpulkan jumlah objek dari database sumber dan basis data target baru. Perbaiki perbedaan dalam database target.	DBA
Periksa dependensi.	Periksa apakah dependensi (tautan) ke dan dari database	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lain valid dan berfungsi seperti yang diharapkan.	
Lakukan tes.	Jika ini adalah siklus pengujian, lakukan pengujian kueri, kumpulkan metrik, dan perbaiki masalah.	DBA

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Beralih ke database target.	Alihkan aplikasi klien ke infrastruktur baru.	DBA, Pemilik aplikasi, Administrator sistem
Berikan dukungan pengujian.	Memberikan dukungan untuk pengujian aplikasi fungsional.	DBA

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya.	Matikan AWS sumber daya sementara yang Anda buat untuk migrasi.	DBA, Administrator sistem
Validasi dokumen proyek.	Meninjau dan memvalidasi dokumen proyek.	DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik.	Kumpulkan metrik seperti waktu untuk bermigrasi, persentase upaya manual versus otomatis, penghematan biaya, dan sebagainya.	DBA, Pemilik aplikasi, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tutup proyek.	Tutup proyek dan berikan umpan balik.	DBA, Pemilik aplikasi, Administrator sistem
Menonaktifkan database sumber.	Ketika semua tugas migrasi dan cutover selesai, nonaktifk an database lokal.	DBA, Administrator sistem

Sumber daya terkait

Referensi

- Strategi migrasi untuk database relasional
- AWS DMS situs web
- AWS DMS dokumentasi
- Dokumentasi Amazon RDS
- Harga Amazon RDS
- Amazon VPC dan Amazon RDS
- Penerapan Amazon RDS Multi-AZ
- Migrasikan database MySQL lokal ke Aurora MySQL menggunakan Percona, Amazon EFS, dan Amazon S3 XtraBackup
- Penyimpanan instans Amazon RDS DB

Tutorial

- Memulai dengan AWS DMS
- Memulai dengan Amazon RDS

Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server

Henrique Lobao, Jonathan Pereira Cruz, dan Vishal Singh, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk bermigrasi dari database Microsoft SQL Server lokal ke Amazon Relational Database Service (Amazon RDS) untuk SQL Server. Ini menjelaskan dua opsi untuk migrasi: menggunakan AWS Data Migration Service (AWS DMS) atau menggunakan alat Microsoft SQL Server asli seperti Copy Database Wizard.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Microsoft SQL Server sumber di pusat data lokal

Batasan

- Batas ukuran database: 16 TB

Versi produk

- Untuk daftar terbaru versi dan fitur yang didukung, lihat Microsoft SQL Server di Amazon RDS dalam dokumentasi AWS. Jika Anda menggunakan AWS DMS, lihat juga Menggunakan Database Microsoft SQL Server sebagai Target untuk AWS DMS untuk versi SQL Server yang didukung oleh AWS DMS.

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server lokal

Tumpukan teknologi target

- Instans Amazon RDS for SQL Server DB

Arsitektur sumber dan target

Menggunakan AWS DMS:

Menggunakan alat SQL Server asli:

Alat

- AWS DMS mendukung beberapa jenis basis data sumber dan target. Untuk detailnya, lihat AWS DMS Step-by-Step Walkthrough. Jika AWS DMS tidak mendukung basis data sumber, pilih metode lain untuk memigrasikan data.
- Alat Microsoft SQL Server asli termasuk backup dan restore, Copy Database Wizard, salin dan lampirkan database.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target dan mesin.		DBA
Identifikasi persyaratan perangkat keras untuk instance server target.		DBA, Administrator sistem
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, Administrator sistem
Pilih jenis instans yang tepat berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.		DBA, Administrator sistem
Identifikasi persyaratan keamanan akses jaringan		DBA, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
untuk basis data sumber dan target.		
Identifikasi strategi migrasi aplikasi.		DBA, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).		Administrator sistem
Buat grup keamanan.		Administrator sistem
Konfigurasikan dan mulai instans Amazon RDS DB.		DBA, Administrator sistem

Migrasikan data - opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan alat SQL Server asli atau alat pihak ketiga untuk memigrasikan objek dan data database.		DBA

Migrasikan data - opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan AWS DMS.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.		DBA, Pemilik aplikasi, Administrator sistem

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.		DBA, Pemilik aplikasi, Administrator sistem

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, Administrator sistem
Meninjau dan memvalidasi dokumen proyek.		DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik seperti waktu untuk bermigrasi, persentase tugas manual versus otomatis, dan penghematan biaya.		DBA, Pemilik aplikasi, Administrator sistem
Tutup proyek dan berikan umpan balik.		DBA, Pemilik aplikasi, Administrator sistem

Sumber daya terkait

Referensi

- Menyebarkan Microsoft SQL Server di Amazon Web Services
- Situs AWS DMS
- Harga Amazon RDS
- Produk Microsoft di AWS
- Lisensi Microsoft di AWS
- Microsoft SQL Server di AWS
- Menggunakan Otentikasi Windows dengan Instans Microsoft SQL Server DB
- Penyebaran Multi-AZ Amazon RDS

Tutorial dan video

- Memulai dengan AWS DMS
- Memulai dengan Amazon RDS
- AWS DMS (video)
- Amazon RDS (video)

Migrasi data dari Microsoft Azure Blob ke Amazon S3 dengan menggunakan Rclone

Suhas Basavaraj, Aidan Keane, dan Corey Lane, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan Rclone untuk memigrasikan data dari penyimpanan objek Microsoft Azure Blob ke bucket Amazon Simple Storage Service (Amazon S3). Anda dapat menggunakan pola ini untuk melakukan migrasi satu kali atau sinkronisasi data yang sedang berlangsung. Rclone adalah program baris perintah yang ditulis dalam Go dan digunakan untuk memindahkan data di berbagai teknologi penyimpanan dari penyedia cloud.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Data disimpan dalam layanan kontainer Azure Blob

Arsitektur

Tumpukan teknologi sumber

- Wadah penyimpanan Azure Blob

Tumpukan teknologi target

- Buket Amazon S3
- Contoh Amazon Elastic Compute Cloud (Amazon EC2) Linux

Arsitektur

Alat

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Rclone adalah program baris perintah open-source yang terinspirasi oleh rsync. Ini digunakan untuk mengelola file di banyak platform penyimpanan cloud.

Praktik terbaik

Saat Anda memigrasikan data dari Azure ke Amazon S3, perhatikan pertimbangan ini untuk menghindari biaya yang tidak perlu atau kecepatan transfer yang lambat:

- Buat infrastruktur AWS Anda di Wilayah geografis yang sama dengan akun penyimpanan Azure dan wadah Blob—misalnya, `us-east-1` Wilayah AWS (Virginia N.) dan wilayah Azure. `East US`
- Hindari menggunakan NAT Gateway jika memungkinkan, karena akan dikenakan biaya transfer data untuk bandwidth masuk dan keluar.
- Gunakan titik akhir gateway VPC untuk Amazon S3 untuk meningkatkan kinerja.
- Pertimbangkan untuk menggunakan EC2 instans berbasis prosesor AWS Graviton2 (ARM) untuk biaya yang lebih rendah dan kinerja yang lebih tinggi dibandingkan instans Intel x86. Rclone sangat dikompilasi silang dan menyediakan biner ARM yang telah dikompilasi sebelumnya.

Epik

Siapkan sumber daya cloud AWS dan Azure

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan ember S3 tujuan.	Buat bucket S3 baru di Wilayah AWS yang sesuai atau pilih bucket yang sudah ada sebagai tujuan data yang ingin Anda migrasikan.	Administrator AWS
Buat peran instans IAM untuk Amazon EC2.	Buat peran AWS Identity and Access Management (IAM) baru untuk Amazon EC2. Peran ini memberikan akses penulisan EC2 instance Anda ke bucket S3 tujuan.	Administrator AWS
Lampirkan kebijakan ke peran instans IAM.	Gunakan konsol IAM atau AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) untuk membuat kebijakan inline	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk peran EC2 instance yang memungkinkan izin akses tulis ke bucket S3 tujuan. Untuk kebijakan contoh, lihat bagian Informasi tambahan.	
Luncurkan sebuah EC2 instance.	Luncurkan EC2 instans Amazon Linux yang dikonfigurasi untuk menggunakan peran layanan IAM yang baru dibuat. Instans ini juga memerlukan akses ke titik akhir API publik Azure melalui internet. **Note** Pertimbangkan untuk menggunakan EC2 instans berbasis AWS Graviton untuk menurunkan biaya. Rclone menyediakan binari yang dikompilasi ARM.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kepala layanan Azure AD.	Gunakan Azure CLI untuk membuat prinsip layanan Azure Active Directory (Azure AD) yang memiliki akses hanya-baca ke wadah penyimpanan Azure Blob sumber. Untuk petunjuk, lihat bagian Informasi tambahan. Simpan kredensyal ini pada EC2 instans Anda ke lokasi. `~/azure-principal.json`	Administrator awan, Azure

Instal dan konfigurasikan Rclone

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh dan instal Rclone.	Unduh dan instal program baris perintah Rclone. Untuk petunjuk penginstalan, lihat dokumentasi instalasi Rclone.	AWS Umum, Administrator Cloud
Konfigurasikan Rclone.	Salin file `rclone.conf` contoh berikut. Ganti `AZStorageAccount` dengan nama akun Azure Storage Anda dan `us-east-1` dengan Wilayah AWS tempat bucket S3 Anda berada. Simpan file ini ke lokasi `~/.config/rclone/rclone.conf` pada EC2 instance Anda.	AWS Umum, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`[AZStorageAccount]`
	`type = azureblob`
	`account = AZStorage Account`
	`service_principal_f ile = azure-pri ncipal.json`
	
	`[s3]`
	`type = s3`
	`provider = AWS`
	`env_auth = true`
	`region = us-east-1`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi konfigurasi Rclone.	Untuk mengonfirmasi bahwa Rclone dikonfigurasi dan izin berfungsi dengan baik, verifikasi bahwa Rclone dapat mengurai file konfigurasi Anda dan objek di dalam wadah Azure Blob dan bucket S3 dapat diakses. Lihat berikut ini misalnya perintah validasi.
	• Buat daftar remote yang dikonfigurasi dalam file konfigurasi. Ini akan memastikan bahwa file konfigurasi Anda sedang diurai dengan benar. Tinjau output untuk memastikan bahwa itu cocok dengan `rclone.conf` file Anda.
	`rclone listremotes`
	`AZStorageAccount:`
	`s3:`
	• Buat daftar wadah Azure Blob di akun yang dikonfigurasi. Ganti `AZStorage Account` dengan nama akun penyimpanan yang Anda gunakan dalam `rclone.conf` file.
	`rclone lsd AZStorage`
	`Account:`
	`2020-04-29 08:29:26`
	`docs`
		AWS Umum, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Buat daftar file dalam wadah Azure Blob. Ganti dokumen dalam perintah ini dengan nama wadah Blob yang sebenarnya di akun penyimpanan Azure Anda.
	`rclone ls AZStorage Account:docs 824884 administr ator-en.a4.pdf`
	• Buat daftar bucket di akun AWS Anda.
	`[root@ip-10-0-20-157 ~]# rclone lsd s3: 2022-03-07 01:44:40 amzn-s3-demo-bucket1 2022-03-07 01:45:16 amzn-s3-demo-bucket2 2022-03-07 02:12:07 amzn-s3-demo-bucket3`
	• Buat daftar file di ember S3.
	`[root@ip-10-0-20-1 57 ~]# rclone ls s3:amzn-s3-demo-bu cket1 template0.yaml template1.yaml`	

Migrasi data menggunakan Rclone

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Migrasikan data dari kontainer Anda.	Jalankan perintah Rclone copy atau sync.
	
	Contoh: copy
	
	Perintah ini menyalin data dari wadah Azure Blob sumber ke bucket S3 tujuan.
	
	`rclone copy AZStorageAccount:blob-container s3:amzn-s3-demo-bucket1`
	
	Contoh: sinkronisasi
	
	Perintah ini menyinkronkan data antara wadah Azure Blob sumber dan bucket S3 tujuan.
	
	`rclone sync AZStorageAccount:blob-container s3:amzn-s3-demo-bucket1`
	
	Important
	Saat Anda menggunakan perintah sinkronisasi, data yang tidak ada di wadah sumber akan dihapus dari bucket S3 tujuan.
		AWS Umum, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sinkronkan wadah Anda.	Setelah salinan awal selesai, jalankan perintah sinkronisasi Rclone untuk migrasi yang sedang berlangsung sehingga hanya file baru yang hilang dari bucket S3 tujuan yang akan disalin.	AWS Umum, Administrator Cloud
Verifikasi bahwa data telah berhasil dimigrasi.	Untuk memeriksa apakah data berhasil disalin ke bucket S3 tujuan, jalankan perintah Rclone lsd dan ls.	AWS Umum, Administrator Cloud

Sumber daya terkait

- Panduan Pengguna Amazon S3 (dokumentasi AWS)
- Peran IAM untuk Amazon EC2 (dokumentasi AWS)
- Membuat wadah Microsoft Azure Blob (dokumentasi Microsoft Azure)
- Perintah Rclone (dokumentasi Rclone)

Informasi tambahan

Contoh kebijakan peran untuk EC2 instance

Kebijakan ini memberikan akses baca dan tulis EC2 instans Anda ke bucket tertentu di akun Anda. Jika bucket Anda menggunakan kunci terkelola pelanggan untuk enkripsi sisi server, kebijakan tersebut mungkin memerlukan akses tambahan ke AWS Key Management Service (AWS KMS).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:DeleteObject",
```

```
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
              "arn:aws:s3:::amzn-s3-demo-bucket/*",
              "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

Membuat prinsipal layanan Azure AD hanya-baca

Prinsipal layanan Azure adalah identitas keamanan yang digunakan oleh aplikasi pelanggan, layanan, dan alat otomatisasi untuk mengakses sumber daya Azure tertentu. Anggap saja sebagai identitas pengguna (login dan kata sandi atau sertifikat) dengan peran tertentu dan izin yang dikontrol ketat untuk mengakses sumber daya Anda. Untuk membuat prinsipal layanan hanya-baca agar mengikuti izin hak istimewa paling sedikit dan melindungi data di Azure dari penghapusan yang tidak disengaja, ikuti langkah-langkah berikut:

1. Masuk ke portal akun cloud Microsoft Azure Anda dan luncurkan Cloud Shell PowerShell atau gunakan Antarmuka Baris Perintah Azure (CLI) di workstation Anda.
2. Buat prinsip layanan dan konfigurasikan dengan akses hanya-baca ke akun penyimpanan Azure Blob Anda. Simpan output JSON dari perintah ini ke file lokal bernama`azure-principal.json`. File akan diunggah ke EC2 instance Anda. Ganti variabel placeholder yang ditampilkan dalam tanda kurung gigi ({dan}) dengan ID langganan Azure, nama grup sumber daya, dan nama akun penyimpanan Anda.

```
az ad sp create-for-rbac `
--name AWS-Rclone-Reader `
--role "Storage Blob Data Reader" `
--scopes /subscriptions/{Subscription ID}/resourceGroups/{Resource Group Name}/
providers/Microsoft.Storage/storageAccounts/{Storage Account Name}
```

Migrasi dari Couchbase Server ke Couchbase Capella di AWS

Battulga Purevragchaa dan Saurabh Shanbhag, Amazon Web Services

Mark Gamble, Tidak Ada

Ringkasan

Couchbase Capella adalah database NoSQL yang dikelola sepenuhnya sebagai layanan (DBaaS) untuk aplikasi mission-critical (misalnya, profil pengguna atau katalog online dan manajemen inventaris). Couchbase Capella mengelola beban kerja DBaa S Anda di akun Amazon Web Services (AWS) yang dikelola Couchbase. Capella memudahkan untuk menjalankan dan mengelola replikasi multi-cluster, Multiple—AWS Region, multicloud, dan hybrid-cloud dalam satu antarmuka.

Couchbase Capella membantu Anda secara instan menskalakan aplikasi Server Couchbase Anda, membantu Anda membuat klaster multi-node dalam hitungan menit. Couchbase Capella mendukung semua fitur Server Couchbase, termasuk SQL++, Pencarian Teks Lengkap, Layanan Eventing, dan Layanan Analytics. Ini juga menghilangkan kebutuhan untuk mengelola instalasi, upgrade, backup, dan pemeliharaan database umum.

Pola ini menjelaskan langkah-langkah dan praktik terbaik untuk memigrasikan lingkungan Server Couchbase yang dikelola sendiri ke AWS Cloud. Pola ini menyediakan proses berulang untuk memigrasikan data dan indeks dari kluster Couchbase Server, berjalan baik di tempat atau di cloud, ke Couchbase Capella. Menggunakan langkah-langkah ini membantu Anda menghindari masalah selama migrasi dan mempercepat proses migrasi secara keseluruhan.

Pola ini menyediakan dua opsi migrasi berikut:

- Opsi 1 sesuai jika Anda memiliki kurang dari 50 indeks untuk dimigrasi.
- Opsi 2 sesuai jika Anda memiliki lebih dari 50 indeks untuk dimigrasi.

Anda juga dapat mengatur data sampel di Server Couchbase yang dikelola sendiri untuk mengikuti panduan migrasi.

Jika Anda memilih opsi migrasi 2, atau jika Anda menggunakan cakupan atau koleksi selain nilai default, Anda harus menggunakan contoh file konfigurasi, yang ada di bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Akun berbayar Couchbase Capella yang ada. Anda juga dapat membuat akun Couchbase Capella di AWS dan menggunakan uji coba gratis Couchbase Capella, lalu memutakhirkan ke akun berbayar untuk mengonfigurasi kluster Anda untuk migrasi.. Untuk memulai dengan versi uji coba, ikuti petunjuk di Memulai dengan Couchbase Capella.
- Lingkungan Server Couchbase yang dikelola sendiri yang ada baik di tempat atau digunakan di penyedia layanan cloud.
- Untuk opsi migrasi 2, Couchbase Shell dan file konfigurasi. Untuk membuat file konfigurasi, Anda dapat menggunakan file contoh yang ada di bagian Informasi tambahan.
- Keakraban dengan mengelola Couchbase Server dan Couchbase Capella.
- Keakraban dengan membuka port TCP dan menjalankan perintah dalam antarmuka baris perintah (CLI).

Proses migrasi juga membutuhkan peran dan keahlian yang dijelaskan dalam tabel berikut.

Peran	Keahlian	Tanggung jawab
Administrator Couchbase	• Keakraban dengan Couchbase Server dan Couchbase Capella • Pengetahuan baris perintah dasar sangat membantu tetapi tidak diperlukan	• Server Couchbase dan tugas khusus Capella
Administrator sistem, administrator TI	• Keakraban dengan lingkungan dan administrasi sistem Couchbase Server yang dikelola sendiri	• Membuka port dan menentukan alamat IP pada node cluster Couchbase Server yang dikelola sendiri

Batasan

- Pola ini digunakan untuk memigrasikan data, indeks, dan indeks Pencarian Teks Lengkap Couchbase dari Couchbase Server ke Couchbase Capella di AWS. Pola ini tidak berlaku untuk memigrasi Couchbase Eventing Service, atau ke Couchbase Analytics.
- Couchbase Capella tersedia di beberapa Wilayah AWS. Untuk up-to-date informasi tentang Wilayah yang didukung Capella, lihat Amazon Web Services di dokumentasi Couchbase.

Versi produk

- Couchbase Server (Komunitas atau Perusahaan) Edition versi 5.x atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Server Couchbase

Tumpukan teknologi target

- Couchbase Capella

Arsitektur target

1. Anda mengakses Couchbase Capella dengan menggunakan Capella Control Plane. Anda dapat menggunakan Capella Control Plane untuk melakukan hal berikut:
   - Kontrol dan pantau akun Anda.
   - Kelola cluster dan data, indeks, pengguna dan grup, izin akses, pemantauan, dan acara.
2. Cluster dibuat.
3. Capella Data Plane ada di akun AWS yang dikelola Couchbase. Setelah Anda membuat klaster baru, Couchbase Capella menerapkannya di beberapa Availability Zone di Wilayah AWS yang dipilih.
4. Anda dapat mengembangkan dan menerapkan aplikasi Couchbase di VPC di akun AWS Anda. Biasanya, VPC ini mengakses Capella Data Plane melalui VPC peering.

Alat

- Couchbase Cross Data Center Replication (XDCR) membantu mereplikasi data di seluruh cluster yang terletak di penyedia cloud yang berbeda dan pusat data yang berbeda. Ini digunakan untuk memigrasikan data ke Couchbase Capella dari cluster Server Couchbase yang dikelola sendiri.

> **Note**
>
> XDCR tidak dapat digunakan dengan Couchbase Server Community Edition untuk bermigrasi ke Couchbase Capella. Sebagai gantinya, Anda dapat menggunakan cbexport. Untuk informasi selengkapnya, lihat epik Migrasi data dari Edisi Komunitas.

- Couchbase Shell adalah shell baris perintah untuk Couchbase Server dan Couchbase Capella untuk mengakses cluster Couchbase lokal dan jarak jauh. Dalam pola ini, Couchbase Shell digunakan untuk memigrasikan indeks.
- cbexport adalah utilitas Couchbase untuk mengekspor data dari cluster Couchbase. Termasuk dalam alat CLI Server Couchbase.

Epik

Siapkan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Evaluasi ukuran cluster Couchbase Server yang dikelola sendiri.	Masuk ke Konsol Web Couchbase untuk Server Couchbase, dan nilai node dan bucket cluster yang dikelola sendiri. 1. Untuk menampilkan daftar node cluster, pilih tab Server di bilah navigasi. 2. Catat jumlah node, lalu pilih setiap node pada daftar untuk menampilkan propertinya. 3. Rekam memori dan penyimpanan untuk setiap node individu. 4. Pilih tab Bucket di bilah navigasi, lalu pilih setiap	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bucket dalam daftar untuk menampilkan propertin ya. Catat kuota RAM dan pengaturan resolusi konflik untuk setiap bucket. Anda akan menggunakan konfigurasi cluster Couchbase Server yang dikelola sendiri sebagai panduan umum untuk mengukur dan mengonfigurasi cluster target di Couchbase Capella. Untuk bantuan dengan latihan ukuran Couchbase Capella yang lebih detail, hubungi Couchbase.	
Rekam distribusi Layanan Couchbase di cluster Couchbase Server yang dikelola sendiri.	1. Pada Couchbase Web Console, pilih tab Server untuk menampilkan daftar node cluster. 2. Pilih setiap node untuk menampilkan propertin ya dan kemudian rekam distribusi Layanan Couchbase untuk setiap node (Layanan Data, Layanan Kueri, Layanan Indeks, Layanan Pencarian , Layanan Analisis, dan Layanan Eventing).	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rekam alamat IP node cluster Couchbase Server yang dikelola sendiri.	(Abaikan langkah ini jika Anda menggunakan Edisi Komunitas.) Catat alamat IP untuk setiap node di cluster Anda. Mereka akan ditambahkan ke daftar izinkan di cluster Couchbase Capella Anda nanti.	Administrator Couchbase, Administrator sistem

Terapkan dan konfigurasikan sumber daya di Couchbase Capella

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih templat.	1. Masuk ke Couchbase Capella Control Plane Anda, pilih tab Dashboard atau tab Clusters di navigasi utama, lalu pilih Create Cluster. 2. Menggunakan informasi yang Anda rekam dari evaluasi cluster Couchbase Server yang dikelola sendiri, pilih template cluster yang memenuhi persyaratan konfigurasi. Jika Anda tidak menemukan template yang sesuai, pilih Template Kustom di editor Ukuran Cluster.	Administrator Couchbase
Pilih dan konfigurasikan node.	Pilih dan konfigurasikan node agar sesuai dengan	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lingkungan cluster Couchbase Server yang dikelola sendiri, termasuk jumlah node, distribusi layanan, komputasi atau RAM, dan penyimpanan. Couchbase Capella menggunakan praktik terbaik penskalaan multidimensi. Layanan dan node hanya dapat dipilih sesuai dengan praktik terbaik penerapan . Ini mungkin berarti bahwa Anda tidak dapat sama persis dengan konfigurasi cluster Couchbase Server yang dikelola sendiri.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan cluster.	Pilih zona dukungan dan paket dukungan, lalu gunakan cluster. Untuk langkah dan instruksi terperinci, lihat Membuat klaster di dokumentasi Couchbase. **Important** Jika Anda menggunakan uji coba gratis Couchbase Capella, Anda harus mengonversinya menjadi akun berbayar sebelum memulai migrasi. Untuk mengonversi akun Anda, buka bagian Penagihan pada Couchbase Capella Control Plane, lalu pilih Tambahkan ID Aktivasi. ID Aktivasi dikirim ke alamat email kontak penagihan Anda setelah Anda menyelesaikan perjanjian pembelian dengan Couchbase Sales, atau setelah Anda melakukan pembelian melalui AWS Marketplace.	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna kredenal basis data.	Pengguna kredenal basis data khusus untuk klaster dan terdiri dari nama pengguna, kata sandi, dan sekumpula n hak istimewa bucket. Pengguna ini diperlukan untuk membuat bucket dan mengakses data bucket. Di Couchbase Capella Control Plane, buat kredensi database untuk cluster baru dengan mengikuti petunjuk di Configure database credentials dalam dokumenta si Couchbase Capella. **Note** Pengguna organisasi memerlukan kredensi peran organisasi yang ditetapkan kepada mereka jika mereka ingin mengakses data bucket pada klaster tertentu, baik dari jarak jauh atau melalui UI Couchbase Capella. Ini terpisah dari kredensi database, yang biasanya digunakan oleh aplikasi dan integrasi . Membuat pengguna	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	organisasi memungkinkan Anda membuat dan mengelola bucket target di cluster Couchbase Capella Anda.	
Jika menggunakan opsi migrasi 2, instal Couchbase Shell.	Anda dapat menginstal Couchbase Shell pada sistem apa pun yang memiliki akses jaringan ke Server Couchbase yang dikelola sendiri dan cluster Couchbase Capella. Untuk informasi selengkapnya, lihat Menginstal Couchbase Shell versi 1.0.0-beta.5 di dokumentasi Couchbase Shell. Konfirmasikan bahwa Couchbase Shell diinstal dengan menguji koneksi ke cluster yang dikelola sendiri di terminal baris perintah.	Administrator Couchbase, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Izinkan alamat IP.	1. Di Couchbase Capella Control Plane, pilih Clusters, lalu pilih cluster target Anda. 2. Pilih tab Connect untuk klaster, dan rekam titik akhir koneksi untuk klaster Anda yang tercantum di bawah Kelola IP yang Diizinkan. 3. Untuk menambahkan alamat IP untuk sistem tempat Anda menginsta l Couchbase Shell dan alamat IP instance cluster Couchbase Server yang dikelola sendiri sebagai alamat IP yang diizinkan, lakukan hal berikut: a. Di bawah Jaringan Area Luas, pilih Kelola IP yang Diizinkan. b. Pilih Add Allowed IP, masukkan alamat IP untuk sistem tempat Anda menginstal Couchbase Shell, lalu pilih Add IP. c. Ulangi langkah sebelumnya untuk menambahkan alamat IP instance cluster Couchbase Server yang dikelola sendiri.	Administrator Couchbase, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkap nya tentang alamat IP yang diizinkan, lihat Mengonfig urasi alamat IP yang diizinkan dalam dokumentasi Couchbase.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan sertifikat.	1. Untuk mengunduh sertifikat root untuk klaster Anda, di bawah Root Certificate, pilih Unduh. 2. Simpan sertifikat root menggunakan ekstensi file.pem di folder pada sistem yang akan menjalankan Couchbase Shell. 3. Selanjutnya, masuk ke Konsol Web Server Couchbase yang dikelola sendiri, pilih Keamanan di bilah navigasi kiri, lalu pilih tab Sertifikat. 4. Salin sertifikat root untuk cluster Couchbase Server yang dikelola sendiri dan simpan sebagai file.pem ke folder yang sama tempat Anda menyimpan file sertifikat root untuk cluster Couchbase Capella Anda. Untuk informasi selengkapnya tentang sertifikat root, lihat sertifikat Root di dokumentasi Couchbase Server.	Administrator Couchbase, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file konfigurasi untuk Couchbase Shell.	Buat dotfile konfigurasi di direktori home instalasi Couchbase Shell (misalnya ,). `/<HOME_DIRECTORY>/.cbsh/config` Untuk informasi selengkapnya, lihat Config dotfiles dalam dokumentasi Couchbase. Tambahkan properti koneksi untuk cluster sumber dan target ke file konfigurasi. Anda dapat menggunakan contoh file konfigurasi yang ada di bagian Informasi tambahan dan mengedit pengaturan untuk cluster Anda. Simpan file konfigurasi dengan pengaturan yang diperbaru i ke `.cbsh` folder (misalnya ,`/<HOME_DIRECTORY>/.cbsh/config`).	Administrator Couchbase, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember target.	Untuk setiap bucket sumber, buat satu bucket target di cluster Couchbase Capella Anda dengan mengikuti petunjuk di Buat bucket dalam dokumentasi Couchbase. Konfigurasi bucket target Anda harus sesuai dengan nama bucket, pengatura n memori, dan pengatura n resolusi konflik bucket di cluster Couchbase Server yang dikelola sendiri.	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat cakupan dan koleksi.	Setiap bucket berisi cakupan dan koleksi default dengan keyspace_default._default . Jika Anda menggunakan ruang kunci lain untuk cakupan dan koleksi Anda, Anda harus membuat ruang kunci yang identik di cluster Capella target.
	
	1. Buka terminal baris perintah pada sistem tempat Anda menginstal Couchbase Shell.
	2. Untuk memulai Couchbase Shell, jalankan perintah berikut.
	
	`./cbsh`
	
	3. Untuk setiap bucket yang ingin dimigrasikan, buat cakupan dan koleksi di cluster Capella dengan menjalankan perintah berikut. Pastikan Anda mengganti <BUCKET_NAME>dengan nama bucket yang ingin dimigrasikan.
	
	`scopes --clusters "On-Prem-Cluster" --bucket <BUCKET_NAME> | select scope | where scope != "_default" | each`
		Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`{ |it| scopes create $it.scope --clusters "Capella-Cluster" } collections --clusters "On-Prem-Cluster" --bucket <BUCKET_NAME> | select scope collection | where $it.scope != "_default" | where $it.collection != "_default" | each { |it| collections create $it.collection --clusters "Capella-Cluster" --bucket <BUCKET_NAME> --scope $it.scope }`	

Migrasi data dari Enterprise Edition

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka port TCP pada node cluster Couchbase Server yang dikelola sendiri.	Pastikan port yang sesuai terbuka untuk komunikasi XDCR pada node cluster Couchbase Server yang dikelola sendiri. Untuk informasi selengkapnya, lihat dokumentasi port Server Couchbase.	Administrator Couchbase, Administrator sistem
Jika Anda menggunakan Couchbase Server Enterprise Edition, siapkan Couchbase XDCR.	1. Di navigasi utama Couchbase Capella Control Plane, pilih Clusters, lalu pilih cluster target untuk migrasi.	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di bawah Root Certificate, pilih Salin. 3. Masuk ke Konsol Web Server Couchbase yang dikelola sendiri, dan di navigasi utama, pilih XDCR. Kemudian pilih Add Remote. 4. Masukkan setelan berikut: • Nama Cluster — Nama untuk koneksi cluster Capella • IP/HostName - Titik akhir koneksi untuk cluster Couchbase Capella Anda • Nama pengguna untuk Remote Cluster - Pengguna database untuk cluster Couchbase Capella Anda • Kata sandi — Kata sandi pengguna database untuk cluster Couchbase Capella Anda • Aktifkan Koneksi Aman — Dipilih • Lengkap (TLS mengenkripsi kata sandi dan data) - Dipilih 5. Rekatkan sertifikat root cluster Capella yang Anda salin sebelumnya, lalu pilih Simpan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai Couchbase XDCR.	1. Di Konsol Web Server Couchbase yang dikelola sendiri, pilih XDCR di navigasi utama, lalu pilih Tambah Replikasi. 2. Masukkan setelan berikut: • Replikasi Dari Bucket — Pilih keranjang sumber untuk migrasi. • Remote Bucket — Masukkan nama bucket target. • Remote Cluster - Pilih cluster target yang Anda buat sebelumnya. 3. Pilih Simpan Replikasi . Proses replikasi akan dimulai dalam beberapa detik.	Administrator Couchbase

Migrasikan indeks dengan menggunakan opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan indeks cluster yang dikelola sendiri ke Couchbase Capella.	**Important** Kami merekomendasikan proses ini jika Anda memiliki kurang dari 50 indeks untuk dimigrasi. Jika Anda memiliki lebih dari 50 indeks untuk dimigrasi	Administrator Couchbase, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	, sebaiknya gunakan opsi migrasi 2. 1. Di Konsol Web Couchbase, pilih Indeks. 2. Dalam daftar indeks, pilih indeks pertama yang ingin Anda migrasi. Definisi indeks kemudian ditampilkan. 3. Salin definisi indeks dengan menggunakan `CREATE` pernyataan, tetapi jangan salin`WITH { "defer_build":true }` . Misalnya, dari definisi indeks contoh berikut, Anda hanya akan menyalin`CREATE INDEX` `` `cityindex` ON `travel-sample`(`city`) `` . `CREATE INDEX `cityindex` ON `travel-sample`(`city`) WITH { "defer_build":true }` 4. Di Couchbase Capella Control Plane, pilih Clusters, lalu pilih cluster target.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Pada daftar dropdown Tools, pilih Query Workbench. Tempelkan `CREATE` pernyataan yang Anda salin sebelumnya ke Editor Kueri, lalu pilih Jalankan. Ini membuat dan membangun indeks. 6. Untuk mengonfirmasi bahwa indeks dibuat, pilih Indeks dari daftar dropdown Alat. Daftar menunjukkan bahwa indeks dibuat dan dibangun. 7. Ulangi proses ini untuk setiap indeks yang harus dimigrasikan.	

Migrasikan indeks dengan menggunakan opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan definisi indeks.	**Important** Kami merekomendasikan proses ini jika Anda memiliki lebih dari 50 indeks untuk dimigrasi. Jika Anda memiliki kurang dari 50 indeks untuk dimigrasi, sebaiknya	Administrator Couchbase, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	gunakan opsi migrasi 1.
	
	1. Buka terminal baris perintah pada sistem tempat Anda menginstal Couchbase Shell.
	2. Untuk memulai Couchbase Shell, jalankan perintah berikut.
	
	`./cbsh`
	
	3. Untuk menyambung ke cluster Couchbase Server yang dikelola sendiri, jalankan perintah berikut.
	
	`cb-env cluster On-Prem-Cluster`
	
	4. Untuk memigrasikan definisi indeks dari cluster Couchbase Server yang dikelola sendiri ke cluster Couchbase Capella, jalankan perintah berikut untuk setiap bucket yang ingin Anda migrasikan. Pastikan Anda mengganti `<BUCKET_NAME>` dengan nama bucket yang sesuai dengan indeks yang ingin dimigrasikan. Opsi migrasi ini mengharuskan nama	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bucket target Anda identik dengan nama bucket sumber. `query indexes --definitions | where bucket =~ <BUCKET_NAME>| get definition | each { |it| query $it --clusters Capella-Cluster }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Membangun definisi indeks.	1. Untuk mengalihkan konteks ke cluster Couchbase Capella, jalankan perintah berikut:
	`cb-env cluster Capella-Cluster`
	2. Untuk membuat definisi indeks yang dimigrasikan ke cluster Couchbase Capella, jalankan perintah berikut, ganti <BUCKET_NAME> dengan nama bucket yang sesuai dengan indeks yang ingin Anda buat.
	`query 'SELECT RAW CONCAT("BUILD INDEX ON ", k , "(['", CONCAT2 ("','", inames), "']);") FROM system:indexes AS s LET bid = CONCAT("` ",s.bucket_id, "`"), sid = CONCAT("`", s.scope_id, "`"), kid = CONCAT("` ", s.keyspace_id, "`"), k = NVL2(bid, CONCAT2(".", bid, sid, kid), kid) WHERE s.namespace_id = "default" AND s.bucket_id = "" GROUP BY k LETTING inames = ARRAY_AGG(s.name) FILTER`
		Administrator Couchbase, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(WHERE s.state = 'deferred') HAVING ARRAY_LENGTH(iname s) > 0;' | each { | it| query $it } 3. Ulangi untuk setiap ember.	

Migrasikan indeks pencarian teks lengkap

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan indeks pencarian teks lengkap klaster yang dikelola sendiri ke Couchbase Capella.	1. Di Konsol Web Couchbase, pilih Cari. 2. Dalam daftar indeks pencarian teks lengkap (FTS), pilih indeks FTS pertama yang ingin Anda migrasi, pilih Tampilkan definisi indeks JSON, dan pilih Salin ke Papan Klip. Catat nama indeks dan ember miliknya. 3. Di Couchbase Capella Control Plane, pilih Clusters dan kemudian pilih cluster target. 4. Pada daftar dropdown Tools, pilih Pencarian Teks Lengkap. 5. Pilih Indeks Impor, dan tempel definisi indeks FTS. 6. Masukkan Nama Indeks, pilih Bucket yang benar,	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	seperti yang tercantum pada klaster yang dikelola sendiri, lalu pilih Buat. 7. Ulangi proses ini untuk setiap indeks FTS yang harus dimigrasikan.	

Migrasi data dari Couchbase Community Edition

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ekspor data dari Edisi Komunitas Server Couchbase yang dikelola sendiri.	XDCR terenkripsi tidak tersedia di Couchbase Community Edition. Anda dapat mengekspor data dari Couchbase Community Edition dan kemudian mengimpor data secara manual ke Couchbase Capella.
	
	Untuk mengekspor data dari bucket sumber, gunakan `cbexport` pada baris perintah.
	
	Perintah berikut diberikan sebagai contoh.
	
	`cbexport json \`
	`--cluster localhost \`
	`--bucket <SOURCE BUCKET NAME> \`
	`--format lines \`
	`--username <USERNAME> \`
		Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--password <PASSWORD> \`
	`--include-key cbkey \`
	`--scope-field cbscope \`
	`--collection-field cbcoll \`
	`--output cbexported_data.json`
	
	Perhatikan bahwa`cbkey`,, `cbscope`, `cbcoll`, dan `cbexported_data.json` merupakan label sewenang-wenang. Mereka akan direferensikan nanti dalam proses, jadi jika Anda memilih untuk menamainya secara berbeda, catat itu.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Impor data ke Couchbase Capella.	1. Di Couchbase Capella Control Plane, pilih Clusters, lalu pilih cluster target. 2. Dalam daftar dropdown Tools, pilih Impor. Ini akan membuka wizard dengan enam langkah berikut: a. Bucket — Pilih ember target. b. File — Pilih JSON, pilih Garis, lalu pilih Menggunakan browser web Anda. Jika Anda memiliki sejumlah besar data, Anda dapat menjelajahi opsi Manual. Pilih file yang dibuat oleh `cbexport`. c. Koleksi — Pilih Pemetaan Koleksi Kustom. Jika database Edisi Komunitas Anda tidak menggunakan cakupan atau koleksi, atau hanya menggunakan _default, Anda dapat memilih opsi Pilih Koleksi Tunggal sebagai gantinya. Untuk Ekspresi Pemetaan Koleksi,	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	masukkan`%cbscope% .%cbcoll%` . Untuk memverifikasi bahwa ekspresi ini berfungsi dengan benar, Anda dapat menempelkan data contoh, seperti berikut ini. `{ "cbscope" :"inventory", "cbcoll":"landmark ", "cbkey":" landmark_3991" }` d. Kunci - Pilih Generasi Pelanggan. (Jika Anda tidak peduli tentang mempertahankan kunci data yang Anda impor, Anda dapat memilih UUID yang Dihasilka n Secara Otomatis sebagai gantinya dan melanjutkan ke langkah 5.) Untuk Ekspresi Generator Nama Kunci, masukkan %cbkey %. Untuk memverifi kasi bahwa ekspresi ini berfungsi dengan benar, tempel beberapa contoh data. e. Konfigurasi - Pilih Abaikan bidang, dan masukkan cbscope,	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	cbcoll, cbkey. Bidang ini berisi informasi sementara yang tidak perlu ada di bucket target setelah impor. Biarkan pengaturan lainnya tetap default. f. Impor - Tinjau, dan pilih Impor saat Anda siap. Tunggu unggahan dan impor data. Untuk file besar, Couchbase Capella mendukung impor baris perintah menggunakan cURL. Anda dapat menjelajahi opsi Impor secara lebih rinci di Impor data dalam dokumenta si Couchbase Capella.	

Uji dan verifikasi migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi migrasi data.	1. Di Couchbase Capella Control Plane, pilih Clusters, lalu pilih cluster target dalam daftar cluster Anda. 2. Pilih tab Bucket untuk kluster target Anda. Verifikasi bahwa jumlah Item (dokumen) di bucket	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	target cocok dengan jumlah item di bucket sumber. 3. Di cluster target, di daftar dropdown Tools, pilih Documents. Verifikasi bahwa semua dokumen telah dimigrasikan. 4. (Opsional) Setelah semua data dimigrasikan, Anda dapat mematikan replikasi dengan menghapusnya. Untuk informasi selengkapnya, lihat Menghapus replikasi dalam dokumentasi Couchbase.	
Verifikasi migrasi indeks.	Di Couchbase Capella Control Plane, di daftar dropdown Tools untuk cluster target Anda, pilih Indexes. Verifikasi bahwa indeks dimigrasikan dan dibuat.	Administrator Couchbase

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi hasil kueri.	1. Di Couchbase Capella Control Plane, di daftar dropdown Tools untuk cluster target Anda, pilih Query Workbench. 2. Jalankan contoh kueri N1QL atau kueri yang digunakan dalam aplikasi Anda. Pastikan Anda menerima hasil yang sama seperti saat menjalankan kueri di cluster Couchbase Server yang dikelola sendiri.	Administrator Couchbase
Verifikasi hasil pencarian teks lengkap (berlaku jika Anda memigrasikan indeks FTS).	1. Di Couchbase Capella Control Plane, di daftar dropdown Tools untuk cluster target Anda, pilih Pencarian Teks Lengkap. 2. Pilih indeks FTS dengan memilih namanya. 3. Pilih Cari. 4. Masukkan contoh permintaan pencarian, dan pilih Cari. 5. Verifikasi bahwa hasilnya sama seperti saat Anda menjalankan pencarian di klaster yang dikelola sendiri.	Administrator Couchbase

Sumber daya terkait

Siapkan migrasi

- Memulai dengan uji coba gratis Couchbase Capella
- Persyaratan penyedia cloud untuk Couchbase Capella
- Pedoman ukuran Couchbase Capella

Migrasi data dan indeks

- Couchbase XDCR
- Dokumentasi Couchbase Shell

Couchbase SLAs Capella dan dukungan

- Perjanjian tingkat layanan Couchbase Capella () SLAs
- Kebijakan dukungan Couchbase Capella Service

Informasi tambahan

Kode berikut adalah contoh file konfigurasi untuk Couchbase Shell.

```
Version = 1

[[clusters]]
identifier = "On-Prem-Cluster"
hostnames = ["<SELF_MANAGED_COUCHBASE_CLUSTER>"]
default-bucket = "travel-sample"
username = "<SELF_MANAGED_ADMIN>"
password = "<SELF_MANAGED_ADMIN_PWD>"
tls-cert-path = "/<ABSOLUTE_PATH_TO_SELF_MANAGED_ROOT_CERT>"
data-timeout = "2500ms"
connect-timeout = "7500ms"
query-timeout = "75s"

[[clusters]]
identifier = "Capella-Cluster"
hostnames = ["<COUCHBASE_CAPELLA_ENDPOINT>"]
default-bucket = "travel-sample"
username = "<CAPELLA_DATABASE_USER>"
```

```
password = "<CAPELLA_DATABASE_USER_PWD>"
tls-cert-path = "/<ABSOLUTE_PATH_TO_COUCHBASE_CAPELLA_ROOT_CERT>"
data-timeout = "2500ms"
connect-timeout = "7500ms"
query-timeout = "75s"
```

Sebelum Anda menyimpan file konfigurasi, gunakan tabel berikut untuk memastikan bahwa Anda menambahkan sumber dan informasi cluster target Anda sendiri.

<SELF_MANAGED_COUCHBASE_CLUSTER>	Gunakan alamat IP untuk cluster Couchbase Server yang dikelola sendiri.
<SELF_MANAGED_ADMIN>	Gunakan pengguna administrator untuk cluster Couchbase Server yang dikelola sendiri.
<ABSOLUTE_PATH_TO_SELF_MANAGED_ROOT_CERT>	Gunakan jalur absolut ke file sertifikat root yang disimpan untuk cluster Couchbase Server yang dikelola sendiri.
<COUCHBASE_CAPELLA_ENDPOINT>	Gunakan titik akhir koneksi untuk cluster Couchbase Capella Anda.
<CAPELLA_DATABASE_USER>	Gunakan pengguna database untuk cluster Couchbase Capella Anda.
<CAPELLA_DATABASE_USER_PWD>	Gunakan kata sandi pengguna database untuk cluster Couchbase Capella Anda.
<ABSOLUTE_PATH_TO_COUCHBASE_CAPELLA_ROOT_CERT>	Gunakan jalur absolut ke file sertifikat root yang disimpan untuk cluster Couchbase Capella Anda.

Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2

Neal Ardeljan dan Afroz Khan, Amazon Web Services

Ringkasan

Pola ini memandu Anda melalui langkah-langkah untuk bermigrasi dari Red Hat Enterprise Linux (RHEL) 6.9 lokal atau sistem yang lebih baru yang menjalankan IBM WebSphere Application Server (WAS) ke RHEL 8 yang menjalankan Apache Tomcat pada instans Amazon Elastic Compute Cloud (Amazon). EC2

Pola ini dapat diterapkan ke versi sumber dan target berikut:

- WebSphere Application Server 7.x ke Apache Tomcat 8 (dengan Java 7 atau lebih baru)
- WebSphere Application Server 8.x ke Apache Tomcat 8 (dengan Java 7 atau lebih baru)
- WebSphere Application Server 8.5.5.x ke Apache Tomcat 9 (dengan Java 8 atau lebih baru)
- WebSphere Application Server 8.5.5.x ke Apache Tomcat 10 (dengan Java 8 atau lebih baru)

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sumber kode Java, dengan asumsi berikut:
  - Menggunakan Java Development Kit (JDK) versi Java 7 atau yang lebih baru
  - Menggunakan kerangka Spring atau Apache Struts
  - Tidak menggunakan kerangka kerja Enterprise Java Beans (EJB) atau fungsionalitas WebSphere server lainnya yang tidak tersedia untuk Tomcat
  - Terutama menggunakan servlet atau Java Server Pages () JSPs
  - Menggunakan konektor Java Database Connectivity (JDBC) untuk terhubung ke database
- Sumber IBM WebSphere Application Server versi 7.x atau lebih tinggi
- Target Apache Tomcat versi 8.5 atau lebih tinggi

Arsitektur

Tumpukan teknologi sumber

- Aplikasi web yang dibangun menggunakan kerangka Apache Struts Model-View-Controller (MVC)
- Aplikasi web yang berjalan pada IBM WebSphere Application Server versi 7.x atau 8.x
- Aplikasi web yang menggunakan konektor Lightweight Directory Access Protocol (LDAP) untuk terhubung ke direktori LDAP (iPlanet/eTrust)
- Aplikasi yang menggunakan konektivitas IBM Tivoli Access Manager (TAM) untuk memperbarui kata sandi pengguna TAM (dalam implementasi saat ini, aplikasi menggunakan PD.jar)

Database lokal

- Oracle Database 21c (21.0.0.0)
- Oracle Database 19c (19.0.0.0)
- Oracle Database 12c Rilis 2 (12.2.0.1)
- Oracle Database 12c Rilis 1 (12.1.0.2)

Tumpukan teknologi target

- Apache Tomcat versi 8 (atau yang lebih baru) berjalan di RHEL pada sebuah instance EC2
- Amazon Relational Database Service (Amazon RDS) untuk Oracle

Untuk informasi selengkapnya tentang versi Oracle yang didukung oleh Amazon RDS, lihat situs web Amazon RDS for Oracle.

Arsitektur target

Alat

- Tingkat aplikasi: Membangun kembali aplikasi Java menjadi file WAR.
- Tingkat basis data: Pencadangan dan pemulihan asli Oracle.
- Alat migrasi Apache Tomcat untuk Jakarta EE. Alat ini mengambil aplikasi web yang ditulis untuk Java EE 8 yang berjalan pada Apache Tomcat 9 dan mengubahnya secara otomatis untuk berjalan di Apache Tomcat 10, yang mengimplementasikan Jakarta EE 9.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi penemuan aplikasi, jejak status saat ini, dan baseline kinerja.		BA, Pimpinan Migrasi
Validasi versi basis data sumber dan target.		DBA
Identifikasi persyaratan perangkat keras untuk EC2 instance server target.		DBA, SysAdmin
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, SysAdmin
Pilih jenis EC2 instans yang tepat berdasarkan kapasitas , fitur penyimpanan, dan fitur jaringan.		DBA, SysAdmin
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA, SysAdmin
Identifikasi strategi dan perkakas migrasi aplikasi.		DBA, Pimpinan Migrasi
Lengkapi desain migrasi dan panduan migrasi untuk aplikasi.		Bangun Pimpinan, Pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi runbook migrasi aplikasi.		Membangun Timbal, Cutover Lead, Testing Lead, Migration Lead

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).		SysAdmin
Buat grup keamanan.		SysAdmin
Konfigurasikan dan mulai Amazon RDS for Oracle.		DBA, SysAdmin

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat atau dapatkan akses ke titik akhir untuk mengambil file cadangan database.		DBA
Gunakan mesin database asli atau alat pihak ketiga untuk memigrasikan objek dan data database.	Untuk detailnya, lihat “Memigrasi objek dan data database” di bagian Informasi tambahan.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ajukan permintaan perubahan (CR) untuk migrasi.		Timbal Cutover
Dapatkan persetujuan CR untuk migrasi.		Timbal Cutover
Ikuti strategi migrasi aplikasi per runbook migrasi aplikasi.	Untuk detailnya, lihat “Menyiapkan tingkat aplikasi” di bagian Informasi tambahan.	DBA, Insinyur Migrasi, Pemilik Aplikasi
Tingkatkan aplikasi (jika perlu).		DBA, Insinyur Migrasi, Pemilik Aplikasi
Lengkapi tes fungsional, non-fungsional, validasi data, SLA, dan kinerja.		Menguji Lead, Pemilik Aplikasi, Pengguna Aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan tanda tangan dari pemilik aplikasi atau pemilik bisnis.		Timbal Cutover
Alihkan klien aplikasi ke infrastruktur baru.		DBA, Insinyur Migrasi, Pemilik Aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, Insinyur Migrasi, SysAdmin
Meninjau dan memvalidasi dokumen proyek.		Pimpinan Migrasi
Kumpulkan metrik seperti waktu untuk bermigrasi, persentase tugas manual versus otomatis, dan penghematan biaya.		Pimpinan Migrasi
Tutup proyek dan berikan umpan balik.		Pemimpin Migrasi, Pemilik Aplikasi

Sumber daya terkait

Referensi

- Dokumentasi Apache Tomcat 10.0
- Dokumentasi Apache Tomcat 9.0
- Dokumentasi Apache Tomcat 8.0
- Panduan instalasi Apache Tomcat 8.0
- Dokumentasi Apache Tomcat JNDI
- Situs web Amazon RDS for Oracle
- Harga Amazon RDS
- Oracle dan Amazon Web Services
- Oracle di Amazon RDS
- Penyebaran Multi-AZ Amazon RDS

Tutorial dan video

- Memulai dengan Amazon RDS

Informasi tambahan

Migrasi objek database dan data

Misalnya, jika Anda menggunakan backup/restore utilitas Oracle asli:

1. Buat cadangan Amazon Simple Storage Service (Amazon S3) untuk file cadangan database (opsional).
2. Cadangkan data Oracle DB ke folder bersama jaringan.
3. Masuk ke server pementasan migrasi untuk memetakan folder berbagi jaringan.
4. Salin data dari folder berbagi jaringan ke bucket S3.
5. Minta penerapan Amazon RDS Multi-AZ untuk Oracle.
6. Pulihkan cadangan database lokal ke Amazon RDS for Oracle.

Menyiapkan tingkat aplikasi

1. Instal Tomcat 8 (atau 9/10) dari situs web Apache Tomcat.
2. Package aplikasi dan library bersama ke dalam file WAR.
3. Menyebarkan file WAR di Tomcat.
4. Pantau log awal ke `Linux cat` pustaka bersama yang hilang dari WebSphere.
5. Tonton catatan awal ke ekstensi deskriptor penerapan WebSphere -spesifik `Linux cat` apa pun.
6. Kumpulkan pustaka Java dependen yang hilang dari WebSphere server.
7. Ubah elemen deskriptor penerapan WebSphere khusus -spesifik dengan padanan yang kompatibel dengan TomCat.
8. Bangun kembali file WAR dengan pustaka Java dependen dan deskriptor penerapan yang diperbarui.
9. Perbarui konfigurasi LDAP, konfigurasi database, dan koneksi uji (lihat Realm Configuration HOW-TO dan JNDI Datasource HOW-TO dalam dokumentasi Apache Tomcat).
10. Uji aplikasi yang diinstal terhadap database Amazon RDS for Oracle yang dipulihkan.
11. Buat Amazon Machine Image (AMI) untuk Linux dari EC2 instance.
12. Luncurkan arsitektur lengkap dengan grup Application Load Balancer dan Auto Scaling.

13. Perbarui URLs (dengan menggunakan persimpangan WebSeal) untuk menunjuk ke Application Load Balancer.
14. Perbarui database manajemen konfigurasi (CMDB).

Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2 dengan Auto Scaling

Kevin Yung dan Afroz Khan, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan aplikasi Java dari IBM WebSphere Application Server ke Apache Tomcat pada instans Amazon Elastic Compute Cloud (Amazon EC2) dengan Amazon Auto EC2 Scaling diaktifkan.

Dengan menggunakan pola ini, Anda dapat mencapai:

- Pengurangan biaya lisensi IBM
- Ketersediaan tinggi menggunakan penyebaran Multi-AZ
- Peningkatan ketahanan aplikasi dengan Amazon EC2 Auto Scaling

Prasyarat dan batasan

Prasyarat

- Aplikasi Java (versi 7. x atau 8. x) harus dikembangkan di tumpukan LAMP.
- Status target adalah untuk meng-host aplikasi Java di host Linux. Pola ini telah berhasil diterapkan di lingkungan Red Hat Enterprise Linux (RHEL) 7. Distribusi Linux lainnya dapat mengikuti pola ini, tetapi konfigurasi distribusi Apache Tomcat harus direferensikan.
- Anda harus memahami dependensi aplikasi Java.
- Anda harus memiliki akses ke kode sumber aplikasi Java untuk membuat perubahan.

Keterbatasan dan perubahan replatforming

- Anda harus memahami komponen arsip perusahaan (EAR) dan memverifikasi bahwa semua pustaka dikemas dalam file WAR komponen web. Anda perlu mengkonfigurasi Plugin Apache Maven WAR dan menghasilkan artefak file WAR.
- Saat menggunakan Apache Tomcat 8, ada konflik yang diketahui antara servlet-api.jar dan paket aplikasi built-in file jar. Untuk mengatasi masalah ini, hapus servlet-api.jar dari paket aplikasi.

- Anda harus mengkonfigurasi WEB-INF/sumber daya yang terletak di classpath konfigurasi Apache Tomcat. Secara default, pustaka JAR tidak dimuat dalam direktori. Atau, Anda dapat menyebarkan semua sumber daya di bawahsrc/main/resources.
- Periksa akar konteks hard-code dalam aplikasi Java, dan perbarui root konteks baru Apache Tomcat.
- Untuk mengatur opsi runtime JVM, Anda dapat membuat file konfigurasi setenv.sh di folder bin Apache Tomcat; misalnya, JAVA_OPTS, JAVA_HOME, dll.
- Otentikasi dikonfigurasi pada tingkat kontainer dan diatur sebagai ranah dalam konfigurasi Apache Tomcat. Otentikasi dibuat untuk salah satu dari tiga bidang berikut:
  - JDBC Database Realm mencari pengguna dalam database relasional yang diakses oleh driver JDBC.
  - DataSource Database Realm mencari pengguna dalam database yang diakses oleh JNDI.
  - JNDI Directory Realm mencari pengguna di direktori Lightweight Directory Access Protocol (LDAP) yang diakses oleh penyedia JNDI. Pencarian membutuhkan:
    - Detail koneksi LDAP: basis pencarian pengguna, filter pencarian, basis peran, filter peran
    - Kunci JNDI Directory Realm: Menghubungkan ke LDAP, mengautentikasi pengguna, dan mengambil semua grup di mana pengguna menjadi anggota
- Otorisasi: Dalam kasus wadah dengan otorisasi berbasis peran yang memeriksa batasan otorisasi di web.xml, sumber daya web harus didefinisikan dan dibandingkan dengan peran yang ditentukan dalam batasan. Jika LDAP tidak memiliki pemetaan peran grup, Anda harus menyetel atribut < security-role-ref > di web.xml untuk mencapai pemetaan peran grup. Untuk melihat contoh dokumen konfigurasi, lihat dokumentasi Oracle.
- Koneksi database: Buat definisi sumber daya di Apache Tomcat dengan URL titik akhir Amazon Relational Database Service (Amazon RDS) dan detail koneksi. Perbarui kode aplikasi untuk referensi DataSource dengan menggunakan pencarian JNDI. Koneksi DB yang ada yang didefinisikan tidak WebSphere akan berfungsi, karena menggunakan nama WebSphere JNDI. Anda dapat menambahkan <resource-ref>entri di web.xml dengan nama JNDI dan definisi DataSource tipe. Untuk melihat contoh dokumen konfigurasi, lihat dokumentasi Apache Tomcat.
- Logging: Secara default, Apache Tomcat log ke konsol atau ke file log. Anda dapat mengaktifkan penelusuran tingkat alam dengan memperbarui logging.properties (lihat Logging in Tomcat). Jika Anda menggunakan Apache Log4j untuk menambahkan log ke file, Anda harus mengunduh tomcat-Juli dan menambahkannya ke classpath.
- Manajemen sesi: Jika Anda mempertahankan IBM WebSeal untuk penyeimbangan beban aplikasi dan manajemen sesi, tidak diperlukan perubahan. Jika Anda menggunakan Application Load

Balancer atau Network Load Balancer di AWS untuk mengganti komponen IBM WebSeal, Anda harus menyiapkan manajemen sesi dengan menggunakan instans ElastiCache Amazon dengan klaster Memcached dan menyiapkan Apache Tomcat untuk menggunakan manajemen sesi sumber terbuka.

- Jika Anda menggunakan proxy forward IBM WebSeal, Anda harus menyiapkan Network Load Balancer baru di AWS. Gunakan yang IPs disediakan oleh Network Load Balancer untuk konfigurasi persimpangan WebSeal.
- Konfigurasi SSL: Kami menyarankan Anda menggunakan Secure Sockets Layer (SSL) untuk end-to-end komunikasi. Untuk mengatur konfigurasi server SSL di Apache Tomcat, ikuti petunjuk dalam dokumentasi Apache Tomcat.

Arsitektur

Tumpukan teknologi sumber

- Server WebSphere Aplikasi IBM

Tumpukan teknologi target

- Arsitekturnya menggunakan Elastic Load Balancing (versi 2). Jika Anda menggunakan IBM WebSeal untuk manajemen Identifikasi dan penyeimbangan beban, Anda dapat memilih Network Load Balancer di AWS untuk diintegrasikan dengan proxy balik IBM WebSeal.
- Aplikasi Java dikerahkan ke server aplikasi Apache Tomcat, yang berjalan pada EC2 instance dalam grup Amazon Auto Scaling. EC2 Anda dapat menyiapkan kebijakan penskalaan berdasarkan CloudWatch metrik Amazon seperti pemanfaatan CPU.
- Jika Anda menghentikan penggunaan IBM WebSeal untuk load balancing, Anda dapat menggunakan Amazon untuk Memcached ElastiCache untuk manajemen sesi.
- Untuk database back-end, Anda dapat menerapkan Ketersediaan Tinggi (Multi-AZ) untuk Amazon RDS dan memilih jenis mesin database.

Arsitektur target

Alat

- AWS CloudFormation
- AWS Command Line Interface (AWS CLI)
- Apache Tomcat (versi 7. x atau 8. x)
- RHEL 7 atau Centos 7
- Penyebaran Amazon RDS Multi-AZ
- Amazon ElastiCache untuk Memcached (opsional)

Epik

Siapkan VPC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).		
Buat subnet.		
Buat tabel routing jika perlu.		
Buat daftar kontrol akses jaringan (ACLs).		
Siapkan AWS Direct Connect atau koneksi VPN perusahaa n.		

Memplatform ulang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Refactor aplikasi membangun konfigurasi Maven untuk menghasilkan artefak WAR.		

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Refactor sumber data ketergantungan aplikasi di Apache Tomcat.		
Refactor kode sumber aplikasi untuk menggunakan nama JNDI di Apache Tomcat.		
Menyebarkan artefak WAR ke Apache Tomcat.		
Validasi dan tes aplikasi lengkap.		

Konfigurasikan jaringan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan firewall perusahaan untuk memungkinkan koneksi ke layanan ketergantungan.		
Konfigurasikan firewall perusahaan untuk memungkinkan akses pengguna akhir ke Elastic Load Balancing di AWS.		

Buat infrastruktur aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat dan menyebarkan aplikasi pada sebuah EC2 instance.		
Buat Amazon ElastiCache untuk klaster Memcached untuk manajemen sesi.		
Buat instans Amazon RDS Multi-AZ untuk database backend.		
Buat sertifikat SSL dan impor ke AWS Certificate Manager (ACM).		
Instal sertifikat SSL pada penyeimbang beban.		
Instal sertifikat SSL untuk server Apache Tomcat.		
Validasi dan tes aplikasi lengkap.		

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan infrastruktur yang ada.		
Kembalikan database dari produksi ke Amazon RDS.		

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Potong aplikasi dengan membuat perubahan DNS.		

Sumber daya terkait

Referensi

- Dokumentasi Apache Tomcat 7.0
- Panduan instalasi Apache Tomcat 7.0
- Dokumentasi Apache Tomcat JNDI
- Penyebaran Multi-AZ Amazon RDS
- Amazon ElastiCache untuk Memcached

Tutorial dan video

- Memulai dengan Amazon RDS

Migrasi aplikasi.NET dari Microsoft Azure App Service ke AWS Elastic Beanstalk

Raghavender Madamshitti, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan aplikasi web.NET yang dihosting di Microsoft Azure App Service ke AWS Elastic Beanstalk. Ada dua cara untuk memigrasikan aplikasi ke Elastic Beanstalk:

- Gunakan AWS Toolkit for Visual Studio - Plugin untuk Microsoft Visual Studio IDE ini menyediakan cara termudah dan paling mudah untuk menerapkan aplikasi.NET kustom ke AWS. Anda dapat menggunakan pendekatan ini untuk menyebarkan kode.NET langsung ke AWS dan untuk membuat sumber daya pendukung, seperti Amazon Relational Database Service (Amazon RDS) untuk database SQL Server, langsung dari Visual Studio.
- Unggah dan terapkan ke Elastic Beanstalk - Setiap Layanan Aplikasi Azure menyertakan layanan latar belakang yang disebut Kudu, yang berguna untuk menangkap dump memori dan log penyebaran, melihat parameter konfigurasi, dan mengakses paket penerapan. Anda dapat menggunakan konsol Kudu untuk mengakses konten Azure App Service, mengekstrak paket deployment, dan kemudian mengunggah paket ke Elastic Beanstalk dengan menggunakan opsi upload dan deploy di konsol Elastic Beanstalk.

Pola ini menjelaskan pendekatan kedua (mengunggah aplikasi Anda ke Elastic Beanstalk melalui Kudu). Pola ini juga menggunakan layanan AWS berikut: AWS Elastic Beanstalk, Amazon Virtual Private Cloud (Amazon VPC), Amazon, Amazon, Amazon Elastic Compute Cloud (CloudWatchAmazon) Auto Scaling, EC2 Amazon Simple Storage Service (Amazon S3), dan Amazon Route 53.

Aplikasi web.NET digunakan untuk AWS Elastic Beanstalk, yang berjalan di Amazon EC2 Auto Scaling Group. Anda dapat menyiapkan kebijakan penskalaan berdasarkan CloudWatch metrik Amazon seperti pemanfaatan CPU. Untuk database, Anda dapat menggunakan Amazon RDS di lingkungan multi-AZ, atau Amazon DynamoDB, tergantung pada aplikasi dan persyaratan bisnis Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Aplikasi web.NET yang berjalan di Azure App Service

- Izin untuk menggunakan konsol Azure App Service Kudu

Versi produk

- .NET Core (x64) 1.0.1, 2.0.0, atau yang lebih baru, atau .NET Framework 4.x, 3.5 (lihat .NET pada riwayat platform Windows Server)
- Internet Information Services (IIS) versi 8.0 atau yang lebih baru, berjalan pada Windows Server 2012 atau yang lebih baru
- .NET 2.0 atau 4.0 Runtime.

Arsitektur

Tumpukan teknologi sumber

- Aplikasi yang dikembangkan menggunakan .NET Framework 3.5 atau yang lebih baru, atau.NET Core 1.0.1, 2.0.0, atau yang lebih baru, dan di-host di Azure App Service (aplikasi web atau aplikasi API)

Tumpukan teknologi target

- AWS Elastic Beanstalk berjalan di grup Amazon EC2 Auto Scaling

Arsitektur migrasi

Alur kerja penerapan

Alat

Alat

- .NET Core atau .NET Framework
- C#
- IIS
- Konsol Kudu

Layanan dan fitur AWS

- AWS Elastic Beanstalk — Elastic Beanstalk easy-to-use adalah layanan untuk menyebarkan dan menskalakan aplikasi web.NET. Elastic Beanstalk secara otomatis mengelola penyediaan kapasitas, load balancing, dan auto scaling.
- Grup EC2 Auto Scaling Amazon — Elastic Beanstalk menyertakan grup Auto Scaling yang mengelola instans Amazon di lingkungan. EC2 Dalam lingkungan instans tunggal, grup Auto Scaling memastikan bahwa selalu ada satu instans yang berjalan. Di lingkungan load-balanced, Anda dapat mengonfigurasi grup dengan berbagai instance untuk dijalankan, dan Amazon EC2 Auto Scaling menambahkan atau menghapus instance sesuai kebutuhan, berdasarkan pemuatan.
- Elastic Load Balancing — Saat Anda mengaktifkan load balancing di AWS Elastic Beanstalk, Elastic Load Balancing akan menciptakan penyeimbang beban yang mendistribusikan lalu lintas antar instans di lingkungan. EC2
- Amazon CloudWatch - Elastic Beanstalk secara otomatis CloudWatch menggunakan Amazon untuk memberikan informasi tentang sumber daya aplikasi dan lingkungan Anda. Amazon CloudWatch mendukung metrik standar, metrik khusus, dan alarm.
- Amazon Route 53 - Amazon Route 53 adalah layanan web Sistem Nama Domain (DNS) cloud yang sangat tersedia dan dapat diskalakan. Anda dapat menggunakan catatan alias Route 53 untuk memetakan nama domain khusus ke lingkungan AWS Elastic Beanstalk.

Epik

Menyiapkan VPC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan cloud pribadi virtual (VPC).	Di akun AWS Anda, buat VPC dengan informasi yang diperlukan.	Administrator sistem
Buat subnet.	Buat dua atau lebih subnet di VPC Anda.	Administrator sistem
Buat tabel rute.	Buat tabel rute, berdasarkan kebutuhan Anda.	Administrator sistem

Mengatur Elastic Beanstalk

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses konsol Azure App Service Kudu.	Akses Kudu melalui portal Azure dengan menavigasi ke dasbor App Service, lalu pilih Advanced Tools, Go. Atau, Anda dapat memodifikasi URL Layanan Aplikasi Azure sebagai berikut:`https://<appservicename>.scm.azurewebsites.net` .	Pengembang aplikasi, Administrator sistem
Unduh paket penyebaran dari Kudu.	Arahkan ke Windows PowerShell dengan memilih DebugConsoleopsi. Ini akan membuka konsol Kudo. Buka `wwwroot` folder dan unduh. Ini akan mengunduh paket penyebaran Layanan Aplikasi Azure sebagai file zip. Sebagai contoh, lihat lampirannya.	Pengembang aplikasi, Administrator sistem
Buat paket untuk Elastic Beanstalk.	Buka zip paket penerapan yang Anda unduh dari Azure App Service. Buat file JSON bernama `aws-windows-deployment-manifest.json` (file ini hanya diperlukan untuk aplikasi. NET Core). Buat file zip yang menyertakan `aws-windows-deployment-manifest.json` dan file paket penerapan Layanan Aplikasi	Pengembang aplikasi, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Azure. Sebagai contoh, lihat lampirannya.	
Buat aplikasi Elastic Beanstalk baru.	Buka Konsol Elastic Beanstalk . Pilih aplikasi yang sudah ada atau buat aplikasi baru.	Pengembang aplikasi, Administrator sistem
Buat lingkungan.	Di menu Tindakan konsol Elastic Beanstalk, pilih Buat lingkungan. Pilih lingkungan server web dan platform.NET/ IIS. Untuk kode aplikasi, pilih Unggah. Unggah file zip yang Anda siapkan untuk Elastic Beanstalk, lalu pilih Create Environment.	Pengembang aplikasi, Administrator sistem
Konfigurasikan Amazon CloudWatch.	Secara default, CloudWatch pemantauan dasar diaktifka n. Jika Anda ingin mengubah konfigurasi, di wizard Elastic Beanstalk, pilih aplikasi yang diterbitkan, lalu pilih Monitorin g.	Administrator sistem
Verifikasi bahwa paket penerapan ada di Amazon S3.	Ketika lingkungan aplikasi telah dibuat, Anda dapat menemukan paket penerapan di bucket S3.	Pengembang aplikasi, Administrator sistem
Uji aplikasinya.	Ketika lingkungan telah dibuat, gunakan URL yang disediaka n di konsol Elastic Beanstalk untuk menguji aplikasi.	Administrator sistem

Sumber daya terkait

- Konsep AWS Elastic Beanstack (dokumentasi Elastic Beanstalk)
- Memulai dengan.NET pada Elastic Beanstalk (dokumentasi Elastic Beanstalk)
- Konsol Kudu () GitHub
- Menggunakan “Kudu” untuk Mengelola Aplikasi Web Azure (artikel GS Lab)
- Penerapan Batang Kacang Elastis ASP.NET Core Kustom (Panduan Pengguna AWS Toolkit for Visual Studio)
- Dokumentasi Elastic Load Balancing
- Platform yang Didukung AWS Elastic Beanstalk (Dokumentasi Elastic Beanstalk)
- Menerapkan Aplikasi Web ke AWS (artikel C# Corner)
- Menskalakan Ukuran Grup Auto Scaling Anda (dokumentasi Amazon EC2)
- Ketersediaan Tinggi (Multi-AZ) untuk Amazon RDS (dokumentasi Amazon RDS)

Informasi tambahan

Catatan

- Jika Anda memigrasikan database lokal atau Azure SQL Server ke Amazon RDS, Anda juga harus memperbarui detail koneksi database.
- Untuk tujuan pengujian, contoh aplikasi demo dilampirkan.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Bermigrasi dari Oracle WebLogic ke Apache Tomcat (ToMee) di Amazon ECS

Anya Epishcheva dan Harshad Gohil, Amazon Web Services

Ringkasan

Pola ini membahas langkah-langkah untuk memigrasikan sistem Oracle Solaris SPARC lokal yang menjalankan Oracle WebLogic ke instalasi berbasis kontainer Docker yang menjalankan Apache ToMee (Apache Tomcat dengan dukungan kontainer tambahan) dengan Amazon Elastic Container Service (Amazon ECS).

Untuk informasi tentang migrasi database yang terkait dengan aplikasi yang Anda migrasi dari Oracle WebLogic ke Tomcat, lihat pola migrasi database dalam katalog ini.

Praktik terbaik

Langkah-langkah untuk memigrasi aplikasi web Java dan Java Enterprise Edition (Java EE) bervariasi, tergantung pada jumlah sumber daya khusus kontainer yang digunakan oleh aplikasi. Aplikasi berbasis pegas biasanya lebih mudah untuk dimigrasi, karena mereka memiliki sejumlah kecil dependensi pada wadah penerapan. Sebaliknya, aplikasi Java EE yang menggunakan Enterprise JavaBeans (EJBs) dan sumber daya kontainer terkelola seperti kumpulan thread, Java Authentication and Authorization Service (JAAS), dan Container-Managed Persistence (CMP) memerlukan lebih banyak usaha.

Aplikasi yang dikembangkan untuk Oracle Application Server sering menggunakan suite Oracle Identity Management. Pelanggan yang bermigrasi ke server aplikasi open-source sering memilih untuk mengimplementasikan kembali identitas dan manajemen akses menggunakan federasi berbasis SAML. Orang lain menggunakan Oracle HTTP Server Webgate untuk kasus ketika bermigrasi dari Oracle Identity Management suite bukanlah pilihan.

Aplikasi web Java dan Java EE adalah kandidat yang bagus untuk penerapan pada layanan AWS yang berbasis Docker, seperti AWS Fargate dan Amazon ECS. Pelanggan sering memilih gambar Docker dengan versi terbaru dari server aplikasi target (seperti ToMee) dan Java Development Kit (JDK) yang sudah diinstal sebelumnya. Mereka menginstal aplikasi mereka di atas image Docker dasar, mempublikasikannya di registri Amazon Elastic Container Registry (Amazon ECR), dan menggunakannya untuk penerapan aplikasi mereka yang dapat diskalakan di AWS Fargate atau Amazon ECS.

Idealnya, penerapan aplikasi bersifat elastis; yaitu, jumlah instance aplikasi masuk atau keluar, tergantung pada lalu lintas atau beban kerja. Ini berarti bahwa instance aplikasi harus online atau dihentikan untuk menyesuaikan kapasitas permintaan.

Saat memindahkan aplikasi Java ke AWS, pertimbangkan untuk membuatnya tanpa kewarganegaraan. Ini adalah prinsip arsitektur utama AWS Well-Architected Framework yang akan memungkinkan penskalaan horizontal menggunakan kontainerisasi. Misalnya, sebagian besar aplikasi web berbasis Java menyimpan informasi sesi pengguna secara lokal. Untuk bertahan dari penghentian instans aplikasi karena penskalaan otomatis di Amazon Elastic Compute Cloud EC2 (Amazon) atau karena alasan lain, informasi sesi pengguna harus disimpan secara global sehingga pengguna aplikasi web dapat terus bekerja dengan lancar dan transparan tanpa menghubungkan kembali atau masuk kembali ke aplikasi web. Ada beberapa opsi arsitektur untuk pendekatan ini, termasuk Amazon ElastiCache untuk Redis, atau menyimpan status sesi dalam database global. Server aplikasi seperti ToMee memiliki plugin, yang memungkinkan penyimpanan dan manajemen sesi melalui Redis, database, dan penyimpanan data global lainnya.

Gunakan alat logging dan debugging yang umum dan terpusat yang mudah diintegrasikan dengan Amazon CloudWatch dan AWS X-Ray. Migrasi memberikan kesempatan untuk meningkatkan kemampuan siklus hidup aplikasi. Misalnya, Anda mungkin ingin mengotomatiskan proses build sehingga perubahan mudah dilakukan menggunakan pipeline continuous integration dan continuous delivery (CI/CD). Ini mungkin memerlukan perubahan pada aplikasi sehingga dapat digunakan tanpa downtime.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Sumber kode Java dan JDK
- Aplikasi sumber yang dibangun dengan Oracle WebLogic
- Solusi yang ditetapkan untuk manajemen identitas dan akses (SAMP atau Oracle Webgate)
- Solusi yang ditetapkan untuk manajemen sesi aplikasi (pindah like-for-like atau dengan Amazon ElastiCache, atau membuat aplikasi tanpa kewarganegaraan jika diperlukan)
- Memahami apakah tim perlu memfaktorkan ulang pustaka khusus J2EE untuk portabilitas ke Apache ToMee (lihat Status Implementasi Java EE 7 di situs web Apache)
- Gambar ToMee yang diperkeras berdasarkan persyaratan keamanan Anda
- Gambar kontainer dengan ToMee target yang sudah diinstal sebelumnya
- Remediasi aplikasi disetujui dan diimplementasikan jika diperlukan (misalnya, logging debug build, otentikasi)

Versi produk

- Oracle WebLogic OC4 J, 9i, 10g
- Tomcat 7 (dengan Java 1.6 atau lebih baru)

Arsitektur

Tumpukan teknologi sumber

- Aplikasi web yang dibangun menggunakan Oracle WebLogic
- Aplikasi web menggunakan otentikasi Oracle Webgate atau SAMP
- Aplikasi web terhubung ke Oracle Database versi 10g dan yang lebih baru

Tumpukan teknologi target

- ToMee (Apache Tomcat dengan dukungan kontainer tambahan) berjalan di Amazon ECS (lihat juga Menyebarkan Aplikasi Web Java dan Java Microservices di Amazon ECS)
- Amazon Relational Database Service (Amazon RDS) untuk Oracle; untuk versi Oracle yang didukung oleh Amazon RDS, lihat Amazon RDS for Oracle

Arsitektur target

Alat

Untuk beroperasi di ToMee, aplikasi Java harus dibangun kembali menjadi file.war. Dalam beberapa kasus, perubahan aplikasi mungkin diperlukan untuk mengoperasikan aplikasi di ToMee; Anda harus memeriksa untuk memastikan bahwa opsi konfigurasi yang diperlukan dan properti lingkungan didefinisikan dengan benar.

Selain itu, pencarian Java Naming and Directory Interface (JNDI) dan ruang nama JavaServer Pages (JSP) harus didefinisikan dengan benar. Pertimbangkan untuk memeriksa nama file yang digunakan oleh aplikasi untuk menghindari tabrakan penamaan dengan pustaka T bawaan. Misalnya, persistence.xml adalah nama file yang digunakan oleh kerangka Apache OpenJPA (yang dibundel dengan OpenEJB di ToMe) untuk tujuan konfigurasi. File persistence.xml di PUI berisi deklarasi kacang kerangka Spring.

ToMee versi 7.0.3 dan yang lebih baru (Tomcat 8.5.7 dan yang lebih baru) mengembalikan respons HTTP 400 (permintaan buruk) untuk mentah (tidak dikodekan) dengan karakter khusus. URLs Respons server muncul sebagai halaman kosong untuk pengguna akhir. Versi ToMee dan Tomcat sebelumnya memungkinkan penggunaan karakter khusus tertentu yang tidak dikodekan URLs; Namun, itu dianggap tidak aman, seperti yang dinyatakan di situs web CVE-2016-6816. Untuk mengatasi masalah pengkodean URL, URLs diteruskan ke browser secara langsung melalui JavaScript harus dikodekan dengan metode encodeUri () alih-alih digunakan sebagai string mentah.

Setelah Anda menyebarkan file.war di ToMee, pantau log mulai ke Linux cat untuk setiap pustaka bersama yang hilang dan ekstensi khusus Oracle untuk menambahkan komponen yang hilang dari pustaka Tomcat.

Prosedur umum

- Konfigurasikan aplikasi di ToMee.
- Mengidentifikasi dan mengkonfigurasi ulang file konfigurasi khusus server aplikasi dan sumber daya dari sumber ke format target.
- Identifikasi dan konfigurasi ulang sumber daya JNDI.
- Sesuaikan namespace dan pencarian EJB ke format yang diperlukan oleh server aplikasi target (jika ada).
- Konfigurasikan ulang peran keamanan khusus wadah aplikasi JAAS dan pemetaan prinsip (jika ada).
- Package aplikasi dan library bersama ke dalam file.war.
- Terapkan file.war di ToMee dengan menggunakan wadah Docker yang disediakan.
- Pantau log awal untuk mengidentifikasi pustaka bersama yang hilang dan ekstensi deskriptor penerapan. Jika ada yang ditemukan, kembali ke tugas pertama.
- Uji aplikasi yang diinstal terhadap database Amazon RDS yang dipulihkan.
- Luncurkan arsitektur lengkap dengan load balancer dan Amazon ECS cluster dengan mengikuti petunjuk di Deploy Docker Containers.
- Perbarui URLs untuk menunjuk ke penyeimbang beban.
- Perbarui database manajemen konfigurasi (CMDB).

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan penemuan aplikasi (jejak keadaan saat ini dan garis dasar kinerja).		BA, Pimpinan Migrasi
Validasi versi dan mesin basis data sumber dan target.		DBA
Validasi desain aplikasi sumber dan target (identitas dan manajemen sesi).		DBA, Insinyur Migrasi, Pemilik Aplikasi
Identifikasi persyarat an perangkat keras dan penyimpanan untuk instance server target.		DBA, SysAdmin
Pilih jenis instans yang tepat berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.		DBA, SysAdmin
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA, SysAdmin
Identifikasi strategi dan perkakas migrasi aplikasi.		DBA, Pimpinan Migrasi
Lengkapi desain migrasi dan panduan migrasi untuk aplikasi.		Bangun Pimpinan, Pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi runbook migrasi aplikasi.		Membangun Timbal, Cutover Lead, Testing Lead, Migration Lead

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).		SysAdmin
Buat grup keamanan.		SysAdmin
Konfigurasikan dan mulai instans Amazon RDS DB.		DBA, SysAdmin
Konfigurasikan penerapan Amazon ECS.		SysAdmin
Package aplikasi Anda sebagai image Docker.		SysAdmin
Dorong gambar ke registri Amazon ECR (atau lewati langkah ini dan dorong ke cluster Amazon ECS).		SysAdmin
Konfigurasikan definisi tugas untuk aplikasi dan opsi layanan Amazon ECS.		SysAdmin
Konfigurasikan klaster Anda, tinjau setelan keamanan, dan setel peran AWS Identity and Access Management (IAM).		SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan penyiapan Anda dan jalankan pengujian sesuai dengan runbook migrasi aplikasi Anda.		SysAdmin

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan izin tim jaminan keamanan Anda untuk memindahkan data produksi ke AWS.		DBA, Insinyur Migrasi, Pemilik Aplikasi
Buat dan dapatkan akses ke titik akhir untuk mengambil file cadangan database.		DBA
Gunakan mesin database asli atau alat pihak ketiga untuk memigrasikan objek dan data database.		DBA
Jalankan pengujian yang diperlukan dari runbook migrasi aplikasi untuk mengonfirmasi migrasi data yang berhasil.		DBA, Insinyur Migrasi, Pemilik Aplikasi

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat permintaan perubahan (CR) untuk migrasi.		Timbal Cutover
Dapatkan persetujuan CR untuk migrasi.		Timbal Cutover
Ikuti strategi migrasi aplikasi dari runbook migrasi aplikasi.		DBA, Insinyur Migrasi, Pemilik Aplikasi
Tingkatkan aplikasi (jika diperlukan).		DBA, Insinyur Migrasi, Pemilik Aplikasi
Fungsional lengkap, non-fungsional, validasi data, SLA, dan tes kinerja.		Menguji Lead, Pemilik Aplikasi, Pengguna Aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan tanda tangan dari aplikasi atau pemilik bisnis.		Timbal Cutover
Jalankan latihan topik tabel untuk berjalan melalui semua langkah runbook cutover.		DBA, Insinyur Migrasi, Pemilik Aplikasi
Alihkan klien aplikasi ke infrastruktur baru.		DBA, Insinyur Migrasi, Pemilik Aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		DBA, Insinyur Migrasi, SysAdmin
Meninjau dan memvalidasi dokumen proyek.		Pimpinan Migrasi
Kumpulkan metrik di sekitar waktu untuk bermigrasi,% manual vs. alat, penghematan biaya, dll.		Pimpinan Migrasi
Tutup proyek dan berikan umpan balik.		Pemimpin Migrasi, Pemilik Aplikasi

Sumber daya terkait

Referensi

- Dokumentasi Apache Tomcat 7.0
- Panduan instalasi Apache Tomcat 7.0
- Dokumentasi Apache Tomcat JNDI
- Dokumentasi Apache ToMee
- Amazon RDS for Oracle
- Harga Amazon RDS
- Oracle dan AWS
- Oracle pada dokumentasi Amazon RDS
- Penerapan Amazon RDS Multi-AZ
- Memulai dengan Amazon ECS
- Memulai dengan Amazon RDS

Tutorial dan video

- Praktik Terbaik untuk Menjalankan Database Oracle di Amazon RDS (re:Invent 2018 presentasi)

Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS for Oracle menggunakan AWS DMS

Chethan Gangadharaiah dan Brian motzer, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah untuk memigrasikan database Oracle di Amazon Elastic Compute Cloud (Amazon EC2) ke Amazon Relational Database Service (Amazon RDS) untuk Oracle menggunakan AWS Database Migration Service (AWS DMS). Pola ini juga menggunakan Oracle SQL Developer atau SQL*Plus untuk terhubung ke instans Oracle DB Anda, dan menyertakan CloudFormation template AWS yang mengotomatiskan beberapa tugas.

Migrasi ke Amazon RDS for Oracle memungkinkan Anda untuk fokus pada bisnis dan aplikasi Anda sementara Amazon RDS menangani tugas-tugas administrasi database seperti penyediaan database, backup dan recovery, patch keamanan, upgrade versi, dan manajemen penyimpanan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Gambar Mesin Amazon (AMI) untuk Oracle Database di Amazon EC2

Versi produk

- AWS DMS mendukung Oracle versi 11g (versi 11.2.0.3.v1 dan yang lebih baru), 12c, dan 18c untuk database instans Amazon RDS untuk edisi Enterprise, Standard, Standard One, dan Standard Two. Untuk informasi terbaru tentang versi yang didukung, lihat Menggunakan Database Oracle sebagai Target untuk AWS DMS dalam dokumentasi AWS. (CloudFormation Template AWS terlampir menggunakan Oracle versi 12c sebagai database sumber.)
- Pengembang Oracle SQL 4.0.3

Arsitektur

Arsitektur sumber

- Database Oracle di Amazon EC2

Arsitektur target

- Amazon RDS for Oracle

Arsitektur migrasi

Alat

- AWS DMS — AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan database ke AWS dengan cepat dan aman. Ini mendukung migrasi homogen dan heterogen. Untuk informasi tentang versi dan edisi database Oracle yang didukung, lihat Menggunakan Database Oracle sebagai Sumber untuk AWS DMS dan Menggunakan Database Oracle sebagai Target untuk AWS DMS dalam dokumentasi AWS.
- Oracle SQL Developer atau SQL* Plus — Alat ini memungkinkan Anda untuk terhubung ke Amazon RDS for Oracle DB instans.

Epik

Siapkan basis data target Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon RDS for Oracle DB.	Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/. Buat instans Oracle DB dengan memilih mesin yang sesuai, template, pengaturan kredenal database, jenis instans, penyimpanan, pengaturan Multi-AZ, virtual private cloud (VPC) dan konfigurasi, kredensi login, dan pengaturan tambahan untuk database Oracle. Untuk	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	petunjuk, lihat tautan di bagian “Sumber daya terkait”. Atau gunakan CloudFormation template AWS (create_rds.yaml) di lampiran untuk membuat instans Amazon RDS for Oracle DB.	
Connect ke Amazon RDS dan berikan hak istimewa kepada pengguna Oracle.	Ubah grup keamanan untuk membuka port yang sesuai untuk terhubung dari mesin lokal dan instans replikasi AWS DMS. Saat Anda mengonfigurasi konektivitas, pastikan opsi “Dapat diakses secara publik” dipilih sehingga Anda dapat terhubung ke database dari luar VPC. Connect ke Amazon RDS dengan Oracle SQL Developer atau SQL *Plus dengan menggunakan kredenal login, buat pengguna AWS DMS, dan berikan hak istimewa yang diperlukan kepada pengguna AWS DMS untuk memodifikasi database.	Developer

Konfigurasikan grup keamanan EC2 instance sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa apakah database Oracle aktif dan berjalan.	Gunakan Secure Shell (SSH) untuk terhubung ke EC2 instance, dan coba	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sambungkan ke database Oracle dengan menggunakan SQL *Plus.	
Ubah grup keamanan.	Ubah grup keamanan EC2 instans untuk membuka port yang sesuai, sehingga Anda dapat terhubung dari mesin lokal dan instans replikasi AWS DMS.	Developer

Mengatur AWS DMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans replikasi AWS DMS.	Di AWS DMS, buat instance replikasi di VPC yang sama dengan instans Amazon RDS for Oracle DB Anda. Tentukan nama dan deskripsi untuk instance replikasi, pilih kelas instans dan versi mesin replikasi (gunakan default), pilih VPC tempat Anda membuat instans Amazon RDS DB, atur pengatura n Multi-AZ jika diperluka n, alokasikan penyimpan an, tentukan Availability Zone, dan konfigurasikan pengaturan tambahan. Atau, Anda dapat menggunakan CloudFormation template AWS (DMS.yaml) di lampiran	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk mengimplementasikan langkah ini.	
Connect ke titik akhir basis data sumber dan target.	Buat titik akhir basis data sumber dan target dengan menentukan pengenal titik akhir, mesin, server, port, kredensi login, dan atribut koneksi tambahan. Untuk server sumber, gunakan DNS publik dari EC2 instance yang menghosting database Oracle. Untuk server target, gunakan endpoint Amazon RDS for Oracle. Jalankan tes untuk memverifikasi bahwa koneksi sumber dan target berfungsi. Atau, Anda dapat menggunakan CloudFormation template AWS (DMS.yaml) di lampiran untuk mengimplementasikan langkah ini.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas AWS DMS.	Buat tugas AWS DMS untuk memigrasikan data dari titik akhir sumber ke titik akhir target, untuk menyiapkan replikasi antara titik akhir sumber dan tujuan, atau keduanya. Saat membuat tugas AWS DMS, tentukan instance replikasi, titik akhir sumber, titik akhir target, jenis migrasi (hanya data, replikasi saja, atau keduanya) , pemetaan tabel, dan filter. Jalankan tugas AWS DMS, pantau tugas, periksa statistik tabel, dan periksa log di Amazon CloudWatch. Atau, Anda dapat menggunakan CloudFormation template AWS (DMS.yaml) di lampiran untuk mengimplementasikan langkah ini.	DBA

Sumber daya terkait

- Membuat instans Amazon RDS DB
- Menghubungkan ke Instans DB Menjalankan Oracle Database Engine
- Dokumentasi AWS DMS
- Panduan AWS DMS Step-by-Step
- Migrasi Database Oracle ke AWS Cloud

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Memigrasi database Oracle lokal ke Amazon OpenSearch Service menggunakan Logstash

Aditya Goteti, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memindahkan data dari database Oracle lokal ke Amazon OpenSearch Service menggunakan Logstash. Ini mencakup pertimbangan arsitektur dan beberapa keahlian dan rekomendasi yang diperlukan. Data dapat dari satu tabel atau dari beberapa tabel di mana pencarian teks lengkap perlu dilakukan.

OpenSearch Layanan dapat dikonfigurasi dalam virtual private cloud (VPC), atau dapat ditempatkan secara publik dengan pembatasan berbasis IP. Pola ini menjelaskan skenario di mana OpenSearch Layanan dikonfigurasi dalam VPC. Logstash digunakan untuk mengumpulkan data dari database Oracle, menguraikannya ke format JSON, dan kemudian memasukkan data ke dalam Layanan. OpenSearch

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Java 8 (diperlukan oleh Logstash 6.4.3)
- Konektivitas antara server database lokal dan instans Amazon Elastic Compute Cloud EC2 (Amazon) dalam VPC, dibuat menggunakan AWS Virtual Private Network (AWS VPN)
- Sebuah query untuk mengambil data yang diperlukan untuk didorong ke OpenSearch Service dari database
- Driver Konektivitas Database Oracle Java (JDBC)

Batasan

- Logstash tidak dapat mengidentifikasi catatan yang dihapus dari database

Versi produk

- Oracle Database 12c
- OpenSearch Layanan 6,3

- Logstash 6.4.3

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal
- AWS VPN lokal

Tumpukan teknologi target

- VPC
- EC2 contoh
- OpenSearch Layanan
- Logstash
- NAT Gateway (untuk pembaruan sistem operasi pada EC2 instance dan untuk menginstal Java 8, Logstash, dan plugin)

Arsitektur migrasi data

Alat

- Logstash 6.4.3
- Plugin masukan JDBC (unduh dan informasi lebih lanjut)
- Plugin keluaran Logstash (logstash-output-amazon_es)
- Oracle JDBC driver

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi ukuran data sumber.	Ukuran data sumber adalah salah satu parameter	DBA, Pengembang basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang Anda gunakan untuk menentukan jumlah pecahan yang akan dikonfigurasi dalam indeks.	
Menganalisis tipe data dari setiap kolom dan data yang sesuai.	OpenSearch Layanan secara dinamis memetakan tipe data ketika bidang yang sebelumnya tidak terlihat ditemukan dalam dokumen. Jika ada tipe atau format data tertentu (misalnya, bidang tanggal) yang perlu dideklarasikan secara eksplisit, identifikasi bidang dan tentukan pemetaan untuk bidang tersebut selama pembuatan indeks.	Pemilik aplikasi, Pengembang, Pengembang Basis Data
Tentukan apakah ada kolom dengan kunci primer atau unik.	Untuk menghindari duplikasi catatan di OpenSearch Layanan Amazon selama pembaruan atau penyisipan, Anda perlu mengkonfigurasi `document_id` pengaturan di bagian output `amazon_es` plugin (misalnya, `document_id => "%{customer_id}"` di mana `customer_id` kunci utama).	Pemilik aplikasi, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Analisis jumlah dan frekuensi catatan baru yang ditambahkan; periksa seberapa sering catatan dihapus.	Tugas ini diperlukan untuk memahami tingkat pertumbuhan data sumber. Jika data dibaca secara intensif dan penyisipan jarang terjadi, Anda dapat memiliki satu indeks. Jika catatan baru sering dimasukkan dan tidak ada penghapusan, ukuran pecahan dapat dengan mudah melebihi ukuran maksimum yang disarankan 50 GB. Dalam hal ini, Anda dapat secara dinamis membuat indeks dengan mengkonfigurasi pola indeks di Logstash dan dalam kode di mana Anda dapat mengaksesnya dengan menggunakan alias.	Pemilik aplikasi, Pengembang
Tentukan berapa banyak replika yang dibutuhkan.		Pemilik aplikasi, Pengembang
Tentukan jumlah pecahan yang akan dikonfigurasi pada indeks.		Pemilik aplikasi, Pengembang
Identifikasi jenis instance untuk node master khusus, node data, dan EC2 instance.	Untuk informasi selengkapnya, lihat bagian Sumber daya terkait.	Pemilik aplikasi, Pengembang
Tentukan jumlah node master khusus dan node data yang diperlukan.	Untuk informasi selengkapnya, lihat bagian Sumber daya terkait.	

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan sebuah EC2 instance.	Luncurkan EC2 instance dalam VPC yang terhubung dengan AWS VPN.	Konstruksi Amazon VPC, AWS VPN
Instal Logstash pada instance. EC2		Developer
Instal plugin Logstash.	Instal plugin `jdbc-input` Logstash yang diperlukan dan. `logstash-output-amazon_es`	Developer
Konfigurasikan Logstash.	Buat keystore Logstash untuk menyimpan informasi sensitif seperti kunci AWS Secrets Manager dan kredensyal database, lalu tempatkan referensi dalam file konfigurasi Logstash.	Developer
Konfigurasikan antrian huruf mati dan antrian persisten.	Secara default, ketika Logstash menemukan peristiwa yang tidak dapat diproses karena data berisi kesalahan pemetaan atau masalah lainnya, pipeline Logstash akan menggantung atau menghentikan peristiwa yang gagal. Untuk melindungi dari kehilangan data dalam situasi ini, Anda dapat mengonfigurasi Logstash untuk menulis peristiwa yang gagal ke	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	antrian huruf mati alih-alih menjatuhkannya. Untuk melindungi dari kehilanga n data selama penghentian abnormal, Logstash memiliki fitur antrian persisten yang akan menyimpan antrian pesan pada disk. Antrian persisten memberikan daya tahan data di Logstash.	
Buat domain OpenSearch Layanan Amazon.	Buat domain Amazon OpenSearch Service dengan kebijakan akses yang tidak memerlukan permintaan penandatanganan dengan kredensyal AWS Identity and Access Management (IAM). Domain OpenSearch Layanan Amazon harus dibuat dalam VPC yang sama. Anda juga harus memilih jenis instance dan mengatur jumlah node khusus dan master berdasark an analisis Anda.	Developer
Konfigurasikan log OpenSearc h Layanan Amazon yang diperlukan.	Untuk informasi selengkapnya, lihat dokumentasi OpenSearch Layanan.	
Buat indeks.		Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai Logstash.	Jalankan Logstash sebagai layanan latar belakang. Logstash menjalankan kueri SQL yang dikonfigurasi, menarik data, mengubahn ya menjadi format JSON, dan mengumpankannya ke Layanan. OpenSearc h Untuk pemuatan awal, jangan konfigurasikan penjadwal dalam file konfigura si Logstash.	Developer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa dokumen.	Periksa jumlah dokumen pada indeks dan apakah semua dokumen ada dalam database sumber. Selama pemuatan awal, mereka ditambahkan ke indeks dan digunakan untuk menghentikan Logstash. Ubah konfigurasi Logstash untuk menambahkan penjadwal yang berjalan pada interval tetap tergantun g pada kebutuhan klien, dan mulai ulang Logstash. Logstash hanya akan memilih catatan yang diperbarui atau ditambahkan setelah proses terakhir, dan stempel waktu proses terakhir disimpan dalam file yang dikonfigurasi dengan properti `last_run_ metadata_path => "/usr/share/logsta sh/.logstash_jdbc_ last_run"` dalam file konfigurasi Logstash.	Developer

Sumber daya terkait

- CloudWatch Alarm yang Direkomendasikan
- Node Master OpenSearch Layanan Amazon Khusus
- Ukuran Domain OpenSearch Layanan Amazon
- Dokumentasi Logstash
- Plugin masukan JDBC

- Plugin keluaran Logstash
- Situs web Amazon OpenSearch Service

Memigrasikan database Oracle lokal ke Amazon RDS for Oracle

Baji Shaik dan Pavan Pusuluri, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah untuk memigrasikan database Oracle lokal ke Amazon Relational Database Service (Amazon RDS) untuk Oracle. Sebagai bagian dari proses migrasi, Anda membuat rencana migrasi dan mempertimbangkan faktor-faktor penting tentang infrastruktur basis data target Anda berdasarkan basis data sumber Anda. Anda dapat memilih salah satu dari dua opsi migrasi berdasarkan persyaratan bisnis dan kasus penggunaan:

- AWS Database Migration Service (AWS DMS) — Anda dapat menggunakan AWS DMS untuk memigrasikan database ke AWS Cloud dengan cepat dan aman. Database sumber Anda tetap beroperasi penuh selama migrasi, yang meminimalkan waktu henti ke aplikasi yang bergantung pada database. Anda dapat mengurangi waktu migrasi dengan menggunakan AWS DMS untuk membuat tugas yang menangkap perubahan yang sedang berlangsung setelah Anda menyelesaikan migrasi muatan penuh awal melalui proses yang disebut change data capture (CDC).
- Alat Oracle asli - Anda dapat memigrasikan database dengan menggunakan alat Oracle asli, seperti Oracle dan Data Pump Export dan Data Pump Import dengan Oracle untuk CDC. GoldenGate Anda juga dapat menggunakan alat Oracle asli seperti utilitas Ekspor asli dan utilitas Impor asli untuk mengurangi waktu muat penuh.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database Oracle lokal
- Sebuah instans database Amazon RDS Oracle (DB)

Batasan

- Batas ukuran database: 64 TB

Versi produk

- Oracle versi 11g (versi 11.2.0.3.v1 dan yang lebih baru) dan hingga 12.2 dan 18c. Untuk daftar terbaru versi dan edisi yang didukung, lihat Amazon RDS for Oracle di dokumentasi AWS. Untuk versi Oracle yang didukung oleh AWS DMS, lihat Menggunakan database Oracle sebagai sumber AWS DMS dalam dokumentasi AWS DMS.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal

Tumpukan teknologi target

- Amazon RDS for Oracle

Arsitektur sumber dan target

Diagram berikut menunjukkan cara memigrasikan database Oracle lokal ke Amazon RDS for Oracle dengan menggunakan AWS DMS.

Diagram menunjukkan alur kerja berikut:

1. Buat atau gunakan pengguna database yang ada, berikan izin AWS DMS yang diperlukan kepada pengguna tersebut, aktifkan mode ARCHIVELOG, lalu atur pencatatan tambahan.
2. Konfigurasikan gateway internet antara jaringan lokal dan AWS.
3. Konfigurasikan titik akhir sumber dan target untuk AWS DMS.
4. Konfigurasikan tugas replikasi AWS DMS untuk memigrasikan data dari database sumber ke database target.
5. Selesaikan aktivitas pasca-migrasi pada basis data target.

Diagram berikut menunjukkan cara memigrasi database Oracle lokal ke Amazon RDS for Oracle dengan menggunakan alat Oracle asli.

Diagram menunjukkan alur kerja berikut:

1. Buat atau gunakan pengguna database yang ada dan berikan izin yang diperlukan untuk mencadangkan database Oracle dengan menggunakan utilitas Oracle Export (`exp`) dan Import (`imp`).
2. Konfigurasikan gateway internet antara jaringan lokal dan AWS.
3. Konfigurasikan klien Oracle pada host Bastion untuk mengambil database cadangan.
4. Unggah database cadangan ke bucket Amazon Simple Storage Service (Amazon S3).
5. Kembalikan cadangan database dari Amazon S3 ke database Amazon RDS for Oracle.
6. Konfigurasikan Oracle GoldenGate untuk CDC.
7. Selesaikan aktivitas pasca-migrasi pada basis data target.

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi pengaturan cloud dan lokal.
- Alat Oracle asli membantu Anda melakukan migrasi homogen. Anda dapat menggunakan Oracle Data Pump untuk memigrasikan data antara basis data sumber dan target Anda. Pola ini menggunakan Oracle Data Pump untuk melakukan beban penuh dari database sumber ke database target.
- Oracle GoldenGate membantu Anda melakukan replikasi logis antara dua atau lebih database. Pola ini digunakan GoldenGate untuk mereplikasi perubahan delta setelah pemuatan awal dengan menggunakan Oracle Data Pump.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dokumen proyek dan catat detail database.	1. Dokumentasikan sasaran migrasi Anda, persyarat an migrasi, pemangku kepentingan proyek utama, tonggak proyek, tenggat waktu proyek, metrik utama,	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	risiko migrasi, dan rencana mitigasi risiko. 2. Dokumentasikan informasi penting tentang basis data sumber Anda, termasuk RAM, IOPS, dan CPUs. Anda nantinya akan menggunakan informasi ini untuk menentukan instans DB target yang sesuai. 3. Validasi versi basis data sumber dan target Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi persyaratan penyimpanan.	Identifikasi dan dokumenta sikan kebutuhan penyimpanan Anda, termasuk yang berikut: 1. Hitung penyimpanan yang dialokasikan untuk instans DB sumber. 2. Kumpulkan metrik pertumbuhan historis dari instans DB sumber. 3. Forecast future growth untuk instans DB target. **Note** Untuk volume SSD Tujuan Umum (gp2), Anda mendapatk an tiga IOPS per 1 GB penyimpanan. Alokasikan penyimpan an dengan menghitun g jumlah total IOPS baca dan tulis pada database sumber.	DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih jenis instans yang tepat berdasarkan persyaratan komputasi.	1. Tentukan persyaratan komputasi instans DB target. 2. Identifikasi masalah kinerja. 3. Pertimbangkan faktor-fa ktor untuk menentukan jenis instans yang sesuai: • Pemanfaatan CPU dari instans DB sumber • IOPS (baca dan tulis) untuk instans DB sumber • Jejak memori pada instans DB sumber	SysAdmin
Identifikasi persyaratan keamanan akses jaringan.	1. Identifikasi dan dokumenta sikan persyaratan keamanan akses jaringan untuk basis data sumber dan target Anda. 2. Konfigurasikan grup keamanan yang sesuai untuk memungkinkan aplikasi berkomunikasi dengan database.	DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi strategi migrasi aplikasi.	1. Menentukan dan mendokumentasikan strategi cutover migrasi. 2. Tentukan dan dokumenta sikan tujuan waktu pemulihan (RTO) aplikasi Anda dan tujuan titik pemulihan (RPO), dan kemudian rencanakan pemotongan yang sesuai.	DBA, SysAdmin, Pemilik aplikasi
Identifikasi risiko migrasi.	Menilai database dan mendokumentasikan risiko dan mitigasi spesifik migrasi. Contoh: • Identifikasi tabel tanpa pencatatan dan sorot risiko kehilangan data jika terjadi pemulihan. • Ekstrak pengguna database sumber dan hak istimewa, dan sorot konflik dengan hak istimewa Amazon RDS. • Tinjau log peringatan untuk setiap kesalahan dan peringatan khusus Oracle. • Identifikasi fitur yang didukung dan tidak didukung dari instans DB target. • Tinjau fitur usang dari mesin versi DB target.	DBA

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Buat Amazon Virtual Private Cloud (Amazon VPC) baru untuk instans DB target.	SysAdmin
Buat grup keamanan.	Buat grup keamanan di VPC baru Anda untuk memungkinkan koneksi masuk ke instans DB.	SysAdmin
Buat instans Amazon RDS for Oracle DB.	Buat instans DB target dengan VPC baru dan grup keamanan, lalu mulai instance.	SysAdmin

Opsi 1 - Gunakan Oracle asli atau alat pihak ketiga untuk memigrasi data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan database sumber.	1. Buat direktori Pompa Data atau gunakan yang sudah ada. 2. Buat pengguna migrasi dan berikan izin untuk melakukan ekstrak Pompa Data. 3. Ekstrak peran, pengguna, dan tablespace dari database sumber sebagai skrip SQL. 4. Transfer dump Pompa Data yang diekstraksi ke direktori	DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	instans `data pump` DB target.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan database target.	1. Konfirmasikan bahwa semua opsi database (misalnya, teks dan Java) diinstal atau diaktifkan pada instans Amazon RDS for Oracle DB target. 2. Buat direktori Pompa Data atau gunakan yang sudah ada. 3. Buat pengguna migrasi dan berikan izin untuk melakukan impor Pompa Data. 4. Buat ruang tabel, pengguna, dan peran yang diperlukan pada instans DB target. 5. Impor dump ekspor Pompa Data yang ditransfer ke database target. 6. Buat indeks apa pun yang dikecualikan selama impor atau pembuatan objek. 7. Buat kendala apa pun yang dikecualikan selama impor. 8. Validasi atau kompilasi ulang objek yang tidak valid. 9. Membangun kembali indeks yang tidak valid. 10. Validasi jumlah objek database antara sumber dan database target.	DBA, SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	11. Selesaikan setiap perbedaan yang ditemukan antara jumlah objek.	

Opsi 2 - Gunakan AWS DMS untuk memigrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan datanya.	1. Bersihkan data dalam database sumber. 2. Buat contoh replikasi. 3. Buat titik akhir sumber dan titik akhir target. 4. Identifikasi jumlah tabel dan objek yang akan dimigrasi kan.	DBA
Migrasikan data.	1. Jatuhkan batasan kunci asing dan pemicu pada database target. 2. Jatuhkan indeks sekunder pada database target. 3. Konfigurasikan pengatura n tugas beban penuh AWS DMS dari database sumber ke database target. 4. Aktifkan kunci asing. 5. Aktifkan AWS DMS CDC untuk mereplikasi perubahan yang sedang berlangsung. 6. Aktifkan pemicu. 7. Perbarui urutannya.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. Validasi sumber dan data target.	

Potong ke database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.	1. Hentikan semua layanan aplikasi dan koneksi klien yang menunjuk ke Oracle. 2. Jalankan tugas AWS DMS. 3. Siapkan tugas rollback (misalnya, membalikkan CDC dari database Amazon RDS ke database Oracle lokal). 4. Validasi data. 5. Mulai layanan aplikasi pada database target baru dengan mengonfig urasi Amazon Route 53 ke instans Amazon RDS for Oracle DB baru. 6. Tambahkan CloudWatch pemantauan Amazon ke instans Amazon RDS for Oracle DB baru Anda.	DBA, SysAdmin, Pemilik aplikasi
Terapkan rencana rollback Anda.	1. Hentikan semua layanan aplikasi yang menunjuk ke instans Amazon RDS for Oracle DB.	DBA, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Kembalikan perubahan ke database Oracle lokal sumber dengan menggunakan tugas AWS DMS. 3. Hentikan tugas AWS DMS yang berjalan dari database Oracle lokal ke database Amazon RDS for Oracle. 4. Konfigurasikan aplikasi kembali pada database Oracle sumber. 5. Konfirmasikan penerapan rollback selesai.	

Menutup proyek migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Matikan atau hapus sumber daya AWS sementara, seperti instans replikasi AWS DMS dan bucket S3.	DBA, SysAdmin
Tinjau dokumen proyek.	Tinjau dokumen dan sasaran perencanaan migrasi, lalu konfirmasikan bahwa Anda telah menyelesaikan semua langkah migrasi yang diperlukan.	DBA, SysAdmin, Pemilik aplikasi
Kumpulkan metrik.	Catat metrik migrasi utama, termasuk berapa lama waktu yang dibutuhkan untuk menyelesaikan migrasi,	DBA, SysAdmin, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	persentase tugas manual vs berbasis alat, penghemat an biaya, dan metrik relevan lainnya.	
Tutup proyek.	Tutup proyek migrasi dan tangkap umpan balik tentang upaya tersebut.	DBA, SysAdmin, Pemilik aplikasi

Sumber daya terkait

Referensi

- Memigrasi database Oracle ke AWS Cloud (AWS Prescriptive Guidance)
- AWS Database Migration Service (dokumentasi AWS DMS)
- Harga Amazon RDS (dokumentasi Amazon RDS)

Tutorial dan video

- Memulai AWS Database Migration Service (dokumentasi AWS DMS)
- Sumber daya Amazon RDS (dokumentasi Amazon RDS)
- AWS Database Migration Service (DMS) (YouTube)

Memigrasikan database Oracle lokal ke Amazon RDS for Oracle menggunakan Oracle Data Pump

Mohan Annam dan Brian motzer, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Oracle dari pusat data lokal ke Amazon Relational Database Service (Amazon RDS) untuk instans Oracle DB dengan menggunakan Oracle Data Pump.

Polanya melibatkan pembuatan file dump data dari database sumber, menyimpan file dalam bucket Amazon Simple Storage Service (Amazon S3), dan kemudian memulihkan data ke instans Amazon RDS for Oracle DB. Pola ini berguna saat Anda menemukan batasan menggunakan AWS Database Migration Service (AWS DMS) untuk migrasi.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Izin yang diperlukan untuk membuat peran di AWS Identity and Access Management (IAM) dan untuk unggahan multipart Amazon S3
- Izin yang diperlukan untuk mengekspor data dari database sumber
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi

Versi produk

- Oracle Data Pump hanya tersedia untuk Oracle Database 10g Release 1 (10.1) dan versi yang lebih baru.

Arsitektur

Tumpukan teknologi sumber

- Database Oracle lokal

Tumpukan teknologi target

- Amazon RDS for Oracle
- Klien SQL (Pengembang Oracle SQL)
- Ember S3

Arsitektur sumber dan target

Alat

Layanan AWS

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya. Dalam pola ini, IAM digunakan untuk membuat peran dan kebijakan yang diperlukan untuk memigrasi data dari Amazon S3 ke Amazon RDS for Oracle.
- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di AWS Cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat lainnya

- Oracle Data Pump membantu Anda memindahkan data dan metadata dari satu database ke database lainnya dengan kecepatan tinggi. Dalam pola ini, Oracle Data Pump digunakan untuk mengekspor file dump data (.dmp) ke server Oracle, dan untuk mengimpornya ke Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Mengimpor data ke Oracle di Amazon RDS dalam dokumentasi Amazon RDS.
- Oracle SQL Developer adalah lingkungan pengembangan terintegrasi yang menyederhanakan pengembangan dan pengelolaan database Oracle baik dalam penerapan tradisional maupun berbasis cloud. Ini berinteraksi dengan database Oracle lokal dan Amazon RDS for Oracle untuk menjalankan perintah SQL yang diperlukan untuk mengekspor dan mengimpor data.

Epik

Buat Bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket.	Untuk membuat bucket S3, ikuti petunjuk dalam dokumentasi AWS.	Administrator sistem AWS

Buat peran IAM dan tetapkan kebijakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan izin IAM.	Untuk mengonfigurasi izin, ikuti petunjuk dalam dokumentasi AWS.	Administrator sistem AWS

Buat instans Amazon RDS for Oracle DB target dan kaitkan peran integrasi Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat target Amazon RDS for Oracle DB instans.	Untuk membuat instans Amazon RDS for Oracle, ikuti petunjuk dalam dokumentasi AWS.	Administrator sistem AWS
Kaitkan peran dengan instans DB.	Untuk mengaitkan peran dengan instans, ikuti instruksi dalam dokumentasi AWS.	DBA

Buat pengguna database pada database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat pengguna.	Connect ke target Amazon RDS for Oracle database dari Oracle SQL Developer atau SQL* Plus, dan jalankan perintah SQL berikut untuk membuat pengguna mengimpor skema ke dalam.
	`create user SAMPLE_SCHEMA identified by <PASSWORD>;`
	`grant create session, resource to <USER NAME>;`
	`alter user <USER NAME> quota 100M on users;`
		DBA

Buat file ekspor dari database sumber Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file dump data.	Untuk membuat file dump bernama `sample.dmp` dalam `DATA_PUMP_DIR` direktori untuk mengekspor `SAMPLE_SCHEMA` pengguna, gunakan skrip berikut.
	`DECLARE`
	`  hdnl NUMBER;`
	`BEGIN`
	`  hdnl := dbms_datapump.open(operation => 'EXPORT',`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`job_mode => 'SCHEMA', job_name => NULL); dbms_datapump.add_file(handle => hdnl, filename => 'sample.dmp', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_dump_file); dbms_datapump.add_file(handle => hdnl, filename => 'export.log', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file); dbms_datapump.metadata_filter(hdnl, 'SCHEMA_EXPR', 'IN ('''SAMPLE_SCHEMA''')'); dbms_datapump.start_job(hdnl); END;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`/` Tinjau detail ekspor dengan meninjau `export.log` file di `DATA_PUMP_DIR` direktori lokal Anda.	

Unggah file dump ke bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file dump data dari sumber ke bucket S3.	Menggunakan AWS CLI, jalankan perintah berikut. `aws s3 cp sample.dmp s3://<bucket_created_epic_1>/`	DBA

Unduh file ekspor dari bucket S3 ke instance RDS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh file dump data ke Amazon RDS	Untuk menyalin file dump `sample.dmp` dari bucket S3 ke database Amazon RDS for Oracle, jalankan perintah SQL berikut. Dalam contoh ini, `sample.dmp` file diunduh dari bucket S3 `my-s3-integration1` ke direktori Oracle. `DATA_PUMP_DIR` Pastikan Anda memiliki ruang disk yang cukup dialokasikan ke instance RDS Anda untuk	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	mengakomodasi database dan file ekspor.
	
	`-- If you want to download all the files in the S3 bucket remove the p_s3_prefix line.`
	
	`SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3( p_bucket_name => 'my-s3-integration', p_s3_prefix => 'sample.dmp', p_directory_name => 'DATA_PUMP_DIR') AS TASK_ID FROM DUAL;`
	
	Perintah sebelumnya mengeluarkan ID tugas. Untuk meninjau status unduhan dengan meninjau data di ID tugas, jalankan perintah berikut.
	
	`SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-<task_id>.log'));`
	
	Untuk melihat file dalam `DATA_PUMP_DIR` direktori, jalankan perintah berikut.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`SELECT filename, type,filesize/1024 /1024 size_megs ,to_char(mtime,'DD -MON-YY HH24:MI:SS') timestamp FROM TABLE(rdsadmin.rds _file_util.listdir (p_directory => upper('DATA_PUMP_D IR'))) order by 4;`	

Impor file dump ke database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan skema dan data ke Amazon RDS.	Untuk mengimpor file dump ke dalam skema `sample_sc hema` database, jalankan perintah SQL berikut dari SQL Developer atau SQL* Plus. `DECLARE hdnl NUMBER; BEGIN hdnl := DBMS_DATA PUMP.OPEN( operation => 'IMPORT', job_mode => 'SCHEMA', job_name= >null); DBMS_DATAPUMP.ADD_ FILE( handle => hdnl, filename => 'sample.d mp', directory => 'DATA_PUMP_DIR', filetype => dbms_data`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pump.ku$_file_type _dump_file); DBMS_DATAPUMP.ADD_FILE (handle => hdnl, filename => 'import.l og', directory => 'DATA_PUMP_DIR', filetype => dbms_data pump.ku$_file_type _log_file); DBMS_DATAPUMP. METADATA_FILTER(hd nl,'SCHEMA_EXPR',' IN (''SAMPLE_SCHEMA'')'); DBMS_DATAPUMP.START_J OB(hdnl); END; / Untuk melihat file log dari impor, jalankan perintah berikut. SELECT text FROM table(rdsadmin.rds _file_util.read_te xt_file('DATA_PUMP _DIR','import.log'));	

Hapus file dump dari direktori DATA_PUMP_DIR

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Daftar dan bersihkan file ekspor.	Daftar dan hapus file ekspor di `DATA_PUMP_DIR` direktori, jalankan perintah berikut.
	`-- List the files SELECT filename, type,filesize/1024 /1024 size_megs ,to_char(mtime,'DD -MON-YY HH24:MI:S S') timestamp FROM TABLE(rdsadmin.rds _file_util.listdir (p_directory => upper('DATA_PUMP_D IR'))) order by 4;`
	`-- Remove the files EXEC UTL_FILE. FREMOVE('DATA_PUMP _DIR','sample.dmp'); EXEC UTL_FILE.FREMOVE(' DATA_PUMP_DIR','im port.log');`
		Administrator sistem AWS

Sumber daya terkait

- Integrasi Amazon S3
- Buat instance DB
- Mengimpor data ke Oracle di Amazon RDS
- Dokumentasi Amazon S3
- Dokumentasi IAM
- Dokumentasi Amazon RDS

- Dokumentasi Pompa Data Oracle
- Pengembang Oracle SQL

Bermigrasi dari PostgreSQL di Amazon ke Amazon RDS untuk PostgreSQL menggunakan pglogical EC2

Rajesh Madiwale, Amazon Web Services

Ringkasan

Pola ini menguraikan langkah-langkah untuk memigrasikan database PostgreSQL (versi 9.5 dan yang lebih baru) dari Amazon Elastic Compute Cloud (Amazon) ke Amazon Relational Database EC2 Service (Amazon RDS) untuk PostgreSQL dengan menggunakan ekstensi pglogical PostgreSQL. Amazon RDS sekarang mendukung ekstensi pglogical untuk PostgreSQL versi 10.

Prasyarat dan batasan

Prasyarat

- Pilih jenis instans Amazon RDS yang tepat. Untuk informasi selengkapnya, lihat Jenis Instans Amazon RDS.
- Pastikan bahwa versi sumber dan target PostgreSQL adalah sama.
- Instal dan integrasikan ekstensi pglogical dengan PostgreSQL di Amazon. EC2

Versi produk

- PostgreSQL versi 10 dan yang lebih baru di Amazon RDS, dengan fitur yang didukung di Amazon RDS (lihat PostgreSQL di Amazon RDS dalam dokumentasi AWS). Pola ini diuji dengan memigrasikan PostgreSQL 9.5 ke PostgreSQL versi 10 di Amazon RDS, tetapi juga berlaku untuk versi PostgreSQL yang lebih baru di Amazon RDS.

Arsitektur

Arsitektur migrasi data

Alat

- ekstensi pglogis
- Utilitas asli PostgreSQL: pg_dump dan pg_restore

Epik

Migrasi data dengan menggunakan ekstensi pglogical

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon RDS PostgreSQL DB.	Siapkan instans PostgreSQL DB di Amazon RDS. Untuk petunjuk, lihat dokumentasi Amazon RDS for PostgreSQL.	DBA
Dapatkan dump skema dari database PostgreSQL sumber dan mengembalikannya ke database PostgreSQL target.	1. Gunakan utilitas pg_dump dengan `-s` opsi untuk menghasilkan file skema dari database sumber. 2. Gunakan utilitas psql dengan `-f` opsi untuk memuat skema ke dalam database target.	DBA
Nyalakan decoding logis.	Di grup parameter Amazon RDS DB, atur parameter `rds.logical_replication` statis ke 1. Untuk petunjuk, lihat dokumentasi Amazon RDS.	DBA
Buat ekstensi pglogical pada basis data sumber dan target.	1. Buat `pglogical` ekstensi pada database PostgreSQL sumber: `psql -h <amazon-ec2-endpoint> -d target-dbname -U target-dbuser -c "create extension pglogical ;"`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buat `pglogical` ekstensi pada database PostgreSQL target: `psql -h <amazon-rds-endpoint> -d source-dbname -U source-dbuser -c "create extension pglogical ;"`	

Buat penerbit di database PostgreSQL sumber.	Untuk membuat penerbit, jalankan:
	`psql -d dbname -p 5432 <<EOF`
	`SELECT pglogical.create_node( node_name := 'provider1', dsn := 'host=<ec2-endpoint> port=5432 dbname=source-dbname user=source-dbuser' );`
	`EOF`
		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat set replikasi, tambahkan tabel dan urutan.	Untuk membuat kumpulan replikasi pada database PostgreSQL sumber, dan untuk menambahkan tabel dan urutan ke set replikasi, jalankan: `psql -d dbname -p 5432 <<EOF SELECT pglogical .replication_set_add_all_tables('default', '{public}'::text[],synchronize_data := true); EOF`	DBA
Buat pelanggan.	Untuk membuat pelanggan pada database PostgreSQL target, jalankan: `psql -h <rds-endpoint> -d target-dbname -U target-dbuser <<EOF SELECT pglogical .create_node( node_name := 'subscriber1', dsn := 'host=<rds-endpoint> port=5432 dbname=target-dbname password=postgres user=target-dbuser' ); EOF`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat langganan.	Untuk membuat langganan pada database PostgreSQL target, jalankan: `psql -h <rds-endpoint> -d target -U postgres <<EOF SELECT pglogical .create_subscription( subscription_name := 'subscription1', replication_sets := array['default'], provider_dsn := 'host=<ec2-endpoint> port=5432 dbname=<source-dbname> password=<password> user=source-dbuser' );`	DBA

Validasi data Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa basis data sumber dan target.	Periksa basis data sumber dan target untuk mengonfirmasi bahwa data berhasil direplikasi. Anda dapat melakukan validasi dasar dengan menggunakan `select count(1)` dari tabel sumber dan target.	DBA

Sumber daya terkait

- Amazon RDS
- Replikasi logis untuk PostgreSQL di Amazon RDS (dokumentasi Amazon RDS)
- pglogis (repositori) GitHub
- Keterbatasan pglogical (file README GitHub repositori)
- Memigrasi PostgreSQL dari lokal atau Amazon ke EC2 Amazon RDS menggunakan replikasi logis (blog AWS Database)

Memigrasikan database PostgreSQL lokal ke Aurora PostgreSQL

Baji Shaik dan Jitender Kumar, Amazon Web Services

Ringkasan

Amazon Aurora PostgreSQL Compatible Edition menggabungkan kinerja dan ketersediaan database komersial kelas atas dengan kesederhanaan dan efektivitas biaya database open-source. Aurora memberikan manfaat ini dengan menskalakan penyimpanan di tiga Availability Zone di Wilayah AWS yang sama, dan mendukung hingga 15 instance replika baca untuk meningkatkan beban kerja baca dan menyediakan ketersediaan tinggi dalam satu Wilayah. Dengan menggunakan database global Aurora, Anda dapat mereplikasi database PostgreSQL hingga lima Wilayah untuk akses baca jarak jauh dan pemulihan bencana jika terjadi kegagalan Wilayah. Pola ini menjelaskan langkah-langkah untuk memigrasikan database sumber PostgreSQL lokal ke database yang kompatibel dengan Aurora PostgreSQL. Pola ini mencakup dua opsi migrasi: menggunakan AWS Data Migration Service (AWS DMS) atau menggunakan alat PostgreSQL asli (seperti pg_dump, pg_restore, dan psql) atau alat pihak ketiga.

Langkah-langkah yang dijelaskan dalam pola ini juga berlaku untuk menargetkan database PostgreSQL di Amazon Relational Database Service (Amazon RDS) dan Amazon Elastic Compute Cloud (Amazon) instans. EC2

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber PostgreSQL di pusat data lokal
- Instans DB yang kompatibel dengan Aurora PostgreSQL atau Amazon RDS for PostgreSQL DB instans

Batasan

- Batas ukuran database adalah 64 TB untuk Amazon RDS for PostgreSQL dan 128 TB untuk Aurora PostgreSQL kompatibel.
- Jika Anda menggunakan opsi migrasi AWS DMS, tinjau batasan AWS DMS saat menggunakan database PostgreSQL sebagai sumber.

Versi produk

- Untuk dukungan versi mayor dan minor PostgreSQL di Amazon RDS, lihat pembaruan Amazon RDS for PostgreSQL dalam dokumentasi Amazon RDS.
- Untuk dukungan PostgreSQL di Aurora, lihat pembaruan Amazon Aurora PostgreSQL dalam dokumentasi Aurora.
- Jika Anda menggunakan opsi migrasi AWS DMS, lihat versi PostgreSQL yang didukung dalam dokumentasi AWS DMS.

Arsitektur

Tumpukan teknologi sumber

- Database PostgreSQL lokal

Tumpukan teknologi target

- Instans DB yang kompatibel dengan Aurora PostgreSQL

Arsitektur sumber

Arsitektur target

Arsitektur migrasi data

Menggunakan AWS DMS

Menggunakan alat PostgreSQL asli

Alat

- AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan penyimpanan data ke AWS Cloud atau antara kombinasi konfigurasi cloud dan lokal. Layanan ini mendukung berbagai sumber dan basis data target. Untuk informasi tentang cara memvalidasi sumber PostgreSQL dan menargetkan versi dan edisi database yang didukung untuk digunakan dengan AWS DMS, lihat Menggunakan database PostgreSQL sebagai sumber AWS DMS. Kami

menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif.

- Alat PostgreSQL asli termasuk pg_dump, pg_restore, dan psql.

Epik

Menganalisis migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target.	Jika Anda menggunakan AWS DMS, pastikan Anda menggunakan PostgreSQL versi yang didukung.	DBA
Identifikasi jenis penyimpanan dan persyaratan kapasitas.	1. Hitung penyimpanan yang dialokasikan untuk instance database sumber. 2. Kumpulkan metrik pertumbuhan historis untuk contoh database sumber. 3. Antisipasi perkiraan pertumbuhan future untuk instance database target. 4. Alokasikan penyimpanan dengan menghitung jumlah total IOPS baca dan tulis pada database sumber. Volume General Purpose SSD (gp2) menyediakan 3 IOPS untuk setiap 1 GB penyimpanan.	DBA, Administrator sistem
Pilih jenis instans, kapasitas , fitur penyimpanan, dan fitur jaringan yang tepat.	Tentukan persyaratan komputasi dari instance database target. Tinjau masalah kinerja yang	DBA, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	diketahui yang mungkin memerlukan perhatian tambahan. Pertimbangkan faktor-faktor berikut untuk menentukan jenis instance yang sesuai: • Pemanfaatan CPU dari instance database sumber • IOPS (operasi baca dan tulis) untuk instance database sumber • Jejak memori pada instance database sumber Untuk informasi selengkapnya, lihat kelas instans Aurora DB dalam dokumentasi Aurora.	
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.	Tentukan kelompok keamanan yang sesuai yang akan memungkinkan aplikasi untuk berbicara dengan database.	DBA, Administrator sistem
Identifikasi strategi migrasi aplikasi.	• Tentukan strategi cutover migrasi berdasarkan kompleksitas aplikasi Anda. • Tentukan tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO) untuk aplikasi, dan rencanakan pemotongan yang sesuai.	DBA, Pemilik aplikasi, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Buat virtual private cloud (VPC) baru untuk instance database target.	Administrator sistem
Buat grup keamanan.	Buat grup keamanan dalam VPC (seperti yang ditentukan dalam epik sebelumnya) untuk memungkinkan koneksi masuk ke instance database.	Administrator sistem
Konfigurasikan dan mulai cluster Aurora DB.	Buat instance database target dengan VPC baru dan grup keamanan dan mulai instance.	Administrator sistem

Migrasikan data - opsi 1 (menggunakan AWS DMS)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi langkah-langkah pra-migrasi.	1. Bersihkan data di database sumber. 2. Buat contoh replikasi. 3. Buat titik akhir sumber dan target. 4. Identifikasi jumlah tabel dan objek yang tersedia untuk dimigrasi.	DBA
Lengkapi langkah migrasi.	1. Jatuhkan batasan kunci asing dan pemicu pada database target. 2. Jatuhkan indeks sekunder pada database target.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Gunakan tugas beban penuh untuk memigrasikan data dari sumber ke database target. 4. Aktifkan kunci asing. 5. Jika Anda menggunakan migrasi flash-cut dan aplikasi Anda memerlukan waktu henti minimal, aktifkan change data capture (CDC) untuk mereplikasi perubahan yang sedang berlangsung 6. Aktifkan pemicu. 7. Perbarui urutan. 8. Validasi sumber dan data target.	
Validasi data.	Untuk memastikan bahwa data Anda dimigrasikan secara akurat dari sumber ke target, ikuti langkah-langkah validasi data dalam dokumentasi AWS DMS.	DBA

Migrasikan data - opsi 2 (menggunakan pg_dump dan pg_restore)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan database sumber.	1. Buat direktori untuk menyimpan cadangan pg_dump jika belum ada.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buat pengguna migrasi yang memiliki izin untuk menjalankan pg_dump pada objek database. 3. Connect ke EC2 instance dan jalankan pg_dump backup. Untuk informasi selengkap nya, lihat dokumentasi pg_dump dan panduan dalam dokumentasi AWS DMS.	
Siapkan database target.	1. Buat pengguna migrasi yang memiliki izin untuk menggunakan pg_restore pada objek database. 2. Impor dump database dengan menggunakan pg_restore. Untuk informasi selengkap nya, lihat dokumentasi pg_restore dan panduan dalam dokumentasi AWS DMS.	DBA
Validasi data.	1. Bandingkan jumlah objek database antara sumber dan basis data target. 2. Selesaikan setiap perbedaan yang ditemukan antara jumlah objek.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.	Terapkan strategi migrasi aplikasi yang Anda buat di epik pertama.	DBA, Pemilik aplikasi, Administrator sistem

Potong ke basis data target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.	1. Hentikan semua layanan aplikasi dan koneksi klien yang mengarah ke database PostgreSQL lokal. 2. Jalankan tugas AWS DMS. 3. Siapkan tugas rollback (membalikkan CDC dari Aurora PostgreSQL yang kompatibel dengan database PostgreSQL lokal) jika diperlukan. 4. Validasi data. 5. Mulai layanan aplikasi pada target baru dengan mengonfigurasi Amazon Route 53 ke instans DB baru yang kompatibel dengan Aurora PostgreSQL. 6. Tambahkan pemantauan Amazon CloudWatch dan Performance Insights pada instans DB baru yang	DBA, Pemilik aplikasi, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kompatibel dengan Aurora PostgreSQL.	
Jika Anda perlu memutar kembali migrasi.	1. Hentikan semua layanan aplikasi yang mengarah ke database yang kompatibel dengan Aurora PostgreSQL. 2. Kembalikan perubahan ke database PostgreSQL sumber lokal dengan menggunakan tugas AWS DMS yang Anda buat di cerita sebelumnya. 3. Hentikan tugas AWS DMS yang berjalan dari database PostgreSQL lokal ke database yang kompatibel dengan Aurora PostgreSQL. 4. Konfigurasikan aplikasi sehingga menunjuk kembali ke sumber database PostgreSQL lokal. 5. Konfirmasikan bahwa semua penerapan rollback selesai.	DBA, Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya.	Matikan sumber daya AWS sementara.	DBA, Administrator sistem
Validasi dokumen.	Tinjau dan validasi dokumen proyek.	DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik.	Kumpulkan metrik di sekitar waktu untuk bermigrasi, persen penghematan biaya manual versus alat, dan sebagainya.	DBA, Pemilik aplikasi, Administrator sistem
Tutup proyek.	Tutup proyek dan berikan umpan balik apa pun.	DBA, Pemilik aplikasi, Administrator sistem

Sumber daya terkait

Referensi

- Layanan Migrasi Data AWS
- VPCs dan Amazon Aurora
- Harga Amazon Aurora
- Menggunakan database PostgreSQL sebagai sumber AWS DMS
- Cara membuat instance replikasi AWS DMS
- Cara membuat titik akhir sumber dan target menggunakan AWS DMS

Sumber daya tambahan

- Memulai dengan AWS DMS
- step-by-stepPenelusuran migrasi data
- Sumber daya Amazon Aurora

Memigrasi database Microsoft SQL Server lokal ke Microsoft SQL Server di Amazon yang menjalankan Linux EC2

Tirumala Dasari, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara bermigrasi dari database Microsoft SQL Server lokal yang berjalan di Microsoft Windows, ke Microsoft SQL Server di instans Amazon Elastic Compute Cloud (EC2Amazon) Linux dengan menggunakan utilitas pencadangan dan pemulihan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Amazon EC2 Linux AMI (Gambar Mesin Amazon) dengan Microsoft SQL Server
- AWS Direct Connect antara Windows lokal dan Microsoft SQL Server di instans Linux EC2

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server lokal

Tumpukan teknologi target

- EC2 Instance Linux dengan database Microsoft SQL Server

Arsitektur migrasi database

Alat

- WinSCP - Alat ini memungkinkan pengguna Windows untuk dengan mudah berbagi file dengan pengguna Linux.
- Sqlcmd - Utilitas baris perintah ini memungkinkan Anda mengirimkan pernyataan T-SQL atau batch ke instance SQL Server lokal dan jarak jauh. Utilitas ini sangat berguna untuk tugas-tugas database berulang seperti pemrosesan batch atau pengujian unit.

Epik

Siapkan instance EC2 Linux dengan SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih AMI yang menyediak an sistem operasi Linux dan termasuk Microsoft SQL Server.		Sysadmin
Konfigurasikan AMI untuk membuat EC2 instance.		Sysadmin
Buat aturan masuk dan keluar untuk grup keamanan.		Sysadmin
Konfigurasikan EC2 instance Linux untuk database Microsoft SQL Server.		DBA
Buat pengguna dan berikan izin seperti pada database sumber.		Pemilik aplikasi, DBA
Instal alat SQL Server dan utilitas sqlcmd pada instance Linux. EC2		DBA

Cadangkan database dan pindahkan file cadangan ke EC2 instance Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Cadangkan database SQL Server lokal.		DBA
Instal WinSCP di Microsoft SQL Server.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pindahkan file cadangan ke EC2 instance Linux yang menjalankan Microsoft SQL Server.		DBA

Kembalikan database pada EC2 instance Linux yang menjalankan SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan database dari file cadangan database dengan menggunakan utilitas sqlcmd.		DBA
Validasi objek database dan data.		Pengembang, Insinyur uji

Potong dari Windows SQL Server ke Windows SQL Server pada instance Linux EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi objek database dan data.		Pengembang, Insinyur uji
Potong dari database Microsoft SQL Server lokal ke EC2 instance Linux yang menjalankan Microsoft SQL Server.		DBA

Sumber daya terkait

- Cara mengkonfigurasi SQL Server 2017 di Amazon Linux dan Ubuntu AMIs
- Instalasi alat SQL pada instance Linux

- Pencadangan dan pemulihan dari database Microsoft SQL Server lokal ke Microsoft SQL Server pada instance Linux EC2

Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan server tertaut

Kevin Yung, Viqash Adwani, dan Vishal Singh, Amazon Web Services

Ringkasan

Server tertaut memungkinkan Microsoft SQL Server untuk menjalankan pernyataan SQL pada contoh lain dari server database. Pola ini menjelaskan bagaimana Anda dapat memigrasikan database Microsoft SQL Server lokal ke Amazon Relational Database Service (Amazon RDS) untuk Microsoft SQL Server untuk mencapai biaya yang lebih rendah dan ketersediaan yang lebih tinggi. Saat ini, Amazon RDS for Microsoft SQL Server tidak mendukung koneksi di luar jaringan Amazon Virtual Private Cloud (Amazon VPC).

Anda dapat menggunakan pola ini untuk mencapai tujuan berikut:

- Untuk memigrasikan Microsoft SQL Server ke Amazon RDS untuk Microsoft SQL Server tanpa merusak kemampuan server tertaut.
- Untuk memprioritaskan dan memigrasikan Microsoft SQL Server tertaut dalam gelombang yang berbeda.

Prasyarat dan batasan

Prasyarat

- Periksa apakah Microsoft SQL Server di Amazon RDS mendukung fitur yang Anda butuhkan.
- Pastikan Anda dapat menggunakan Amazon RDS untuk Microsoft SQL Server dengan susunan atau susunan default yang diatur di atas tingkat database.

Arsitektur

Tumpukan teknologi sumber

- Database lokal (Microsoft SQL Server)

Tumpukan teknologi target

- Amazon RDS for SQL Server

Arsitektur negara sumber

Arsitektur negara sasaran

Dalam status target, Anda memigrasikan Microsoft SQL Server ke Amazon RDS untuk Microsoft SQL Server dengan menggunakan server tertaut. Arsitektur ini menggunakan Network Load Balancer untuk mem-proxy lalu lintas dari Amazon RDS untuk Microsoft SQL Server ke server lokal yang menjalankan Microsoft SQL Server. Diagram berikut menunjukkan kemampuan proxy terbalik untuk Network Load Balancer.

Alat

- AWS CloudFormation
- Penyeimbang Beban Jaringan
- Amazon RDS untuk SQL Server di beberapa Availability Zone (Multi-) AZs
- AWS Database Migration Service (AWS DMS)

Epik

Buat VPC landing zone

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat alokasi CIDR.		AWS SysAdmin
Buat cloud pribadi virtual (VPC).		AWS SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat subnet VPC.		AWS SysAdmin
Buat daftar kontrol akses subnet (ACLs).		AWS SysAdmin
Buat tabel rute subnet.		AWS SysAdmin
Buat koneksi dengan AWS Direct Connect atau AWS Virtual Private Network (VPN).		AWS SysAdmin

Migrasikan database ke Amazon RDS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon RDS for Microsoft SQL Server DB.		AWS SysAdmin
Buat instans replikasi AWS DMS.		AWS SysAdmin
Buat titik akhir basis data sumber dan target di AWS DMS.		AWS SysAdmin
Buat tugas migrasi dan atur replikasi berkelanjutan ke ON setelah beban penuh.		AWS SysAdmin
Meminta perubahan firewall untuk mengizinkan Amazon RDS for Microsoft SQL Server mengakses database SQL Server lokal.		AWS SysAdmin
Buat Network Load Balancer.		AWS SysAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup target yang menargetkan server database di pusat data Anda	Kami menyarankan Anda menggunakan nama host dalam pengaturan target untuk menggabungkan peristiwa failover pusat data (DC).	AWS SysAdmin
Jalankan pernyataan SQL untuk penyiapan server tertaut.	Jalankan pernyataan SQL untuk menambahkan server tertaut dengan menggunak an alat manajemen Microsoft SQL terhadap instans Amazon RDS for Microsoft SQL Server DB. Dalam pernyataan SQL, atur @datasrc untuk menggunak an nama host Network Load Balancer. Tambahkan kredensil login server tertaut dengan menggunakan alat manajemen Microsoft SQL terhadap instans Amazon RDS for Microsoft SQL Server DB.	AWS SysAdmin
Menguji dan memvalidasi fungsi SQL Server.		AWS SysAdmin
Buat cutover.		AWS SysAdmin

Sumber daya terkait

- Tugas Manajemen Umum untuk Microsoft SQL Server di Amazon RDS
- Kumpulan dan Set Karakter untuk Microsoft SQL Server
- Dokumentasi Network Load Balancer

- Menerapkan Server Tertaut dengan Amazon RDS untuk Microsoft SQL Server (posting blog)

Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan metode pencadangan dan pemulihan asli

Tirumala Dasari, David Queiroz, dan Vishal Singh, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Microsoft SQL Server lokal ke Amazon Relational Database Service (Amazon RDS) untuk instans SQL Server DB (migrasi homogen). Proses migrasi didasarkan pada metode pencadangan dan pemulihan SQL Server asli. Menggunakan SQL Server Management Studio (SSMS) untuk membuat file cadangan database, dan bucket Amazon Simple Storage Service (Amazon S3) untuk menyimpan file cadangan sebelum memulihkannya di Amazon RDS for SQL Server.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Kebijakan peran AWS Identity and Access Management (IAM) untuk mengakses bucket S3 dan instans Amazon RDS for SQL Server DB.

Batasan

- Proses yang dijelaskan dalam pola ini hanya memigrasikan database. Login SQL atau pengguna database, termasuk pekerjaan Agen SQL Server apa pun, tidak dimigrasikan karena memerlukan langkah tambahan.

Versi produk

- SQL Server 2012-2017. Untuk daftar terbaru versi dan fitur yang didukung, lihat Microsoft SQL Server di Amazon RDS dalam dokumentasi AWS.

Arsitektur

Tumpukan teknologi sumber

- Database Microsoft SQL Server lokal

Tumpukan teknologi target

- Amazon RDS for SQL Server DB instans

Arsitektur migrasi data

Alat

- Microsoft SQL Server Management Studio (SSMS) adalah lingkungan terintegrasi untuk mengelola infrastruktur SQL Server. Ini menyediakan antarmuka pengguna dan sekelompok alat dengan editor skrip kaya yang berinteraksi dengan SQL Server.

Epik

Membuat instans Amazon RDS for SQL Server DB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih SQL Server sebagai mesin database di Amazon RDS for SQL Server.		DBA
Pilih SQL Server Express Edition.		DBA
Tentukan detail database.	Untuk informasi selengkap nya tentang membuat instans DB, lihat dokumentasi Amazon RDS.	DBA, Pemilik aplikasi

Membuat file cadangan dari database SQL Server lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke database SQL Server lokal melalui SSMS.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cadangan database.	Untuk instruksi, lihat dokumentasi SSMS.	DBA, Pemilik aplikasi

Unggah file cadangan ke Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember di Amazon S3.	Untuk informasi lebih lanjut, lihat Dokumentasi Amazon S3.	DBA
Unggah file cadangan ke bucket S3.	Untuk informasi lebih lanjut, lihat Dokumentasi Amazon S3.	SysOps administrator

Kembalikan database di Amazon RDS for SQL Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan grup opsi ke Amazon RDS.	1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/. 2. Di panel navigasi, pilih Grup opsi, Buat grup. 3. Lengkapi informasi untuk grup opsi, lalu pilih Buat. 4. Tambahkan `SQLSERVER_BACKUP_RESTORE` opsi ke grup opsi, lalu pilih Tambahkan opsi. Untuk informasi selengkapnya, lihat dokumentasi Amazon RDS.	SysOps administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan database.	1. Connect ke Amazon RDS for SQL Server melalui SSMS. 2. Panggil prosedur yang `msdb.dbo.rds_restore_database` disimpan untuk memulihkan database.	DBA

Validasi basis data target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi objek dan data.	Validasi objek dan data antara database sumber dan Amazon RDS for SQL Server. **Note** Tugas ini hanya memigrasikan database. Login dan pekerjaan tidak akan dimigrasikan.	Pemilik aplikasi, DBA

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengarahkan lalu lintas aplikasi.	Setelah validasi, arahkan lalu lintas aplikasi ke instans	Pemilik aplikasi, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Amazon RDS for SQL Server DB.	

Sumber daya terkait

- Dokumentasi Amazon S3
- Dokumentasi Amazon RDS for SQL Server
- Pilihan untuk Microsoft SQL Server Database Engine

Migrasikan database Microsoft SQL Server ke Aurora MySQL dengan menggunakan AWS DMS dan AWS SCT

Mark Szalkiewicz dan Pavan Pusuluri, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database Microsoft SQL Server yang ada di tempat atau di instans Amazon Elastic Compute Cloud (Amazon EC2) ke Amazon Aurora MySQL. Pola ini menggunakan AWS Database Migration Service (AWS DMS) dan AWS Schema Conversion Tool (AWS SCT) untuk migrasi data dan konversi skema.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber Microsoft SQL Server di pusat data lokal atau pada instans EC2
- Driver Java Database Connectivity (JDBC) untuk konektor AWS SCT, diinstal pada mesin lokal atau instance EC2 tempat AWS SCT diinstal

Batasan

- Batas ukuran database: 64 TB

Versi produk

- Microsoft SQL Server 2008, 2008R2, 2012, 2014, 2016, dan 2017 untuk edisi Enterprise, Standard, Workgroup, dan Developer. Edisi Web dan Express tidak didukung oleh AWS DMS. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database Microsoft SQL Server sebagai Sumber untuk AWS DMS. Kami menyarankan Anda menggunakan AWS DMS versi terbaru untuk versi dan dukungan fitur yang paling komprehensif. Untuk informasi tentang versi Microsoft SQL Server yang didukung oleh AWS SCT, lihat dokumentasi AWS SCT.
- MySQL versi 5.5, 5.6, dan 5.7. Untuk daftar terbaru versi yang didukung, lihat Menggunakan Database yang kompatibel dengan MySQL sebagai Target untuk AWS DMS.

Arsitektur

Tumpukan teknologi sumber

Salah satu dari yang berikut:

- Database Microsoft SQL Server lokal
- Database Microsoft SQL Server pada sebuah instance EC2

Tumpukan teknologi target

- Aurora MySQL

Arsitektur migrasi data

- Dari database Microsoft SQL Server yang berjalan di AWS Cloud

- Dari database Microsoft SQL Server yang berjalan di pusat data lokal

Alat

- AWS DMS - AWS Data Migration Service (AWS DMS) membantu Anda memigrasikan data ke dan dari basis data komersial dan sumber terbuka yang banyak digunakan, termasuk Oracle, SQL Server, MySQL, dan PostgreSQL. Anda dapat menggunakan AWS DMS untuk memigrasikan data Anda ke AWS Cloud, antar instans lokal (melalui penyiapan AWS Cloud), atau antara kombinasi pengaturan cloud dan lokal.
- AWS SCT - AWS Schema Conversion Tool (AWS SCT) memudahkan migrasi database heterogen dengan mengonversi skema basis data sumber secara otomatis dan sebagian besar kode kustom ke format yang kompatibel dengan basis data target.

Epik

Bersiaplah untuk migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi basis data sumber dan target dan mesin.		DBA
Buat grup keamanan keluar untuk basis data sumber dan target.		SysAdmin
Buat dan konfigurasikan EC2 instance untuk AWS SCT, jika diperlukan.		DBA
Unduh versi terbaru AWS SCT dan driver terkait.		DBA
Menambahkan dan memvalidasi pengguna prasyarat dan hibah dalam database sumber.		DBA
Buat proyek AWS SCT untuk beban kerja dan sambungkan ke database sumber.		DBA
Menghasilkan laporan penilaian dan mengevaluasi kelayakan.		DBA

Siapkan database target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans DB Amazon RDS target, menggunakan Amazon Aurora sebagai mesin database.		DBA
Ekstrak daftar pengguna, peran, dan hibah dari sumbernya.		DBA
Memetakan pengguna database yang ada untuk pengguna database baru.		Pemilik aplikasi
Buat pengguna di database target.		DBA
Terapkan peran dari langkah sebelumnya ke database target.		DBA
Tinjau opsi database, parameter, file jaringan, dan tautan basis data dalam basis data sumber, dan kemudian evaluasi penerapannya ke basis data target.		DBA
Terapkan pengaturan yang relevan ke target.		DBA

Transfer objek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan konektivitas AWS SCT ke database target.		DBA
Konversikan skema menggunakan AWS SCT.	AWS SCT secara otomatis mengonversi skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan basis data target. Kode apa pun yang tidak dapat dikonversi oleh alat secara otomatis ditandai dengan jelas sehingga Anda dapat mengonversinya sendiri.	DBA
Tinjau laporan SQL yang dihasilkan dan simpan kesalahan dan peringatan apa pun.		DBA
Terapkan perubahan skema otomatis ke target atau simpan sebagai file.sql.		DBA
Validasi bahwa AWS SCT membuat objek pada target.		DBA
Menulis ulang, menolak, atau mendesain ulang item apa pun secara manual yang gagal dikonversi secara otomatis.		DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan peran yang dihasilkan dan hibah pengguna dan tinjau pengecualian apa pun.		DBA

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan metode migrasi.		DBA
Buat instance replikasi dari konsol AWS DMS.	Untuk informasi terperinci tentang penggunaan AWS DMS, lihat tautan di bagian “Sumber daya terkait”.	DBA
Buat titik akhir sumber dan target.		DBA
Buat tugas replikasi.		DBA
Mulai tugas replikasi dan pantau log.		DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan AWS SCT untuk menganalisis dan mengonversi item SQL dalam kode aplikasi.	Ketika Anda mengonversi skema database Anda dari satu mesin ke mesin lain, Anda juga perlu memperbarui kode SQL dalam aplikasi Anda untuk berinteraksi dengan mesin database baru, bukan yang lama. Anda	Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dapat melihat, menganalisis, mengedit, dan menyimpan kode SQL yang dikonvers i. Untuk informasi terperinc i tentang penggunaan AWS SCT, lihat tautan di bagian “Sumber daya terkait”.	
Buat server aplikasi baru di AWS.		Pemilik aplikasi
Migrasikan kode aplikasi ke server baru.		Pemilik aplikasi
Konfigurasikan server aplikasi untuk database target dan driver.		Pemilik aplikasi
Perbaiki kode apa pun yang khusus untuk mesin basis data sumber dalam aplikasi.		Pemilik aplikasi
Optimalkan kode aplikasi untuk mesin target.		Pemilik aplikasi

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan setiap pengguna baru, hibah, dan perubahan kode ke target.		DBA
Kunci aplikasi untuk setiap perubahan.		Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi bahwa semua perubahan disebarkan ke database target.		DBA
Arahkan server aplikasi baru ke database target.		Pemilik aplikasi
Periksa kembali semuanya.		Pemilik aplikasi
Pergilah hidup.		Pemilik aplikasi

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara (instans replikasi AWS DMS dan EC2 instans yang digunakan untuk AWS SCT).		DBA, Pemilik aplikasi
Perbarui umpan balik tentang proses AWS DMS untuk tim internal.		DBA, Pemilik aplikasi
Merevisi proses AWS DMS dan meningkatkan template jika perlu.		DBA, Pemilik aplikasi
Meninjau dan memvalidasi dokumen proyek.		DBA, Pemilik aplikasi
Kumpulkan metrik di sekitar waktu untuk bermigrasi, persen penghematan biaya		DBA, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
manual versus alat, dan sebagainya.		
Tutup proyek dan berikan umpan balik apa pun.		DBA, Pemilik aplikasi

Sumber daya terkait

Referensi

- Panduan Pengguna AWS DMS
- Panduan Pengguna AWS SCT
- Harga Amazon Aurora

Tutorial dan video

- Memulai AWS Database Migration Service
- Memulai dengan AWS Schema Conversion Tool
- Sumber daya Amazon RDS
- Panduan AWS DMS Step-by-Step

Memigrasi database MariaDB lokal ke Amazon RDS for MariaDB menggunakan alat bawaan

Shyam Sunder Rakhecha, Amazon Web Services

Ringkasan

Pola ini memberikan panduan untuk memigrasikan database MariaDB lokal ke Amazon Relational Database Service (Amazon RDS) untuk MariaDB dengan menggunakan alat bawaan. Jika Anda memiliki alat MySQL diinstal, Anda dapat menggunakan mysql dan mysqldump. Jika Anda telah menginstal alat MariaDB, Anda dapat menggunakan mariadb dan mariadb-dump. Alat MySQL dan MariaDB memiliki asal yang sama, tetapi ada perbedaan kecil di MariaDB versi 10.6 dan yang lebih baru.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Database sumber MariaDB di pusat data lokal

Batasan

- Batas ukuran database: 64 TB

Versi produk

- MariaDB versi 10.0-10.6 (untuk daftar terbaru versi yang didukung, lihat MariaDB di Amazon RDS dalam dokumentasi AWS)

Arsitektur

Tumpukan teknologi sumber

- Basis data MariaDB di pusat data lokal

Tumpukan teknologi target

- Amazon RDS untuk instance DB MariaDB

Arsitektur target

Arsitektur migrasi data

Alat

- Alat MySQL asli: mysql dan mysqldump
- Alat MariaDB asli: mariadb dan mariadb-dump

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi dan mesin basis data sumber dan target.		DBA
Identifikasi persyaratan perangkat keras untuk instance server target.		DBA, Administrator sistem
Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).		DBA, Administrator sistem
Pilih jenis instans yang tepat berdasarkan kapasitas, fitur penyimpanan, dan fitur jaringan.		DBA, Administrator sistem
Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.		DBA, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi strategi migrasi aplikasi.		DBA, Pemilik aplikasi, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).		Administrator sistem
Buat grup keamanan.		Administrator sistem
Konfigurasikan dan mulai instans Amazon RDS DB yang menjalankan MariaDB.		Administrator sistem

Migrasikan data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan alat asli untuk memigrasikan objek dan data database.	Dalam database sumber, gunakan mysqldump atau mariadb-dump untuk membuat file output yang berisi objek database dan data. Dalam database target, gunakan mysql atau mariadb untuk mengembalikan data.	DBA
Validasi data.	Periksa basis data sumber dan target untuk mengonfir masi bahwa migrasi data berhasil.	DBA

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strategi migrasi aplikasi.		DBA, Pemilik aplikasi, Administrator sistem

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Alihkan klien aplikasi ke infrastruktur baru.		DBA, Pemilik aplikasi, Administrator sistem

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya AWS sementara.		Administrator sistem
Tinjau dan validasi dokumen proyek.		DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik di sekitar waktu untuk bermigrasi, penghematan biaya yang disediakan oleh alat, dan sebagainya.		DBA, Pemilik aplikasi, Administrator sistem
Tutup proyek dan berikan umpan balik.		DBA, Pemilik aplikasi, Administrator sistem

Sumber daya terkait

Referensi Amazon RDS

- Amazon RDS for MariaDB
- Amazon Virtual Private Cloud VPCs dan Amazon RDS
- Penyebaran Multi-AZ Amazon RDS
- Harga Amazon RDS

Referensi MySQL dan MariaDB

- mariadb-dump/mysqldump
- Klien baris perintah mysql

Tutorial dan video

- Memulai dengan Amazon RDS

Memigrasikan database MySQL lokal ke Aurora MySQL

Igor Obradovic, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan database sumber MySQL lokal ke Amazon Aurora Edisi yang kompatibel dengan MySQL. Ini menjelaskan dua opsi untuk migrasi: menggunakan AWS Database Migration Service (AWS DMS) atau menggunakan alat MySQL asli seperti mysqldbcopy dan mysqldump.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Database MySQL sumber di pusat data lokal

Batasan

- Batas ukuran database: 128 TB

Versi produk

- MySQL versi 8.0 (Aurora MySQL versi 3) tersedia di bawah dukungan standar.
- MySQL versi 5.7 (Aurora MySQL versi 2) tersedia di bawah dukungan diperpanjang, dengan biaya tambahan.

Untuk daftar terbaru versi yang didukung, lihat versi Amazon Aurora dalam dokumentasi. AWS Jika Anda menggunakan AWS DMS, lihat juga Menggunakan Database yang kompatibel dengan MySQL sebagai Target untuk versi AWS DMS MySQL yang didukung oleh. AWS DMS

Arsitektur

Tumpukan teknologi sumber

- Database MySQL lokal

Tumpukan teknologi target

- Edisi yang Kompatibel dengan Amazon Aurora MySQL

Arsitektur target

Data Aurora disimpan dalam volume cluster, yang merupakan volume virtual tunggal yang menggunakan solid state drive ()SSDs. Volume cluster terdiri dari salinan data di tiga Availability Zone dalam satu Wilayah AWS. Karena data secara otomatis direplikasi di seluruh Availability Zone, ini sangat tahan lama dengan kemungkinan kehilangan data yang lebih kecil.

Aurora secara otomatis membagi volume database Anda menjadi segmen 10 GB yang tersebar di banyak disk. Setiap potongan 10 GB volume database Anda direplikasi enam cara, di tiga Availability Zone. Diagram berikut menggambarkan hubungan antara volume cluster, instance DB penulis, dan instance DB pembaca dalam cluster Aurora DB, dan pemisahan kapasitas komputasi dan penyimpanan. Untuk informasi lebih lanjut tentang arsitektur ini, lihat dokumentasi Aurora dan FAQ.

Arsitektur migrasi data

Menggunakan AWS DMS:

Diagram berikut menggambarkan migrasi database MySQL lokal ke cluster yang kompatibel dengan Aurora MySQL di, menggunakan. AWS Cloud AWS DMS

Menggunakan alat MySQL asli:

Diagram berikut mengilustrasikan migrasi database MySQL lokal ke klaster yang kompatibel dengan Aurora MySQL di, menggunakan alat MySQL asli seperti mysqldbcopy dan AWS Cloud mysqldump.

Alat

- AWS Database Migration Service (AWS DMS) mendukung beberapa mesin basis data sumber dan target. Untuk informasi tentang sumber MySQL dan basis data target yang didukung AWS DMS oleh, lihat Memigrasi Database yang kompatibel dengan MySQL ke. AWS Kami menyarankan Anda menggunakan versi terbaru AWS DMS untuk versi dan dukungan fitur yang paling komprehensif.
- mysqldbcopy adalah utilitas MySQL yang menyalin database MySQL pada satu server atau antar server.

- mysqldump adalah utilitas MySQL yang membuat file dump dari database MySQL untuk tujuan pencadangan atau migrasi.

Epik

Rencanakan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi dan mesin.	Validasi versi database dan mesin untuk basis data sumber dan target.	DBA
Identifikasi persyaratan perangkat keras.	Identifikasi persyaratan perangkat keras untuk instance server target.	DBA, Administrator sistem
Identifikasi persyaratan penyimpanan.	Identifikasi persyaratan penyimpanan (jenis dan kapasitas penyimpanan).	DBA, Administrator sistem
Pilih jenis instans.	Pilih jenis instans yang tepat berdasarkan persyaratan komputasi, penyimpanan, dan jaringan Anda.	DBA, Administrator sistem
Tentukan persyaratan keamanan akses jaringan.	Identifikasi persyaratan keamanan akses jaringan untuk basis data sumber dan target.	DBA, Administrator sistem
Tentukan strategi.	Identifikasi strategi migrasi aplikasi.	DBA, Pemilik aplikasi, Administrator sistem

Konfigurasikan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cloud pribadi virtual (VPC).	Untuk petunjuk, lihat Membuat VPC di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).	Administrator sistem
Buat grup keamanan.	Untuk petunjuknya, lihat Membuat grup keamanan untuk VPC Anda di dokumentasi Amazon VPC.	Administrator sistem
Konfigurasikan dan mulai cluster DB yang kompatibel dengan Aurora MySQL di. Akun AWS	Untuk petunjuknya, lihat Membuat klaster DB Amazon Aurora di dokumentasi Aurora.	Administrator sistem

Migrasikan data - opsi 1

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan alat MySQL asli atau alat pihak ketiga untuk memigrasikan objek dan data database.	Untuk petunjuk, lihat dokumentasi untuk alat MySQL seperti mysqldbcopy dan mysqldump.	DBA

Migrasikan data - opsi 2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan data dengan AWS DMS.	Untuk petunjuk, lihat Menggunakan database yang kompatibel dengan MySQL sebagai sumber dan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Menggunakan database yang kompatibel dengan MySQL sebagai target dalam dokumentasi. AWS DMS	

Migrasikan aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti strateginya.	Ikuti strategi migrasi aplikasi.	DBA, Pemilik aplikasi, Administrator sistem

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Beralih klien aplikasi.	Alihkan klien aplikasi untuk terhubung ke titik akhir cluster Aurora yang baru.	DBA, Pemilik aplikasi, Administrator sistem

Tutup proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Matikan sumber daya.	Matikan AWS sumber daya sementara.	DBA, Administrator sistem
Tinjau dokumentasi.	Tinjau dan validasi dokumen proyek.	DBA, Pemilik aplikasi, Administrator sistem
Kumpulkan metrik.	Kumpulkan metrik di sekitar waktu untuk bermigrasi, persentase langkah manual versus penggunaan alat,	DBA, Pemilik aplikasi, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	penghematan biaya, dan sebagainya.	
Selesaikan proyek migrasi.	Tutup proyek dan berikan umpan balik.	Pemilik aplikasi, DBA, administrator Sistem

Sumber daya terkait

Referensi

- Migrasi data ke Amazon Aurora MySQL DB cluster
- AWS DMS situs web
- AWS DMS dokumentasi
- Harga Amazon Aurora
- Membuat dan menghubungkan ke cluster DB MySQL Aurora
- Amazon VPC dan Amazon RDS
- Dokumentasi Amazon Aurora

Tutorial dan video

- Memulai dengan AWS DMS
- Memulai dengan Amazon Aurora

Migrasikan database MySQL lokal ke Aurora MySQL menggunakan Percona, Amazon EFS, dan Amazon S3 XtraBackup

Rohan Jamadagni, Udayasimha Theepireddy, dan sajith menon, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasi database MySQL lokal yang besar secara efisien ke Amazon Aurora MySQL menggunakan Percona. XtraBackup Percona XtraBackup adalah utilitas cadangan open-source dan non-blocking untuk server berbasis MySQL. Pola ini menunjukkan cara menggunakan Amazon Elastic File System (Amazon EFS) untuk mengurangi waktu mengunggah cadangan ke Amazon Simple Storage Service (Amazon S3) dan mengembalikan cadangan ke Amazon Aurora MySQL. Pola ini juga memberikan rincian tentang cara membuat backup Percona tambahan untuk meminimalkan jumlah log biner yang akan diterapkan ke database MySQL Aurora target.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Izin untuk membuat peran dan kebijakan AWS Identity and Access Management (IAM)
- Konektivitas jaringan antara database MySQL lokal dan cloud pribadi virtual (VPC) di AWS

Batasan

- Server sumber harus sistem berbasis Linux yang dapat menginstal klien Network File System (NFS) (nfs-utils/nfs-common).
- Bucket S3 yang digunakan untuk mengunggah file cadangan hanya mendukung enkripsi sisi server (SSE-S3/SSE-KMS).
- Amazon S3 membatasi ukuran file cadangan hingga 5 TB. Jika file cadangan Anda melebihi 5 TB, Anda dapat membaginya menjadi beberapa file yang lebih kecil.
- Jumlah file sumber yang diunggah ke bucket S3 tidak boleh melebihi satu juta file.
- Pola ini mendukung Percona XtraBackup full backup dan incremental backup saja. Itu tidak mendukung cadangan sebagian yang menggunakan`--tables`,,,`--tables-exclude`, `--tables-file` `--databases``--databases-exclude`, atau. `--databases-file`

- Aurora tidak memulihkan pengguna, fungsi, prosedur tersimpan, atau informasi zona waktu dari database MySQL sumber.

Versi produk

- Database sumber harus MySQL versi 5.5, 5.6, atau 5.7.
- Untuk MySQL 5.7, Anda harus menggunakan Percona 2.4. XtraBackup
- Untuk MySQL 5.6 dan 5.6, Anda harus menggunakan Percona 2.3 atau 2.4. XtraBackup

Arsitektur

Tumpukan teknologi sumber

- Sistem operasi berbasis Linux
- Server MySQL
- Percona XtraBackup

Tumpukan teknologi target

- Amazon Aurora
- Amazon S3
- Amazon EFS

Arsitektur target

Alat

Layanan AWS

- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang membuatnya sederhana dan hemat biaya untuk menyiapkan, mengoperasikan, dan menskalakan penyebaran MySQL. Aurora MySQL adalah pengganti drop-in untuk MySQL.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di AWS Cloud.

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- Percona XtraBackup adalah utilitas sumber terbuka yang melakukan streaming, terkompresi, dan pencadangan tambahan dari database MySQL tanpa mengganggu atau memblokir database Anda.

Epik

Buat sistem file Amazon EFS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup keamanan untuk diasosiasikan dengan target pemasangan Amazon EFS.	Buat grup keamanan di VPC yang dikonfigurasi dengan lampiran VPN ke database lokal melalui AWS Transit Gateway. Untuk informasi selengkapnya tentang perintah dan langkah yang dijelaska n dalam cerita ini dan cerita lainnya, lihat tautan di bagian “Sumber daya terkait” di akhir pola ini.	DevOps/database Administr ator AWS
Edit aturan grup keamanan.	Tambahkan aturan masuk, menggunakan tipe NFS, port 2049, dan rentang IP server database lokal sebagai sumbernya. Secara default, aturan keluar memungkin kan semua lalu lintas untuk pergi. Jika ini tidak terjadi, tambahkan aturan keluar untuk membuka koneksi untuk	DevOps/database Administr ator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	port NFS. Tambahkan dua aturan masuk lagi: port 2049 (sumber: ID grup keamanan dari grup keamanan yang sama ini) dan port 22 (sumber: rentang IP dari mana Anda akan terhubung ke sebuah EC2 instance).	
Buat sistem file.	Di target pemasangan, gunakan VPC dan grup keamanan yang Anda buat di cerita sebelumnya. Pilih mode throughput dan kinerja berdasarkan I/O persyarat an database lokal. Secara opsional, aktifkan enkripsi saat istirahat.	DevOps/database Administr ator AWS

Pasang sistem berkas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran profil instans IAM untuk dikaitkan dengan sebuah EC2 instance.	Buat peran IAM yang memiliki izin untuk mengunggah dan mengakses objek di Amazon S3. Pilih bucket S3 tempat cadangan akan disimpan sebagai sumber kebijakan.	AWS DevOps
Buat sebuah EC2 instance.	Luncurkan EC2 instance berbasis Linux dan lampirkan peran profil instans IAM yang Anda buat pada langkah sebelumnya, dan grup	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	keamanan yang Anda buat sebelumnya.	
Instal klien NFS.	Instal klien NFS di server database lokal dan instans. EC2 Untuk petunjuk penginstalan, lihat bagian “Informasi tambahan”.	DevOps
Pasang sistem file Amazon EFS.	Pasang sistem file Amazon EFS di tempat dan di EC2 instans. Di setiap server, buat direktori untuk menyimpan cadangan, dan pasang sistem file dengan menggunakan titik akhir target mount. Sebagai contoh, lihat bagian “Informasi tambahan”.	DevOps

Buat cadangan database sumber MySQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal XtraBackup Percona.	Instal Percona XtraBackup 2.3 atau 2.4 (tergantung pada versi database MySQL Anda) di server database lokal. Untuk tautan instalasi, lihat bagian “Sumber daya terkait”.	Administrator basis data
Hitung skema dan tabel dalam database sumber.	Kumpulkan dan catat jumlah skema dan objek dalam database sumber MySQL. Anda akan menggunakan	Administrator basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	jumlah ini untuk memvalida si database Aurora MySQL setelah migrasi.	
(Opsional) Perhatikan urutan log biner terbaru dari database sumber.	Lakukan langkah ini jika Anda ingin membuat replikasi log biner antara database sumber dan Aurora MySQL untuk meminimalkan waktu henti. log-bin harus diaktifka n, dan server_id harus unik. Perhatikan urutan log biner saat ini dari database sumber, tepat sebelum memulai cadangan. Lakukan langkah ini sebelum pencadangan penuh jika Anda berencana hanya menggunakan cadangan penuh. Jika Anda berencana untuk membuat cadangan tambahan setelah pencadangan penuh, lakukan langkah ini tepat sebelum pencadangan tambahan akhir yang akan Anda pulihkan pada instance Aurora MySQL DB.	Administrator basis data
Mulai backup penuh dari sumber MySQL database.	Buat cadangan lengkap database sumber MySQL menggunakan Percona. XtraBackup Misalnya perintah untuk cadangan penuh dan tambahan, lihat bagian “Informasi tambahan”.	Administrator basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Buat cadangan tambahan menggunakan Percona. XtraBackup	Backup tambahan dapat digunakan untuk mengurangi jumlah log biner yang perlu Anda terapkan untuk menyinkronkan database sumber dengan Aurora MySQL. Database ukuran besar dan berat transaksi mungkin menghasilkan sejumlah besar log biner selama pencadangan. Dengan mengambil cadangan tambahan dan menyimpannya di sistem file Amazon EFS bersama, Anda dapat secara signifikan mengurangi waktu untuk mencadangkan dan mengunggah database Anda. Untuk detailnya, lihat bagian “Informasi tambahan”. Lanjutkan untuk membuat backup tambahan sampai Anda siap untuk memulai proses migrasi ke Aurora.	Administrator basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan cadangan.	Pada langkah ini, log transaksional diterapka n pada cadangan untuk transaksi yang sedang dalam penerbangan selama pencadangan. Terus menerapkan log transaksional (--apply-log-only) ke setiap cadangan tambahan untuk menggabungkan cadangan, kecuali untuk cadangan terakhir. Sebagai contoh, lihat bagian “Informasi tambahan” . <efs_mount_name>Setelah langkah ini, cadangan yang lengkap dan digabungkan akan berada di ~/ /fullbackup.	Administrator basis data
Zip dan pisahkan cadangan gabungan terakhir.	Setelah Anda menyiapka n cadangan terakhir yang digabungkan, gunakan perintah tar, zip, dan split untuk membuat file cadangan zip yang lebih kecil. Sebagai contoh, lihat bagian “Informasi tambahan”.	Administrator basis data

Kembalikan cadangan ke cluster DB MySQL Aurora

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah cadangan ke Amazon S3.	Sistem file Amazon EFS tempat file cadangan disimpan dipasang pada database	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	lokal dan EC2 instans, sehingga file cadangan sudah tersedia untuk instance. EC2 <efs_mount_name><b ucket_name>Hubungkan ke EC2 instance dengan menggunakan Secure Shell (SSH) dan unggah file cadangan zip ke bucket S3 baru atau yang sudah ada; misalnya: aws s3 sync ~/ / fullbackup s3:///fullbackup. Untuk detail tambahan, lihat tautan di bagian “Sumber daya terkait”.	
Buat peran layanan untuk Aurora untuk mengakses Amazon S3.	Buat peran IAM dengan kepercayaan “rds.amaz onaws.com” dan kebijakan yang memungkinkan Aurora mengakses bucket S3 tempat file cadangan disimpan. Izin yang diperlukan adalah ListBucket, GetObject, dan GetObjectVersion.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat konfigurasi jaringan untuk Aurora.	Buat grup subnet cluster DB dengan setidaknya dua Availability Zones dan konfigurasi tabel rute subnet yang memungkinkan konektivitas keluar ke database sumber. Buat grup keamanan yang memungkinkan koneksi keluar ke database lokal, dan memungkinkan administrator untuk terhubung ke klaster Aurora DB. Untuk informasi selengkapnya, lihat tautan di bagian “Sumber daya terkait”.	DevOps/database Administrator AWS
Kembalikan cadangan ke cluster DB MySQL Aurora.	Pulihkan data Anda dari cadangan yang Anda unggah ke Amazon S3. Tentukan versi MySQL dari database sumber Anda, berikan nama bucket S3 dan awalan jalur folder tempat Anda mengunggah file cadangan (misalnya, “fullback up” untuk contoh di bagian “Informasi tambahan”), dan berikan peran IAM yang Anda buat untuk mengotorisasi Aurora mengakses Amazon S3.	DevOps/database Administrator AWS
Validasi database Aurora MySQL.	Validasi jumlah skema dan objek di cluster Aurora DB yang dipulihkan terhadap hitungan yang Anda peroleh dari database sumber.	Administrator basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan replikasi binlog.	Gunakan urutan log biner yang Anda catat sebelumnya, sebelum membuat cadangan terakhir yang dikembali kan ke cluster Aurora DB. Buat pengguna replikasi pada basis data sumber, dan ikuti instruksi di bagian “Informasi tambahan” untuk memberikan hak istimewa yang sesuai, untuk mengaktif kan replikasi pada Aurora, dan untuk mengonfirmasi bahwa replikasi sinkron.	DevOps/database Administr ator AWS

Sumber daya terkait

Membuat sistem file Amazon EFS

- Membuat grup keamanan (dokumentasi Amazon VPC)
- Lampiran VPN gateway transit (dokumentasi Amazon VPC)
- Menskalakan throughput VPN menggunakan AWS Transit Gateway (blog Jaringan & Pengiriman Konten)
- Membuat sistem file Amazon EFS (dokumentasi Amazon EFS)
- Membuat target pemasangan (dokumentasi Amazon EFS)
- Mengenkripsi data saat istirahat (dokumentasi Amazon EFS)

Memasang sistem file

- Peran IAM untuk Amazon EC2 (EC2 dokumentasi Amazon)
- Meluncurkan instans Amazon EC2 Linux (EC2 dokumentasi Amazon)
- Menginstal klien NFS (dokumentasi Amazon EFS)
- Memasang sistem file Amazon EFS di klien lokal Anda (dokumentasi Amazon EFS)

- Memasang Sistem File EFS (dokumentasi Amazon EFS)

Membuat cadangan database sumber MySQL

- Instalasi Percona XtraBackup 2.3 (dokumentasi XtraBackup Percona)
- Instalasi Percona XtraBackup 2.4 (dokumentasi XtraBackup Percona)
- Mengatur konfigurasi master replikasi (dokumentasi MySQL)
- Migrasi data dari database MySQL eksternal ke cluster DB MySQL Aurora (dokumentasi Aurora)
- Pencadangan tambahan (dokumentasi XtraBackup Percona)

Memulihkan cadangan ke Amazon Aurora MySQL

- Membuat ember (dokumentasi Amazon S3)
- Menyambung ke instans Linux Anda menggunakan SSH (dokumentasi Amazon Ec2)
- Mengkonfigurasi AWS CLI (dokumentasi AWS CLI)
- perintah sinkronisasi (referensi perintah AWS CLI)
- Membuat kebijakan IAM untuk mengakses sumber daya Amazon S3 (dokumentasi Aurora)
- Prasyarat cluster DB (dokumentasi Aurora)
- Bekerja dengan grup subnet DB (dokumentasi Aurora)
- Membuat grup keamanan VPC untuk instans DB pribadi (dokumentasi Aurora)
- Memulihkan cluster DB MySQL Aurora dari bucket S3 (dokumentasi Aurora)
- Menyiapkan replikasi dengan MySQL atau cluster Aurora DB lainnya (dokumentasi Aurora)
- mysql.rds_set_external_master prosedur (MySQL pada referensi Amazon RDS SQL)
- mysql.rds_start_replication procedure (MySQL pada referensi Amazon RDS SQL)

Referensi tambahan

- Migrasi data dari database MySQL eksternal ke cluster DB MySQL Aurora (dokumentasi Aurora)
- Unduhan server MySQL (situs web Oracle)

Tutorial dan video

- Migrasi data MySQL ke cluster DB MySQL Aurora menggunakan Amazon S3 (AWS Knowledge Center)
- Penyiapan dan pemasangan Amazon EFS (video)

Informasi tambahan

Menginstal klien NFS

- Jika Anda menggunakan Red Hat atau sistem operasi Linux serupa, gunakan perintah:

```
$ sudo yum -y install nfs-utils
```

- Jika Anda menggunakan Ubuntu atau sistem operasi Linux serupa, gunakan perintah:

```
$ sudo apt-get -y install nfs-common
```

Untuk informasi selengkapnya, lihat panduan di dokumentasi Amazon EFS.

Memasang sistem file Amazon EFS

Gunakan perintah:

```
mkdir ~/<efs_mount_name>
$ sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-
target-IP:/ ~/<efs_mount_name>
```

Untuk informasi selengkapnya, lihat panduan dan Pemasangan Sistem File EFS di dokumentasi Amazon EFS.

Membuat backup dari database sumber MySQL

Cadangan penuh

Gunakan perintah seperti berikut ini, yang mengambil cadangan, ritsleting, dan membaginya menjadi potongan-potongan kecil masing-masing 1 GB:

```
xtrabackup --backup --user=dbuser --password=<password> --binlog-info=AUTO --stream=tar
 --target-dir=~/<efs_mount_name>/fullbackup | gzip - | split -d --bytes=1024MB - ~/
<efs_mount_name>/fullbackup/backup.tar.gz &
```

Jika Anda berencana untuk membuat cadangan tambahan berikutnya setelah pencadangan penuh, jangan zip dan pisahkan cadangan. Sebagai gantinya, gunakan perintah yang mirip dengan berikut ini:

```
xtrabackup --backup --user=dbuser --password=<password> --target-dir=~/
<efs_mount_name>/fullbackup/
```

Pencadangan tambahan

Gunakan jalur cadangan lengkap untuk `--incremental-basedir` parameter; misalnya:

```
xtrabackup --backup --user=dbuser --password=<password> --target-dir=~/
<efs_mount_name>/incremental/backupdate --incremental-basedir=~/<efs_mount_name>/
fullbackup
```

di mana basedir adalah jalur ke cadangan lengkap dan file xtrabackup_checkpoints.

Untuk informasi selengkapnya tentang membuat cadangan, lihat Memigrasi Data dari Database MySQL Eksternal ke Cluster DB MySQL Amazon Aurora dalam dokumentasi Aurora.

Mempersiapkan cadangan

Untuk menyiapkan cadangan lengkap:

```
xtrabackup --prepare --apply-log-only --target-dir=~/<efs_mount_name>/fullbackup
```

Untuk menyiapkan cadangan tambahan:

```
xtrabackup --prepare --apply-log-only --target-dir=~/<efs_mount_name>/fullbackup --
incremental-dir=~/<efs_mount_name>/incremental/06062020
```

Untuk menyiapkan cadangan akhir:

```
xtrabackup --prepare --target-dir=~/<efs_mount_name>/fullbackup --incremental-dir=~/
<efs_mount_name>/incremental/06072020
```

Untuk informasi selengkapnya, lihat Pencadangan tambahan di dokumentasi Percona. XtraBackup

Ritsleting dan pemisahan cadangan yang digabungkan

<efs_mount_name>Untuk zip cadangan gabungan di ~/ /fullbackup:

```
tar -zcvf <backupfilename.tar.gz> ~/<efs_mount_name>/fullbackup
```

Untuk membagi cadangan:

```
split -d -b1024M --verbose  <backupfilename.tar.gz> <backupfilename.tar.gz>
```

Menyiapkan replikasi binlog

Untuk membuat pengguna replikasi pada database sumber dan memberikan hak istimewa yang sesuai:

```
CREATE USER 'repl_user'@'' IDENTIFIED BY ''; GRANT REPLICATION CLIENT, REPLICATION
 SLAVE ON *.* TO 'repl_user'@'';
```

Untuk mengaktifkan replikasi pada Aurora dengan menghubungkan ke cluster Aurora DB, aktifkan log biner di grup parameter cluster DB. Set `binlog_format = mixed` (mode campuran lebih disukai). Perubahan ini mengharuskan Anda memulai ulang instance untuk menerapkan pembaruan.

```
CALL mysql.rds_set_external_master ('sourcedbinstanceIP', sourcedbport, 'repl_user',
 '', 'binlog_file_name', binlog_file_position, 0); CALL mysql.rds_start_replication;
```

Untuk mengonfirmasi bahwa replikasi sinkron:

```
SHOW Slave Status \G;
```

Bidang Detik di belakang master menunjukkan seberapa jauh di belakang Aurora dari database lokal.

Migrasikan aplikasi Java lokal ke AWS menggunakan AWS App2Container

Dhananjay Karanjkar, Amazon Web Services

Ringkasan

AWS App2Container (A2C) adalah alat baris perintah yang membantu mengubah aplikasi yang ada yang berjalan di mesin virtual menjadi wadah, tanpa perlu perubahan kode apa pun. A2C menemukan aplikasi yang berjalan di server, mengidentifikasi dependensi, dan menghasilkan artefak yang relevan untuk penyebaran tanpa batas ke Amazon Elastic Container Service (Amazon ECS) Container Service (Amazon ECS) dan Amazon Elastic Kubernetes Service (Amazon EKS).

Pola ini menyediakan langkah-langkah untuk memigrasikan aplikasi Java lokal dari jarak jauh yang diterapkan pada server aplikasi ke AWS Fargate atau Amazon EKS dengan menggunakan App2Container melalui mesin pekerja.

Mesin pekerja dapat digunakan dalam kasus penggunaan berikut:

- Instalasi Docker tidak diperbolehkan atau tidak tersedia di server aplikasi tempat aplikasi Java berjalan.
- Anda harus mengelola migrasi beberapa aplikasi yang digunakan pada server fisik atau virtual yang berbeda.

Pola ini menggunakan AWS CodeCommit AWS CodePipeline,, dan AWS CodeBuild.

Prasyarat dan batasan

Prasyarat

- Server aplikasi dengan aplikasi Java yang berjalan di server Linux
- Mesin pekerja dengan sistem operasi Linux
- Mesin pekerja dengan setidaknya 20 GB ruang disk yang tersedia

Batasan

- Tidak semua aplikasi didukung. Untuk informasi selengkapnya, lihat Aplikasi yang didukung untuk Linux.

Arsitektur

Tumpukan teknologi sumber

- Aplikasi Java berjalan di server Linux

Tumpukan teknologi target

- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- AWS CodePipeline
- Amazon Elastic Container Registry
- AWS Fargate

Arsitektur target

Alat

Alat

- AWS App2Container — AWS App2Container (A2C) adalah alat baris perintah untuk membantu Anda mengangkat dan memindahkan aplikasi yang berjalan di pusat data lokal atau di mesin virtual, sehingga mereka berjalan dalam wadah yang dikelola oleh Amazon ECS atau Amazon EKS.
- AWS CodeBuild — AWS CodeBuild adalah layanan build yang dikelola sepenuhnya di cloud. CodeBuild mengkompilasi kode sumber Anda, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommit — AWS CodeCommit adalah layanan kontrol versi yang dihosting oleh Amazon Web Services yang dapat Anda gunakan untuk menyimpan dan mengelola aset secara pribadi (seperti dokumen, kode sumber, dan file biner) di cloud.
- AWS CodePipeline — AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perangkat lunak Anda.

- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat untuk menjalankan, menghentikan, dan mengelola kontainer di cluster.
- Amazon ECR — Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola AWS yang aman, terukur, dan andal.
- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) adalah layanan terkelola yang dapat Anda gunakan untuk menjalankan Kubernetes di AWS tanpa perlu menginstal, mengoperasikan, dan memelihara control plane atau node Kubernetes Anda sendiri.
- AWS Fargate — AWS Fargate adalah teknologi yang dapat Anda gunakan dengan Amazon ECS untuk menjalankan container tanpa harus mengelola server atau cluster instans Amazon Elastic Compute Cloud (Amazon). EC2 Dengan Fargate, Anda tidak perlu menyediakan, mengonfigurasi, atau menskalakan klaster mesin virtual untuk menjalankan kontainer.

Epik

Siapkan kredensil

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat rahasia untuk mengakses server aplikasi.	Untuk mengakses server aplikasi dari jarak jauh dari mesin pekerja, buat rahasia di AWS Secrets Manager. Untuk rahasia Anda, Anda dapat menggunakan kunci pribadi SSH atau Sertifikat dan kunci pribadi SSH. Untuk informasi selengkapnya, lihat Mengelola rahasia untuk AWS App2Container.	DevOps, Pengembang

Siapkan mesin pekerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal file tar.	Jalankan `sudo yum install -y tar`.	DevOps, Pengembang
Instal AWS CLI.	Untuk menginstal Amazon Command Line Interface (AWS CLI), jalankan. `curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"` Buka ritsletingan`awscliv2.zip .` Jalankan `sudo ./aws/install .`	DevOps, Pengembang

Instal App2Container.	Jalankan perintah berikut:
	
	`curl -o AWSApp2Container-installer-linux.tar.gz https://app2container-release-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/AWSApp2Container-installer-linux.tar.gz`
	
	`sudo tar xvf AWSApp2Container-installer-linux.tar.gz`
		DevOps, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`sudo ./install.sh`	
Konfigurasikan profil.	Untuk mengonfigurasi profil default AWS, jalankan`sudo aws configure .` Untuk mengonfigurasi profil default AWS bernama, jalankan`sudo aws configure --profile <profile name>`.	DevOps, Pengembang

Pasang Docker.	Jalankan perintah berikut.
	
	`sudo yum install -y docker`
	
	`sudo systemctl enable docker & sudo systemctl restart docker`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi App2Container.	Untuk menginisialisasi App2Container, Anda memerlukan informasi berikut: • `workspace` : Untuk menyimpan artefak kontainerisasi aplikasi. Kami merekomendasikan menyediakan jalur direktori yang memiliki setidaknya 20 GB ruang disk kosong. • `awsProfile` : Profil AWS dikonfigurasi di server. Ini diperlukan untuk mengunggah artefak ke Amazon S3, menjalankan `containerize` perintah, dan menghasilkan artefak AWS untuk penerapan di Amazon ECS atau Amazon EKS. • `s3Bucket`: Untuk mengekstrak dan menyimpan artefak AWS. • `metricsReportPermission` : Untuk mengumpulkan dan menyimpan metrik yang dilaporkan. • `dockerContentTrust` : Untuk menandatangani gambar Docker.	DevOps, Pengembang

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jalankan `sudo app2container init` .	

Konfigurasikan mesin pekerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan mesin pekerja untuk menghubungkan dan menjalankan perintah App2Container dari jarak jauh di server aplikasi.	Untuk mengkonfigurasi mesin pekerja, informasi berikut diperlukan: • `Server FQDN`: Nama domain yang sepenuhnya memenuhi syarat dari server aplikasi. • `Server IP address`: Alamat IP dari server aplikasi. Baik FQDN atau alamat IP sudah cukup. • `SecretARN` Nama Sumber Daya Amazon (ARN) dari rahasia yang digunakan untuk terhubung ke server aplikasi dan disimpan di Secrets Manager. • `AuthMethod` : Metode `key` atau `cert` otentikasi. Jalankan `sudo app2container remote configure` .	DevOps, Pengembang

Temukan, analisis, dan ekstrak aplikasi pada mesin pekerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Temukan aplikasi Java lokal.	Untuk menemukan semua aplikasi yang berjalan di server aplikasi dari jarak jauh, jalankan perintah berikut. `sudo app2container remote inventory --target <FQDN/IP of App server>` Perintah ini menghasilkan daftar aplikasi yang digunakan di`inventory.json` .	Pengembang, DevOps
Analisis aplikasi yang ditemukan.	Untuk menganalisis setiap aplikasi dari jarak jauh dengan menggunakan id aplikasi yang diperoleh pada tahap inventaris, jalankan perintah berikut. `sudo app2container remote analyze --application-id <java-app-id> --target <FQDN/IP of App Server>` Ini menghasilkan `analysis.json` file di lokasi ruang kerja. Setelah file ini dibuat, Anda dapat mengubah parameter kontainerisasi berdasarkan kebutuhan Anda.	Pengembang, DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak aplikasi yang dianalisis.	Untuk menghasilkan arsip aplikasi untuk aplikasi yang dianalisis, jalankan perintah berikut dari jarak jauh, yang akan menghasilkan bundel tar di lokasi ruang kerja. `sudo app2container remote extract --application-id <application id> --target <FQDN/IP of App Server>` Artefak yang diekstraksi dapat dihasilkan pada mesin pekerja lokal.	Pengembang, DevOps

Kontainerisasi artefak yang diekstraksi pada mesin pekerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kontainerisasi artefak yang diekstraksi.	Kontainerisasi artefak yang diekstraksi pada langkah sebelumnya dengan menjalankan perintah berikut. `sudo app2container containerize --input-archive <tar bundle location on worker machine>`	Pengembang, DevOps
Selesaikan target.	Untuk menyelesaikan target, buka`deployment.json` ,	Pengembang, DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang dibuat saat `containerize` perintah berjalan. Untuk menentukan AWS Fargate sebagai target, atur `createEcsArtifacts` ke. `true` Untuk menetapkan Amazon EKS sebagai target, atur `createEksArtifacts` ke true.	

Menghasilkan dan menyediakan artefak AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan artefak penerapan AWS di mesin pekerja.	Untuk menghasilkan artefak penyebaran, jalankan perintah berikut. `sudo app2container generate app-deployment --application-id <application id>` Ini menghasilkan CloudFormation template `ecs-master.yml` AWS di ruang kerja.	DevOps
Menyediakan artefak.	Untuk menyediakan lebih lanjut artefak yang dihasilkan, terapkan CloudFormation template AWS dengan menjalankan perintah berikut. `aws cloudformation deploy --template-`	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`file <path to ecs-master.yml> --capabilities CAPABILITY_NAMED_IAM --stack-name <application id>-ECS`	
Hasilkan pipa.	Modifikasi`pipeline.json` , yang dibuat di cerita sebelumnya, berdasarkan kebutuhan Anda. Kemudian jalankan `generate pipeline` perintah untuk menghasilkan artefak penyebaran pipeline.	DevOps

Sumber daya terkait

- Apa itu App2Container?
- Posting blog AWS App2Container
- Dasar-dasar konfigurasi AWS CLI
- Dasar-dasar Docker untuk Amazon ECS
- Perintah Docker

Migrasi sistem file bersama dalam migrasi AWS besar

Amit Rudraraju, Sam Apa, Bheemeswararao Balla, Wally Lu, dan Sanjeev Prakasam, Amazon Web Services

Ringkasan

Migrasi 300 atau lebih server dianggap sebagai migrasi besar. Tujuan migrasi besar adalah untuk memigrasikan beban kerja dari pusat data lokal yang ada ke pusat data lokal AWS Cloud, dan proyek ini biasanya berfokus pada beban kerja aplikasi dan database. Namun, sistem file bersama memerlukan perhatian terfokus dan rencana migrasi terpisah. Pola ini menjelaskan proses migrasi untuk sistem file bersama dan memberikan praktik terbaik untuk memigrasikannya dengan sukses sebagai bagian dari proyek migrasi besar.

Sistem file bersama (SFS), juga dikenal sebagai jaringan atau sistem file berkerumun, adalah berbagi file yang dipasang ke beberapa server. Sistem file bersama diakses melalui protokol seperti Network File System (NFS), Common Internet File System (CIFS), atau Server Message Block (SMB).

Sistem ini tidak dimigrasikan dengan alat migrasi standar seperti AWS Application Migration Service karena tidak didedikasikan untuk host yang dimigrasi atau direpresentasikan sebagai perangkat blok. Meskipun sebagian besar dependensi host dimigrasikan secara transparan, koordinasi dan pengelolaan sistem file dependen harus ditangani secara terpisah.

Anda memigrasi sistem file bersama dalam fase berikut: temukan, rencanakan, siapkan, potong, dan validasi. Dengan menggunakan pola ini dan buku kerja terlampir, Anda memigrasikan sistem file bersama ke layanan AWS penyimpanan, seperti Amazon Elastic File System (Amazon EFS), Amazon FSx untuk NetApp ONTAP, atau Amazon FSx untuk Windows File Server. Untuk mentransfer sistem file, Anda dapat menggunakan AWS DataSync atau alat pihak ketiga, seperti NetApp SnapMirror.

> **Note**
>
> Pola ini adalah bagian dari seri Panduan AWS Preskriptif tentang migrasi besar ke. AWS Cloud Pola ini mencakup praktik terbaik dan instruksi untuk memasukkan SFSs ke dalam paket gelombang Anda untuk server. Jika Anda memigrasikan satu atau beberapa sistem file bersama di luar proyek migrasi besar, lihat petunjuk transfer data dalam AWS dokumentasi untuk Amazon EFS, Amazon FSx untuk Windows File Server, dan Amazon FSx untuk NetApp ONTAP.

Prasyarat dan batasan

Prasyarat

Prasyarat dapat bervariasi tergantung pada sumber dan target sistem file bersama dan kasus penggunaan Anda. Berikut ini adalah yang paling umum:

- Aktif Akun AWS.
- Anda telah menyelesaikan penemuan portofolio aplikasi untuk proyek migrasi besar Anda dan mulai mengembangkan rencana gelombang. Untuk informasi selengkapnya, lihat Buku pedoman portofolio untuk migrasi AWS besar.
- Virtual private cloud (VPCs) dan grup keamanan yang memungkinkan lalu lintas masuk dan keluar antara pusat data lokal dan lingkungan Anda. AWS Untuk informasi selengkapnya, lihat Opsi konektivitas VPC Network-to Amazon dan persyaratan jaringan.AWS DataSync
- Izin untuk membuat AWS CloudFormation tumpukan atau izin untuk membuat Amazon EFS atau sumber daya Amazon. FSx Untuk informasi selengkapnya, lihat CloudFormation dokumentasi, dokumentasi Amazon EFS, atau FSx dokumentasi Amazon.
- Jika Anda menggunakan AWS DataSync untuk melakukan migrasi, Anda memerlukan izin berikut:
  - Izin AWS DataSync untuk mengirim log ke grup CloudWatch log Amazon Logs. Untuk informasi selengkapnya, lihat DataSync Mengizinkan mengunggah log ke grup CloudWatch log.
  - Izin untuk mengakses grup CloudWatch log Log. Untuk informasi selengkapnya, lihat Ringkasan mengelola izin akses ke sumber daya CloudWatch Log Anda.
  - Izin untuk membuat agen dan tugas di DataSync. Untuk informasi selengkapnya, lihat Izin IAM yang diperlukan untuk menggunakan. AWS DataSync

Batasan

- Pola ini dirancang untuk bermigrasi SFSs sebagai bagian dari proyek migrasi besar. Ini mencakup praktik dan instruksi terbaik untuk memasukkan SFSs ke dalam paket gelombang Anda untuk migrasi aplikasi. Jika Anda memigrasikan satu atau beberapa sistem file bersama di luar proyek migrasi besar, lihat petunjuk transfer data dalam AWS dokumentasi untuk Amazon EFS, Amazon FSx untuk Windows File Server, dan Amazon FSx untuk NetApp ONTAP.
- Pola ini didasarkan pada arsitektur, layanan, dan pola migrasi yang umum digunakan. Namun, proyek dan strategi migrasi besar dapat bervariasi antar organisasi. Anda mungkin perlu menyesuaikan solusi ini atau buku kerja yang disediakan berdasarkan kebutuhan Anda.

Arsitektur

Tumpukan teknologi sumber

Satu atau lebih dari berikut ini:

- Server file Linux (NFS)
- Server file Windows (SMB)
- NetApp array penyimpanan
- Larik penyimpanan Dell EMC Isilon

Tumpukan teknologi target

Satu atau lebih dari berikut ini:

- Sistem File Elastis Amazon
- Amazon FSx untuk NetApp ONTAP
- Amazon FSx untuk Server File Windows

Arsitektur target

Diagram menunjukkan proses berikut:

1. Anda membuat sambungan antara pusat data lokal dan AWS Cloud dengan menggunakan Layanan AWS seperti AWS Direct Connect atau AWS Site-to-Site VPN.
2. Anda menginstal DataSync agen di pusat data lokal.
3. Menurut rencana gelombang Anda, Anda gunakan DataSync untuk mereplikasi data dari sistem file bersama sumber ke berbagi AWS file target.

Fase migrasi

Gambar berikut menunjukkan fase dan langkah-langkah tingkat tinggi untuk memigrasikan SFS dalam proyek migrasi besar.

Bagian Epik dari pola ini berisi petunjuk terperinci tentang cara menyelesaikan migrasi dan menggunakan buku kerja terlampir. Berikut ini adalah ikhtisar tingkat tinggi dari langkah-langkah dalam pendekatan bertahap ini.

Fase	Langkah-langkah
Temukan	1. Dengan menggunakan alat penemuan, Anda mengumpulkan data tentang sistem file bersama, termasuk server, titik pemasangan, dan alamat IP. 2. Menggunakan database manajemen konfigurasi (CMDB) atau alat migrasi Anda, Anda mengumpulkan detail tentang server, termasuk informasi tentang gelombang migrasi, lingkungan, pemilik aplikasi, nama layanan manajemen layanan TI (ITSM), unit organisasi, dan ID aplikasi.
Rencana	3. Menggunakan informasi yang dikumpulk an tentang SFSs dan server, buat rencana gelombang SFS. 4. Menggunakan informasi di lembar kerja build, untuk setiap SFS, pilih target Layanan AWS dan alat migrasi.
Persiapkan	5. Siapkan infrastruktur target di Amazon EFS, Amazon FSx untuk NetApp ONTAP, atau Amazon FSx untuk Windows File Server. 6. Siapkan layanan transfer data, seperti DataSync, dan kemudian mulai sinkronisasi data awal. Saat sinkronisasi awal selesai, Anda dapat mengatur sinkronisasi berulang untuk dijalankan sesuai jadwal,

	7. Perbarui rencana gelombang SFS dengan informasi tentang berbagi file target, seperti alamat IP atau jalur.
Potong	8. Hentikan aplikasi yang secara aktif mengakses sumber SFS. 9. Dalam layanan transfer data, lakukan sinkronisasi data akhir. 10. Ketika sinkronisasi selesai, validasi bahwa itu benar-benar berhasil dengan meninjau data log di CloudWatch Log.
Validasi	11. Di server, ubah titik pemasangan ke jalur SFS baru. 12. Mulai ulang dan validasi aplikasi.

Alat

Layanan AWS

- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- AWS DataSyncadalah layanan transfer dan penemuan data online yang membantu Anda memindahkan file atau data objek ke, dari, dan di antara layanan AWS penyimpanan.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di file AWS Cloud.
- Amazon FSx menyediakan sistem file yang mendukung protokol konektivitas standar industri dan menawarkan ketersediaan dan replikasi yang tinggi. Wilayah AWS

Alat lainnya

- SnapMirroradalah alat replikasi NetApp data yang mereplikasi data dari volume sumber tertentu atau qtrees ke volume target atau qtrees, masing-masing. Anda dapat menggunakan alat ini untuk memigrasikan sistem file NetApp sumber ke Amazon FSx untuk NetApp ONTAP.

- Robocopy, yang merupakan kependekan dari Robust File Copy, adalah direktori baris perintah dan perintah untuk Windows. Anda dapat menggunakan alat ini untuk memigrasikan sistem file sumber Windows ke Amazon FSx untuk Windows File Server.

Praktik terbaik

Pendekatan perencanaan gelombang

Saat merencanakan gelombang untuk proyek migrasi besar Anda, pertimbangkan latensi dan kinerja aplikasi. Ketika SFS dan aplikasi dependen beroperasi di lokasi yang berbeda, seperti satu di cloud dan satu di pusat data lokal, ini dapat meningkatkan latensi dan memengaruhi kinerja aplikasi. Berikut ini adalah opsi yang tersedia saat membuat rencana gelombang:

1. Migrasikan SFS dan semua server dependen dalam gelombang yang sama — Pendekatan ini mencegah masalah kinerja dan meminimalkan pengerjaan ulang, seperti mengonfigurasi ulang titik pemasangan beberapa kali. Disarankan ketika latensi sangat rendah diperlukan antara aplikasi dan SFS. Namun, perencanaan gelombang rumit, dan tujuannya biasanya untuk menghapus variabel dari pengelompokan ketergantungan, bukan menambahkannya. Selain itu, pendekatan ini tidak disarankan jika banyak server mengakses SFS yang sama karena membuat gelombang terlalu besar.
2. Migrasikan SFS setelah server dependen terakhir dimigrasi — Misalnya, jika SFS diakses oleh beberapa server dan server tersebut dijadwalkan untuk bermigrasi dalam gelombang 4, 6, dan 7, jadwalkan SFS untuk bermigrasi dalam gelombang 7.

   Pendekatan ini sering kali paling logis untuk migrasi besar dan direkomendasikan untuk aplikasi yang sensitif terhadap latensi. Ini mengurangi biaya yang terkait dengan transfer data. Ini juga meminimalkan periode latensi antara SFS dan aplikasi tingkat tinggi (seperti produksi) karena aplikasi tingkat yang lebih tinggi biasanya dijadwalkan untuk bermigrasi terakhir, setelah pengembangan dan aplikasi QA.

   Namun, pendekatan ini masih membutuhkan penemuan, perencanaan, dan kelincahan. Anda mungkin perlu memigrasikan SFS dalam gelombang sebelumnya. Konfirmasikan bahwa aplikasi dapat menahan latensi tambahan untuk periode waktu antara gelombang dependen pertama dan gelombang yang mengandung SFS. Lakukan sesi penemuan dengan pemilik aplikasi dan migrasi aplikasi dalam gelombang yang sama aplikasi yang paling sensitif terhadap latensi. Jika masalah kinerja ditemukan setelah memigrasikan aplikasi dependen, bersiaplah untuk berputar dengan cepat untuk memigrasikan SFS secepat mungkin.

3. Migrasikan SFS di akhir proyek migrasi besar — Pendekatan ini direkomendasikan jika latensi bukan faktor, seperti ketika data di SFS jarang diakses atau tidak penting untuk kinerja aplikasi. Pendekatan ini merampingkan migrasi dan menyederhanakan tugas cutover.

Anda dapat memadukan pendekatan ini berdasarkan sensitivitas latensi aplikasi. Misalnya, Anda dapat memigrasikan latensi-sensitif SFSs dengan menggunakan pendekatan 1 atau 2 dan kemudian memigrasikan sisanya dengan menggunakan pendekatan 3. SFSs

Memilih layanan sistem AWS file

AWS menawarkan beberapa layanan cloud untuk penyimpanan file. Masing-masing menawarkan manfaat dan batasan yang berbeda untuk kinerja, skala, aksesibilitas, integrasi, kepatuhan, dan pengoptimalan biaya. Ada beberapa opsi default logis. Misalnya, jika sistem file lokal Anda saat ini mengoperasikan Windows Server, maka Amazon FSx untuk Windows File Server adalah pilihan default. Atau jika sistem file lokal mengoperasikan NetApp ONTAP, Amazon FSx untuk NetApp ONTAP adalah pilihan default. Namun, Anda dapat memilih layanan target berdasarkan persyaratan aplikasi Anda atau untuk mewujudkan manfaat operasi cloud lainnya. Untuk informasi selengkapnya, lihat Memilih layanan penyimpanan AWS file yang tepat untuk penerapan Anda (presentasi AWS Summit).

Memilih alat migrasi

Amazon EFS dan Amazon FSx mendukung penggunaan AWS DataSync untuk memigrasikan sistem file bersama ke file. AWS Cloud Untuk informasi selengkapnya tentang sistem dan layanan penyimpanan yang didukung, manfaat, dan kasus penggunaan, lihat Apa itu AWS DataSync. Untuk gambaran umum tentang proses penggunaan DataSync untuk mentransfer file Anda, lihat Cara kerja AWS DataSync transfer.

Ada juga beberapa alat pihak ketiga yang tersedia, termasuk yang berikut ini:

- Jika memilih Amazon FSx untuk NetApp ONTAP, Anda dapat menggunakannya NetApp SnapMirror untuk memigrasikan file dari pusat data lokal ke cloud. SnapMirror menggunakan replikasi tingkat blok, yang bisa lebih cepat dari DataSync dan mengurangi durasi proses transfer data. Untuk informasi selengkapnya, lihat Memigrasi ke FSx untuk ONTAP menggunakan. NetApp SnapMirror
- Jika Anda memilih Amazon FSx untuk Windows File Server, Anda dapat menggunakan Robocopy untuk memigrasi file ke cloud. Untuk informasi selengkapnya, lihat Memigrasi file yang ada ke FSx Windows File Server menggunakan Robocopy.

Epik

Temukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan buku kerja penemuan SFS.	1. Unduh buku kerja di bagian Lampiran dari pola ini. Ini berisi dua file, SFS-Discovery-Workbook.xlsx dan SFS-Wave-Plan-Workbook.xlsx. 2. Buka file SFS-Discovery-Workbook di Microsoft Excel. 3. Pada lembar kerja Dashboard, lakukan hal berikut: • Di kolom A, perbarui nama lingkungan. • Di kolom B, perbarui urutan lingkungan untuk menempatkannya dalam urutan prioritas terendah (1) ke prioritas tertinggi. • Di kolom D—E, perbarui jadwal gelombang. • Di kolom C dan K, perbarui nama akun AWS. • Di kolom L, perbarui VPC IDs. • Di kolom M—O, perbarui subnet. IDs	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Tinjau template buku kerja lainnya dan perbarui nilai lain yang diperlukan untuk organisasi atau kasus penggunaan Anda. 5. Simpan buku kerja.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan informasi tentang sumber SFS.	1. Dengan menggunakan alat penemuan pilihan Anda, identifikasi semua pemasangan SFS di semua perangkat penyimpan an yang berlaku, server Linux, dan server Windows. Biasanya, Anda perlu mengumpulkan informasi berikut: • Perangkat klien • Alamat IP klien • Rincian SFS • Titik gunung **Note** Anda dapat menambahk an detail titik pemasangan ke buku runbook migrasi untuk mengatur ulang SFS setelah migrasi. 2. Buka file SFS-Discovery-Workbook. 3. Pada lembar kerja Lembar Gelombang, lakukan hal berikut: • Di kolom Lokasi Server (D), dalam rumus,	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	konfirmasikan bahwa format rentang CIDR untuk sumber lokal berfungsi untuk rentang Anda. Misalnya, jika rentang CIDR Anda`10.0.0.0/8` , masukkan`10.*.*.*`. • Di kolom lokasi SFS (E), dalam rumus, konfirmasikan bahwa format rentang CIDR untuk VPC target berfungsi untuk rentang Anda. Misalnya, jika rentang CIDR Anda`176.16.0.0/16` , masukkan`176.16.*.*` . 4. Pada lembar kerja SFS-Data, lakukan hal berikut: • Di kolom Nama Server (A), masukkan nama server tempat SFS dipasang. • Di kolom jalur SFS (B), masukkan nama SFS. • Di kolom alamat IP (C), masukkan alamat IP server. • Tambahkan informasi relevan lainnya yang Anda kumpulkan selama penemuan, seperti titik pemasangan dan ukuran SFS. Anda dapat	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggunakan data ini nanti untuk memodifikasi perhitungan perencanaan gelombang. 5. Simpan buku kerja.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan informasi tentang server.	1. Dengan menggunakan CMDB atau data yang direkam di alat migrasi, identifikasi semua informasi berikut tentang server yang memiliki mount SFS: • Nama Server • Alamat IP • Gelombang • Unit organisasi (OU) • Lingkungan server, sepertiDEV,QA, atau PROD • Nama aplikasi • Pemilik aplikasi dan informasi kontak 2. Buka file SFS-Discovery-Workbook. 3. Pada lembar kerja Server-Data, di kolom A-H, masukkan informasi yang Anda kumpulkan tentang server sumber. Perhatikan hal-hal berikut: • Di kolom Gelombang # (C), masukkan nama gelombang (sepertiWave1), out-of-scope (OOS), atauRetire. • Jika kolom kontak pemilik Aplikasi (H), verifikasi alamat email sudah benar. Alamat email ini	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dibuat secara otomatis berdasarkan nama yang Anda berikan di kolom Pemilik Aplikasi (G). Jika perlu, perbarui nilai secara manual untuk mencerminkan alamat email yang benar. • Jangan memodifikasi kolom I—J, yang berisi rumus. 4. Simpan buku kerja.	

Rencana

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bangun rencana gelombang SFS.	1. Buka file SFS-Discovery-Workbook. 2. Verifikasi semua informasi yang dikumpulkan dalam fase penemuan akurat dan terkini. 3. Pada lembar kerja Lembar Gelombang, filter kolom gelombang SFS (K) pada nilainya. 1 Ini adalah daftar semua SFSs di gelombang pertama. **Note** Nilai 0 dalam kolom ini menunjukkan	Membangun lead, Cutover lead, Insinyur migrasi, Pimpinan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bahwa SFS berada di luar cakupan migrasi. Ini mungkin karena SFS sudah di-host AWS atau karena server yang mengakses share berada di luar cakupan migrasi. 4. Verifikasi bahwa Anda ingin memigrasikan ini SFSs dalam gelombang ini. Untuk informasi selengkap nya tentang SFSs cara menetapkan gelombang, lihat Pendekatan perencana an gelombang di bagian Praktik Terbaik. 5. Pilih dan salin sel yang berisi nilai yang difilter. Jangan menyalin baris header yang berisi judul kolom. 6. Buka file SFS-Wave-Plan-Workbook yang sebelumny a Anda unduh. 7. Pada lembar kerja Ekspor-dari-penemuan, pilih sel A2. 8. Tempel data yang disalin. 9. Simpan file SFS-Discovery-Workbook dan SFS-Wave-Plan-Workbook.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih alat target Layanan AWS dan migrasi.	1. Dalam SFS-Wave-Plan-Workbook file, pada Exported-from-Discovery lembar kerja, pilih dan salin nilai di kolom Old path (C). 2. Pada lembar kerja Build-Wave, pilih sel A2. 3. Tempel data yang disalin. Kolom B—M di lembar kerja ini secara otomatis diperbarui untuk mencermin kan data lain yang terkait dengan jalur ini. 4. Hapus nilai duplikat apa pun di kolom A. Untuk petunjuk, lihat Menghapus nilai duplikat (situs web Microsoft Support). 5. Di kolom Target pattern atau service (F), tinjau target yang direkomen dasikan Layanan AWS dan perbarui sesuai kebutuhan . Untuk informasi selengkap nya, lihat Memilih layanan sistem AWS file di bagian Praktik terbaik dari pola ini. 6. Di kolom Metode migrasi (G), tinjau alat migrasi yang disarankan dan perbarui sesuai kebutuhan. Untuk informasi selengkapnya, lihat Memilih alat migrasi di	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bagian Praktik terbaik dari pola ini. 7. Simpan file SFS-Disco very-Workbook. Anda telah selesai membuat rencana gelombang untuk gelombang ini. 8. Ulangi instruksi ini untuk menyiapkan rencana gelombang untuk setiap gelombang. Karena rencana gelombang dapat berubah selama migrasi, sebaiknya Anda merencana kan tidak lebih dari 5 gelombang sebelumnya.	

Persiapkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan sistem file target.	Menurut detail yang tercatat dalam rencana gelombang Anda, atur sistem file target di target Akun AWS, VPC, dan subnet. Untuk petunjuk, lihat AWS dokumentasi berikut: • Amazon EFS • Amazon FSx untuk NetApp ONTAP • Amazon FSx untuk Server File Windows	Insinyur migrasi, pimpinan Migrasi, Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan alat migrasi dan transfer data.	1. Jika Anda menggunakan AWS DataSync, konfigurasikan pencatatan untuk DataSync tugas. Untuk petunjuk, lihat Mencatat aktivitas AWS DataSync tugas Anda. 2. Siapkan alat migrasi dan lakukan transfer data awal sesuai dengan instruksi untuk alat yang Anda pilih: • Untuk Amazon EFS, lihat yang berikut ini: • Transfer file ke Amazon EFS menggunakan AWS DataSync • Untuk Amazon FSx untuk NetApp ONTAP, lihat yang berikut ini: • Migrasi ke FSx untuk ONTAP menggunakan NetApp SnapMirror • Migrasi ke FSx untuk ONTAP menggunakan AWS DataSync • Untuk Amazon FSx untuk Windows File Server, lihat berikut ini: • Memigrasi file yang ada ke FSx Windows File Server menggunakan AWS DataSync	Administrator AWS, Administrator Cloud, Insinyur migrasi, Pimpinan migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Memigrasi file yang ada ke FSx Windows File Server menggunakan Robocopy 3. Perubahan pada sumber SFS mungkin terjadi selama atau setelah transfer awal. Siapkan transfer data berulang antara sistem file sumber dan target agar data tetap disinkronkan: • Jika Anda menggunakan DataSync, lihat Menjadwalkan AWS DataSync tugas Anda. DataSync mentransfer hanya file yang dimodifikasi atau baru di sumber SFS. • Jika Anda menggunakan alat pihak ketiga, lihat dokumentasi untuk alat yang Anda pilih.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui rencana gelombang.	1. Buka file SFS-Wave-Plan-Workbook untuk gelombang saat ini. 2. Pada lembar kerja Build-Wave, di kolom Alamat IP jalur baru (N), masukkan alamat IP dari sistem file target. Lakukan salah satu hal berikut untuk menemukan alamat IP: • FSx Untuk Windows File Server, di FSx konsol Amazon, pilih Sistem file, pilih sistem file Anda, lalu lihat bagian Jaringan & Keamanan. • FSx Untuk ONTAP, lihat Volume pemasangan. • Untuk Amazon EFS, lihat Memasang dengan alamat IP. 3. Di kolom New path (O), masukkan jalur mount baru. Jalur pemasangan adalah nama DNS dari sistem file. Lakukan salah satu hal berikut untuk menemukan jalur pemasangan: • FSx Untuk Windows File Server, di FSx konsol Amazon, pilih Sistem file, pilih sistem file Anda, lalu pilih Lampirkan.	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• FSx Untuk ONTAP, lihat halaman detail sistem berkas. Untuk petunjuk, lihat Volume pemasangan. • Untuk Amazon EFS, lihat Mengumpulkan Informasi. 4. Pada lembar kerja Remount-Summary, konfirmasikan bahwa kolom New path (C) dan New path IP address (D) mencerminkan nilai yang diperbarui. 5. Konfirmasikan bahwa organisasi Anda telah menyiapkan runbook untuk mengatur ulang sistem file Linux dan Windows setelah cutover. Untuk petunjuk umum, lihat yang berikut ini: • Memasang sistem file Amazon EFS • Mengakses FSx untuk berbagi file Windows File Server • Pemasangan FSx untuk volume ONTAP 6. Jika ada server dependen yang tidak termasuk dalam gelombang ini, catat di lembar kerja App-Team-Communication. Beri tahu pemilik aplikasi atau server	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	masing-masing karena mereka mungkin tidak termasuk dalam komunikasi gelombang standar. 7. Jika SFSs dihapus dari gelombang setelah menyelesaikan rencana gelombang, lacak ini di lembar kerja Descoped.	

Potong

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan aplikasi.	Jika aplikasi atau klien secara aktif melakukan operasi baca dan tulis di SFS sumber, hentikan mereka sebelum Anda melakukan sinkronis asi data akhir. Untuk instruksi , lihat dokumentasi aplikasi atau proses internal Anda untuk menghentikan aktivitas membaca dan menulis. Misalnya, lihat Memulai atau Menghentikan Server Web (IIS 8) (dokumentasi Microsoft) atau Mengelola layanan sistem dengan systemctl (dokumentasi Red Hat).	Pemilik aplikasi, Pengembang aplikasi
Lakukan transfer data akhir.	1. Di alat migrasi, jalankan tugas atau pekerjaan transfer data akhir secara	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	manual untuk menyinkronkan sistem file target dengan SFS sumber. Untuk petunjuk, lihat Memulai DataSync tugas Anda atau lihat dokumentasi untuk alat migrasi pihak ketiga yang Anda pilih. 2. Tunggu tugas transfer data selesai. Untuk informasi selengkapnya, lihat Memantau AWS DataSync aktivitas dengan Amazon CloudWatch dan Memantau DataSync tugas Anda dari baris perintah.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi transfer data.	Jika Anda menggunakan AWS DataSync, lakukan hal berikut untuk memvalidasi transfer data akhir yang berhasil diselesaikan: 1. Di AWS DataSync konsol, buat catatan tugas dan ID eksekusi, seperti `task-0000-exec-1111` . 2. Arahkan ke bagian Task Logging dari DataSync tugas. 3. Pilih tautan grup CloudWatch log. 4. Di log, cari tugas dan ID eksekusi. 5. Catat kesalahan transfer apa pun. Untuk informasi selengkapnya, lihat Kesalahan Umum dalam DataSync dokumentasi. 6. Validasi berikut ini: • Bandingkan daftar file dari sumber dan target SFSs untuk mengonfirmasi bahwa semua data telah ditransfer • Bandingkan izin akses file antara sumber dan target SFSs.	Insinyur migrasi, pimpinan Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jika Anda menggunakan alat pihak ketiga, lihat petunjuk validasi transfer data dalam dokumentasi untuk alat migrasi yang dipilih.	

Validasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Remount sistem file dan validasi fungsi aplikasi dan kinerja.	1. Jika server dependen dimigrasikan dalam gelombang ini, dalam file SFS-Wave-Plan-Work book, pada lembar kerja Remount-Summary, masukkan alamat IP baru server di kolom Alamat IP server baru (F). 2. Di semua server, perbarui titik pemasangan untuk sistem file dari jalur lama ke jalur baru. Gunakan runbook organisasi Anda untuk remounting yang sebelumnya dibahas dalam fase Prepare. 3. Konfirmasikan bahwa sistem file terpasang dengan benar dan dapat diakses dengan memeriksa mount dan memverifikasi file yang ada. Tim infrastru	Administrator sistem AWS, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ktur biasanya melakukan kegiatan ini. 4. Mulai ulang aplikasi dan libatkan pemilik aplikasi atau tim QA untuk menyelesaikan pengujian fungsional dan kinerja pada aplikasi, sesuai kebutuhan untuk aplikasi.	

Pemecahan masalah

Isu	Solusi
Nilai sel di Microsoft Excel tidak diperbarui.	Salin rumus di baris sampel dengan menyeret gagang isian. Untuk informasi selengkapnya, lihat petunjuk untuk Windows atau Mac (situs web Dukungan Microsoft).

Sumber daya terkait

AWS dokumentasi

- AWS DataSync dokumentasi
- Dokumentasi Amazon EFS
- FSx Dokumentasi Amazon
- Migrasi besar ke AWS Cloud
  - Panduan untuk migrasi AWS besar
  - Buku pedoman portofolio untuk migrasi AWS besar

Pemecahan Masalah

- Memecahkan masalah AWS DataSync

- Memecahkan Masalah Amazon EFS
- Memecahkan Masalah Amazon FSx untuk Server File Windows
- Memecahkan masalah Amazon FSx untuk ONTAP NetApp

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Migrasikan database Oracle ke Amazon RDS for Oracle dengan menggunakan adaptor file datar Oracle GoldenGate

Dhairya Jindani dan Baji Shaik, Amazon Web Services

Ringkasan

Oracle GoldenGate adalah layanan pengambilan dan replikasi data real-time untuk database heterogen dan lingkungan TI. Namun, layanan ini saat ini tidak mendukung Amazon Relational Database Service (Amazon RDS) untuk Oracle. Untuk daftar database yang didukung, lihat Oracle GoldenGate for Heterogenous Databases (dokumentasi Oracle). Pola ini menjelaskan cara menggunakan adaptor file GoldenGate datar Oracle GoldenGate dan Oracle untuk menghasilkan file datar dari database Oracle sumber, yang dapat berada di tempat atau di instans Amazon Elastic Compute Cloud (Amazon). EC2 Anda kemudian dapat mengimpor file datar tersebut ke instance database Amazon RDS for Oracle.

Dalam pola ini, Anda menggunakan Oracle GoldenGate untuk mengekstrak file jejak dari database Oracle sumber Anda. Pompa data menyalin file jejak ke server integrasi, yang merupakan EC2 instance Amazon. Pada server integrasi, Oracle GoldenGate menggunakan adaptor file datar untuk menghasilkan serangkaian file datar berurutan berdasarkan pengambilan data transasional dari file jejak.Oracle GoldenGate memformat data sebagai nilai yang dipisahkan pembatas atau nilai yang dibatasi panjang. Anda kemudian menggunakan Oracle SQL* Loader untuk mengimpor file datar ke instans database Amazon RDS for Oracle target.

Audiens yang dituju

Pola ini ditujukan bagi mereka yang memiliki pengalaman dan pengetahuan tentang blok bangunan dasar Oracle GoldenGate. Untuk informasi selengkapnya, lihat Ikhtisar GoldenGate Arsitektur Oracle (dokumentasi Oracle).

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- GoldenGate Lisensi Oracle.
- Lisensi terpisah untuk GoldenGate adaptor Oracle.
- Database Oracle sumber, baik berjalan di tempat atau di instans Amazon EC2 .

- Instans Amazon EC2 Linux yang digunakan sebagai server integrasi. Untuk informasi selengkapnya, lihat Memulai instans Amazon EC2 Linux (EC2 dokumentasi Amazon).
- Target Amazon RDS for Oracle database instance. Untuk informasi selengkapnya, lihat Membuat instans Oracle DB (dokumentasi Amazon RDS).

Versi produk

- Oracle Database Enterprise Edition versi 10g, 11g, 12c, atau yang lebih baru
- Oracle GoldenGate versi 12.2.0.1.1 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

Database Oracle (di tempat atau di EC2 instans Amazon)

Tumpukan teknologi target

Amazon RDS for Oracle

Arsitektur sumber dan target

1. Oracle GoldenGate mengekstrak jejak dari log database sumber.
2. Pompa data mengekstrak jejak dan memigrasikannya ke server integrasi.
3. Adaptor file GoldenGate datar Oracle membaca jejak, definisi sumber, dan parameter ekstrak.
4. Anda keluar dari ekstraksi, yang menghasilkan file kontrol dan file data datar.
5. Anda memigrasikan file data datar ke instance database Amazon RDS for Oracle di file. AWS Cloud

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.

- Amazon Relational Database Service (Amazon RDS) untuk Oracle membantu Anda mengatur, mengoperasikan, dan menskalakan database relasional Oracle di. AWS Cloud

Layanan lainnya

- Oracle GoldenGate adalah layanan yang membantu Anda mereplikasi, memfilter, dan mengubah data dari satu database ke database heterogen lain atau ke topologi target lain, seperti file datar.
- Adaptor GoldenGate aplikasi Oracle memungkinkan Oracle GoldenGate untuk menghasilkan serangkaian file datar berurutan dan file kontrol dari data transaksional yang ditangkap dalam file jejak database sumber. Adaptor ini banyak digunakan untuk operasi ekstrak, transformasi, dan beban (ETL) dalam aplikasi gudang data dan aplikasi berpemilik atau warisan. Oracle GoldenGate melakukan penangkapan ini dan menerapkannya dalam waktu dekat di seluruh database, platform, dan sistem operasi yang heterogen. Adaptor mendukung format yang berbeda untuk file output, seperti CSV atau Apache Parquet. Anda dapat memuat file-file yang dihasilkan ini untuk memuat data ke dalam database heterogen yang berbeda.

Epik

Mengatur Oracle GoldenGate di server database sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh Oracle GoldenGate.	Di server database sumber, unduh Oracle GoldenGate versi 12.2.0.1.1 atau yang lebih baru. Untuk petunjuk, lihat Mengunduh Oracle GoldenGate (dokumentasi Oracle).	DBA
Instal Oracle GoldenGate.	Untuk petunjuk, lihat Menginstal Oracle GoldenGate (dokumentasi Oracle).	DBA
Mengatur Oracle GoldenGate.	Untuk petunjuk, lihat Mempersiapkan Database	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk Oracle GoldenGate (dokumentasi Oracle).	

Siapkan Oracle GoldenGate di server integrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh Oracle GoldenGate.	Di server integrasi, unduh Oracle GoldenGate versi 12.2.0.1.1 atau yang lebih baru. Untuk petunjuk, lihat Mengunduh Oracle GoldenGate (dokumentasi Oracle).	DBA
Instal Oracle GoldenGate.	Buat direktori, atur proses manajer, dan buat `defgen` file untuk lingkungan yang heterogen. Untuk petunjuk, lihat Menginstal Oracle GoldenGate (dokumentasi Oracle).	DBA

Ubah konfigurasi pengambilan GoldenGate data Oracle

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan GoldenGate adaptor Oracle.	Di server integrasi, atur perangkat lunak GoldenGate adaptor Oracle. Lakukan hal-hal berikut: 1. Dari Cloud Pengiriman Perangkat Lunak Oracle,	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	unduh ggs_Adapters_Linux_x64.zip. 2. Buka zip ggs_Adapters_Linux_x64.zip. 3. Jalankan perintah berikut untuk menginstal adaptor: `tar -xvf ggs_Adapters_Linux_x64.tar`	
Konfigurasikan pompa data.	Di server sumber, konfigurasikan pompa data untuk mentransfer file jejak dari server sumber ke server integrasi. Buat file parameter pompa data dan direktori file jejak. Untuk petunjuk, lihat Mengkonfigurasi Adaptor File Datar (dokumentasi Oracle).	DBA

Menghasilkan dan memigrasikan file datar

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan file datar.	Buat file ekstrak dan file kontrol, dan kemudian mulai proses ekstraksi pada server integrasi. Ini mengekstrak perubahan database dan menulis database sumber ke file datar. Untuk petunjuk, lihat Menggunakan Adaptor File Datar (dokumentasi Oracle).	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Muat file datar ke database target.	Muat file datar ke instans database Amazon RDS for Oracle target. Untuk informasi selengkapnya, lihat Mengimpor menggunakan Oracle SQL*Loader (dokumentasi Amazon RDS).	DBA

Pemecahan masalah

Isu	Solusi
Adaptor file GoldenGate datar Oracle menghasilkan kesalahan.	Untuk deskripsi kesalahan adaptor, lihat Menemukan Pesan Kesalahan (dokumentasi Oracle). Untuk petunjuk pemecahan masalah, lihat Memecahkan Masalah Adaptor File Datar (dokumentasi Oracle).

Sumber daya terkait

- Menginstal Oracle GoldenGate (dokumentasi Oracle)
- Mengkonfigurasi Oracle GoldenGate (dokumentasi Oracle)
- Memahami GoldenGate Adaptor Oracle (dokumentasi Oracle)
- Mengkonfigurasi Adaptor File Datar (dokumentasi Oracle)

Ubah aplikasi Python dan Perl untuk mendukung migrasi database dari Microsoft SQL Server ke Amazon Aurora PostgreSQL Edisi yang kompatibel

Dwarika Patra dan Deepesh Jayaprakash, Amazon Web Services

Ringkasan

Pola ini menjelaskan perubahan pada repositori aplikasi yang mungkin diperlukan saat Anda memigrasikan database dari Microsoft SQL Server ke Amazon Aurora PostgreSQL Compatible Edition. Pola mengasumsikan bahwa aplikasi ini berbasis Python atau berbasis Perl, dan memberikan instruksi terpisah untuk bahasa scripting ini.

Migrasi database SQL Server ke Aurora PostgreSQL kompatibel melibatkan konversi skema, konversi objek database, migrasi data, dan pemuatan data. Karena perbedaan antara PostgreSQL dan SQL Server (berkaitan dengan tipe data, objek koneksi, sintaks, dan logika), tugas migrasi yang paling sulit melibatkan membuat perubahan yang diperlukan pada basis kode sehingga bekerja dengan benar dengan PostgreSQL.

Untuk aplikasi berbasis Python, objek koneksi dan kelas tersebar di seluruh sistem. Juga, basis kode Python mungkin menggunakan beberapa pustaka untuk terhubung ke database. Jika antarmuka koneksi database berubah, objek yang menjalankan kueri inline aplikasi juga memerlukan perubahan.

Untuk aplikasi berbasis Perl, perubahan melibatkan objek koneksi, driver koneksi database, pernyataan SQL inline statis dan dinamis, dan bagaimana aplikasi menangani kueri dan set hasil DHTML dinamis yang kompleks.

Saat memigrasikan aplikasi, Anda juga dapat mempertimbangkan kemungkinan penyempurnaan di AWS, seperti mengganti server FTP dengan akses Amazon Simple Storage Service (Amazon S3).

Proses migrasi aplikasi melibatkan tantangan berikut:

- Objek koneksi. Jika objek koneksi tersebar dalam kode dengan beberapa pustaka dan panggilan fungsi, Anda mungkin harus menemukan cara umum untuk mengubahnya untuk mendukung PostgreSQL.
- Penanganan kesalahan atau pengecualian selama pengambilan rekaman atau pembaruan. Jika Anda memiliki operasi membuat, membaca, memperbarui, dan menghapus (CRUD) bersyarat pada database yang mengembalikan variabel, kumpulan hasil, atau bingkai data, kesalahan atau pengecualian apa pun dapat mengakibatkan kesalahan aplikasi dengan efek cascading. Ini

harus ditangani dengan hati-hati dengan validasi yang tepat dan menyimpan poin. Salah satu titik penyimpanan tersebut adalah memanggil kueri SQL sebaris besar atau objek database di dalam blok. `BEGIN...EXCEPTION...END`

- Mengontrol transaksi dan validasinya. Ini termasuk komit dan rollback manual dan otomatis. Driver PostgreSQL untuk Perl mengharuskan Anda untuk selalu secara eksplisit mengatur atribut auto-commit.
- Menangani query SQL dinamis. Ini membutuhkan pemahaman yang kuat tentang logika kueri dan pengujian berulang untuk memastikan bahwa kueri berfungsi seperti yang diharapkan.
- Kinerja. Anda harus memastikan bahwa perubahan kode tidak mengakibatkan penurunan kinerja aplikasi.

Pola ini menjelaskan proses konversi secara rinci.

Prasyarat dan batasan

Prasyarat

- Pengetahuan tentang sintaks Python dan Perl.
- Keterampilan dasar dalam SQL Server dan PostgreSQL.
- Memahami arsitektur aplikasi Anda yang ada.
- Akses ke kode aplikasi Anda, database SQL Server, dan database PostgreSQL.
- Akses ke lingkungan pengembangan Windows atau Linux (atau Unix lainnya) dengan kredensil untuk mengembangkan, menguji, dan memvalidasi perubahan aplikasi.
- Untuk aplikasi berbasis Python, pustaka Python standar yang mungkin diperlukan aplikasi Anda, seperti Pandas untuk menangani frame data, dan psycopg2 atau untuk koneksi database. SQLAlchemy
- Untuk aplikasi berbasis Perl, diperlukan paket Perl dengan pustaka atau modul dependen. Modul Comprehensive Perl Archive Network (CPAN) dapat mendukung sebagian besar persyaratan aplikasi.
- Semua perpustakaan atau modul khusus tergantung yang diperlukan.
- Kredensi database untuk akses baca ke SQL Server dan read/write akses ke Aurora.
- PostgreSQL untuk memvalidasi dan men-debug perubahan aplikasi dengan layanan dan pengguna.
- Akses ke alat pengembangan selama migrasi aplikasi seperti Visual Studio Code, Sublime Text, atau pgAdmin.

Batasan

- Beberapa versi Python atau Perl, modul, pustaka, dan paket tidak kompatibel dengan lingkungan cloud.
- Beberapa pustaka dan kerangka kerja pihak ketiga yang digunakan untuk SQL Server tidak dapat diganti untuk mendukung migrasi PostgreSQL.
- Variasi kinerja mungkin memerlukan perubahan pada aplikasi Anda, untuk kueri Transact-SQL (T-SQL) sebaris, fungsi database, dan prosedur tersimpan.
- PostgreSQL mendukung nama huruf kecil untuk nama tabel, nama kolom, dan objek database lainnya.
- Beberapa tipe data, seperti kolom UUID, disimpan dalam huruf kecil saja. Aplikasi Python dan Perl harus menangani perbedaan kasus tersebut.
- Perbedaan pengkodean karakter harus ditangani dengan tipe data yang benar untuk kolom teks yang sesuai dalam database PostgreSQL.

Versi produk

- Python 3.6 atau yang lebih baru (gunakan versi yang mendukung sistem operasi Anda)
- Perl 5.8.3 atau yang lebih baru (gunakan versi yang mendukung sistem operasi Anda)
- Aurora PostgreSQL Compatible Edition 4.2 atau yang lebih baru (lihat detail)

Arsitektur

Tumpukan teknologi sumber

- Bahasa scripting (pemrograman aplikasi): Python 2.7 atau yang lebih baru, atau Perl 5.8
- Database: Microsoft SQL Server versi 13
- Sistem operasi: Red Hat Enterprise Linux (RHEL) 7

Tumpukan teknologi target

- Bahasa scripting (pemrograman aplikasi): Python 3.6 atau yang lebih baru, atau Perl 5.8 atau yang lebih baru
- Database: Aurora PostgreSQL kompatibel 4.2
- Sistem operasi: RHEL 7

Arsitektur migrasi

Alat

Layanan dan alat AWS

- Aurora PostgreSQL — Compatible Edition adalah mesin basis data relasional relasional yang dikelola sepenuhnya, kompatibel dengan PostgreSQL, dan sesuai dengan asam yang menggabungkan kecepatan dan keandalan database komersial kelas atas dengan efektivitas biaya database sumber terbuka. Aurora PostgreSQL adalah pengganti drop-in untuk PostgreSQL dan membuatnya lebih mudah dan lebih hemat biaya untuk mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL Anda yang baru dan yang sudah ada.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang memungkinkan Anda berinteraksi dengan layanan AWS dengan menggunakan perintah di shell baris perintah Anda.

Alat-alat lainnya

- Pustaka koneksi database Python dan PostgressQL seperti psycopg2 dan SQLAlchemy
- Perl dan modul DBI-nya
- Terminal interaktif PostgreSQL (psql)

Epik

Migrasikan repositori aplikasi Anda ke PostgreSQL — langkah tingkat tinggi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ikuti langkah-langkah konversi kode ini untuk memigrasikan aplikasi Anda ke PostgreSQL.	1. Tetapkan driver dan pustaka ODBC khusus database untuk PostgreSQ L. Misalnya, Anda dapat menggunakan salah satu modul CPAN untuk Perl dan pyodbc, psycopg2, atau untuk Python. SQLAlchemy	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Mengkonversi objek database dengan menggunakan pustaka ini untuk terhubung ke Aurora PostgreSQL kompatibel. 3. Terapkan perubahan kode dalam modul aplikasi yang ada untuk mendapatkan pernyataan T-SQL yang kompatibel. 4. Tulis ulang panggilan fungsi khusus database dan prosedur tersimpan dalam kode aplikasi. 5. Menangani perubahan pada variabel aplikasi Anda dan tipe datanya yang digunakan untuk kueri SQL sebaris. 6. Menangani fungsi khusus database yang tidak kompatibel. 7. end-to-endPengujian lengkap kode aplikasi yang dikonversi untuk migrasi database. 8. Bandingkan hasil dari Microsoft SQL Server dengan aplikasi yang Anda migrasi ke PostgreSQL. 9. Lakukan benchmarking kinerja aplikasi antara	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Microsoft SQL Server dan PostgreSQL. 10. Merevisi prosedur tersimpan atau pernyataan T-SQL inline yang dipanggil oleh aplikasi untuk meningkatkan kinerja. Epos berikut memberikan instruksi terperinci untuk beberapa tugas konversi ini untuk aplikasi Python dan Perl.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan daftar periksa untuk setiap langkah migrasi.	Tambahkan yang berikut ini ke daftar periksa Anda untuk setiap langkah migrasi aplikasi, termasuk langkah terakhir: • Tinjau dokumentasi PostgreSQL untuk memastikan bahwa semua perubahan Anda kompatibel dengan standar PostgreSQL. • Periksa bilangan bulat dan nilai mengambang untuk kolom. • Identifikasi jumlah baris yang disisipkan, diperbarui, dan diekstraksi, bersama dengan nama kolom dan date/time prangko. Anda dapat menggunakan utilitas diff atau menulis skrip untuk mengotomatiskan pemeriksaan ini. • Selesaikan pemeriksaan kinerja untuk pernyataan SQL inline yang besar, dan periksa kinerja keseluruhan aplikasi. • Periksa penanganan kesalahan yang benar untuk operasi database dan keluar program yang	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	anggun dengan menggunakan beberapa try/catch blok. • Periksa untuk memastikan bahwa proses logging yang tepat sudah ada.	

Analisis dan perbarui aplikasi Anda — basis kode Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Analisis basis kode Python Anda yang ada.	Analisis Anda harus mencakup hal-hal berikut untuk memfasilitasi proses migrasi aplikasi: • Identifikasi semua objek koneksi dalam kode. • Identifikasi semua kueri SQL inline yang tidak kompatibel (seperti pernyataan T-SQL dan prosedur tersimpan) dan analisis perubahan yang diperlukan. • Tinjau dokumentasi untuk kode Anda dan lacak alur kontrol untuk memahami fungsionalitas kode. Ini akan membantu nanti ketika Anda menguji aplikasi untuk perbandingan kinerja atau beban. • Pahami tujuan aplikasi sehingga Anda dapat	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengujinya secara efektif setelah konversi database. Sebagian besar aplikasi Python yang merupakan kandidat untuk konversi dengan migrasi database adalah feed yang memuat data dari sumber lain ke dalam tabel database, atau ekstraktor yang mengambil data dari tabel dan mengubahnya menjadi format output yang berbeda (seperti CSV, JSON, atau file datar) yang cocok untuk membuat laporan atau untuk membuat panggilan API untuk melakukan validasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konversikan koneksi database Anda untuk mendukung PostgreSQL.	Sebagian besar aplikasi Python menggunakan pustaka pyodbc untuk terhubung dengan database SQL Server sebagai berikut.
	
	`import pyodbc`
	`....`
	`try:`
	`conn_string = "Driver=ODBC Driver 17 for SQL Server;UID={};PWD={};Server={};Database={}".format (conn_user, conn_password, conn_server, conn_database)`
	`conn = pyodbc.connect(conn_string)`
	`cur = conn.cursor()`
	`result = cur.execute(query_string)`
	`for row in result:`
	`print (row)`
	`except Exception as e:`
	`print(str(e))`
	
	Mengkonversi koneksi database untuk mendukung PostgreSQL sebagai berikut.
	
	`import pyodbc`
	`import psycopg2`
	`....`
	`try:`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> conn_string =
	 'postgresql+psycop
	g2://'+
	    conn_user+':'+conn
	_password+'@'+conn
	_server+'/'+conn_d
	atabase
	    conn = pyodbc.co
	nnect(conn_string,
	 connect_args={'opt
	ions':'-csearch_pa
	th=dbo'})
	    cur = conn.cursor()
	    result = cur.execu
	te(query_string)
	    for row in result:
	    print (row)
	except Exception as e:
	    print(str(e))</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Ubah kueri SQL sebaris ke PostgreSQL.	Konversikan kueri SQL sebaris Anda ke format yang kompatibel dengan PostgreSQL. Misalnya, query SQL Server berikut mengambil string dari tabel.
	
	`dtype = "type1"`
	`stm = ‘"SELECT TOP 1 searchcode FROM TypesTable (NOLOCK) WHERE code="’ + "’" + str(dtype) + "’"`
	`# For Microsoft SQL Server Database Connection`
	`engine = create_engine(‘mssql+pyodbc:///?odbc_connect=%s’ % urllib.parse.quote_plus(conn_string), connect_args={‘connect_timeout’:login_timeout})`
	`conn = engine_connect()`
	`rs = conn.execute(stm)`
	`for row in rs:`
	`print(row)`
	
	Setelah konversi, kueri SQL sebaris yang kompatibel dengan PostgreSQL terlihat seperti berikut.
	
	`dtype = "type1"`
	`stm = ‘"SELECT searchcode FROM TypesTable`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`WHERE code="’ + "’" + str(dtype) + "’ LIMIT 1"`
	`# For PostgreSQL Database Connection`
	`engine = create_engine(‘postgres+psycopg2://%s’ %conn_string, connect_args={‘connect_timeout’:login_timeout})`
	`conn = engine.connect()`
	`rs = conn.execute(stm)`
	`for row in rs:`
	`  print(row)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menangani query SQL dinamis.	SQL dinamis dapat hadir dalam satu skrip atau dalam beberapa skrip Python. Contoh sebelumnya menunjukkan bagaimana menggunakan fungsi penggantian string Python untuk menyisipkan variabel untuk membangun query SQL dinamis. Pendekatan alternatif adalah menambahkan string kueri dengan variabel di mana pun berlaku.
	
	Dalam contoh berikut, string kueri dibangun dengan cepat berdasarkan nilai yang dikembalikan oleh fungsi.
	
	`query = '"SELECT id from equity e join issues i on e.permId=i.permId where e.id'"`
	`query += get_id_filter(ids) + " e.id is NOT NULL`
	
	Jenis kueri dinamis ini sangat umum selama migrasi aplikasi. Ikuti langkah-langkah berikut untuk menangani kueri dinamis:
	
	• Periksa sintaks keseluruhan (misalnya, sintaks untuk
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`SELECT` pernyataan dengan `JOIN` klausa). • Verifikasi semua variabel atau nama kolom yang digunakan dalam kueri, seperti `i` dan`id`. • Periksa fungsi, argumen, dan nilai pengembalian yang digunakan dalam kueri (misalnya, `get_id_filter` dan argumenny `aids`).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menangani set hasil, variabel, dan frame data.	Untuk Microsoft SQL Server, Anda menggunakan metode Python `fetchone()` seperti `fetchall()` atau untuk mengambil hasil yang ditetapkan dari database. Anda juga dapat menggunakan `fetchmany(size)` dan menentukan jumlah catatan yang akan dikembalikan dari hasil yang ditetapkan. Untuk melakukan ini, Anda dapat menggunakan objek koneksi pyodbc seperti yang ditunjukkan pada contoh berikut.
	
	pyodbc (Microsoft SQL Server)
	
	`import pyodbc`
	`server = 'tcp:myse`
	`rver.database.wind`
	`ows.net'`
	`database = 'exampledb'`
	`username = 'exampleu`
	`sername'`
	`password = 'examplep`
	`assword'`
	`conn = pyodbc.co`
	`nnect('DRIVER={ODB`
	`C Driver 17 for SQL`
	` Server};SERVER='+s`
	`erver+';DATABASE='`
	`+database+';UID='+`
	`username+';PWD='+`
	` password)`
	`cursor = conn.cursor()`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cursor.execute("SELECT * FROM ITEMS")`
	`row = cursor.fetchone()`
	`while row:`
	`  print(row[0])`
	`  row = cursor.fetchone()`
	
	Di Aurora, untuk melakukan tugas serupa seperti menghubungkan ke PostgreSQL dan mengambil set hasil, Anda dapat menggunakan psycopg2 atau. SQLAlchemy Pustaka Python ini menyediakan modul koneksi dan objek kursor untuk melintasi catatan database PostgreSQL, seperti yang ditunjukkan pada contoh berikut.
	
	psycopg2 (Kompatibel dengan Aurora PostgreSQL)
	
	`import psycopg2`
	`query = "SELECT * FROM ITEMS;"`
	`//Initialize variables`
	`host=dbname=user=password=port=sslmode=connect_timeout=""`
	`connstring = "host='{host}' dbname='{`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dbname}' user='{user}' \ password='{passw ord}'port='{port}' ".format(host=host ,dbname=dbname,\ user=user,password= password,port=port) conn = psycopg2. connect(connstring) cursor = conn.cursor() cursor.execute(query) column_names = [column[0] for column in cursor.description] print("Column Names: ", column_names) print("Column values: " for row in cursor: print("itemid :", row[0]) print("itemdescrpt ion :", row[1]) print("it emprice :", row[3]))	
	SQLAlchemy (Kompatibel dengan Aurora PostgreSQL) from sqlalchemy import create_engine from pandas import DataFrame conn_string = 'postgres ql://core:database @localhost:5432/ex ampledatabase' engine = create_en gine(conn_string)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`conn = engine.co nnect() dataid = 1001 result = conn.exec ute("SELECT * FROM ITEMS") df = DataFrame (result.fetchall()) df.columns = result.ke ys() df = pd.DataFrame() engine.connect() df = pd.read_sql_query( sql_query, engine, coerce_float=False) print("df=", df)`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aplikasi Anda selama dan setelah migrasi.	Menguji aplikasi Python yang dimigrasi adalah proses yang berkelanjutan. Karena migrasi mencakup perubahan objek koneksi (psycopg2 atau SQLAlchemy), penangana n kesalahan, fitur baru (bingkai data), perubahan SQL sebaris, fungsi salinan massal (`bcp`bukan`COPY`) dan perubahan serupa, migrasi harus diuji dengan cermat selama dan setelah migrasi aplikasi. Periksa: • Kondisi kesalahan dan penanganan • Ketidakcocokan catatan apa pun setelah migrasi • Rekam pembaruan atau penghapusan • Waktu yang dibutuhkan untuk menjalankan aplikasi	Pengembang aplikasi

Analisis dan perbarui aplikasi Anda — Basis kode Perl

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Analisis basis kode Perl Anda yang ada.	Analisis Anda harus mencakup hal-hal berikut untuk memfasilitasi proses migrasi aplikasi. Anda harus mengidentifikasi:	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Kode ini atau berbasis konfigurasi apa pun • Driver Perl Open Database Connectivity (ODBC) standar khusus database atau driver apa pun yang disesuaikan • Perubahan kode diperlukan untuk kueri inline dan T-SQL • Interaksi antara berbagai modul Perl (misalnya, objek koneksi Perl ODBC tunggal yang disebut atau digunakan oleh beberapa komponen fungsional) • Set data dan penanganan set hasil • Pustaka Perl eksternal dan tergantung • Apa pun APIs yang digunakan dalam aplikasi • Kompatibilitas versi Perl dan kompatibilitas driver dengan Aurora PostgreSQL kompatibel	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mengkonversi koneksi dari aplikasi Perl dan modul DBI untuk mendukung PostgreSQL.	Aplikasi berbasis Perl umumnya menggunakan modul Perl DBI, yang merupakan modul akses database standar untuk bahasa pemrograman Perl. Anda dapat menggunakan modul DBI yang sama dengan driver yang berbeda untuk SQL Server dan PostgreSQL.
	
	Untuk informasi selengkapnya tentang modul Perl yang diperlukan, instalasi, dan instruksi lainnya, lihat dokumentasi DBD: :Pg. Contoh berikut terhubung ke Aurora PostgreSQL kompatibel di. `exampletest-aurorapg-database.cluster-sampleclusture.us-east-.rds.amazonaws.com`
	
	`#!/usr/bin/perl`
	`use DBI;`
	`use strict;`
	`my $driver = "Pg";`
	`my $hostname = "exampletest-aurorapg-database-sampleclusture.us-east.rds.amazonaws.com"`
	`my $dsn = "DBI:$driver: dbname = $hostname;host = 127.0.0.1;port = 5432";`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`my $username = "postgres";`
	`my $password = "pass123";`
	`$dbh = DBI->connect("dbi:Pg:dbname=$hostname;host=$host;port=$port;options=$options",`
	`$username,`
	`$password,`
	`{AutoCommit => 0, RaiseError => 1, PrintError => 0}`
	`);`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah kueri SQL Inline ke PostgreSQL.	Aplikasi Anda mungkin memiliki kueri SQL sebaris dengan`SELECT`,, `DELETE``UPDATE`, dan pernyataan serupa yang menyertakan klausa kueri yang tidak didukung PostgreSQL. Misalnya, kata kunci kueri seperti `TOP` dan `NOLOCK` tidak didukung di PostgreSQL. Contoh berikut menunjukkan bagaimana Anda dapat menangani `TOP`,`NOLOCK`, dan variabel Boolean.  Di SQL Server:  `$sqlStr = $sqlStr . "WHERE a.student _id in (SELECT TOP $numofRecords c_student_id \ FROM active_student_rec ord b WITH (NOLOCK) \ INNER JOIN student_c ontributor c WITH (NOLOCK) on c.contrib utor_id = b.c_st)`  Untuk PostgreSQL, konversi ke:  `$sqlStr = $sqlStr . "WHERE a.student _id in (SELECT`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	TOP $numofRecords c_student_id \ FROM active_student_rec ord b INNER JOIN student_contributor c \ on c.contributor_id = b.c_student_contr_id WHERE b_current_1 is true \ LIMIT $numofRecords)"	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menangani query SQL dinamis dan variabel Perl.	Query SQL dinamis adalah pernyataan SQL yang dibangun pada runtime aplikasi. Kueri ini dibangun secara dinamis saat aplikasi berjalan, tergantung pada kondisi tertentu, sehingga teks lengkap kueri tidak diketahui sampai runtime. Contohnya adalah aplikasi analitik keuangan yang menganali sis 10 saham teratas setiap hari, dan saham ini berubah setiap hari. Tabel SQL dibuat berdasarkan performa terbaik, dan nilainya tidak diketahui sampai runtime. Katakanlah kueri SQL sebaris untuk contoh ini diteruskan ke fungsi pembungkus untuk mendapatkan hasil yang ditetapkan dalam variabel, dan kemudian variabel menggunak an kondisi untuk menentukan apakah tabel itu ada: • Jika tabel ada, jangan membuatnya; lakukan beberapa pemrosesan. • Jika tabel tidak ada, buat tabel dan lakukan juga beberapa pemrosesan.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Berikut adalah contoh penanganan variabel, diikuti oleh SQL Server dan PostgreSQL query untuk kasus penggunaan ini.
	
	`my $tableexists = db_read( arg 1, $sql_qry, undef, 'writer'); my $table_already_exists = $tableexists->[0]{table_exists}; if ($table_already_exists){ # do some thing } else { # do something else }`
	
	Server SQL:
	
	`my $sql_qry = "SELECT OBJECT_ID('$backendTable', 'U') table_exists", undef, 'writer') ";`
	
	PostgreSQL:
	
	`my $sql_qry = "SELECT TO_REGCLASS('$backendTable', 'U') table_exists", undef, 'writer')";`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Contoh berikut menggunakan variabel Perl dalam SQL inline, yang menjalankan `SELECT` pernyataan dengan `JOIN` untuk mengambil kunci utama dari tabel dan posisi kolom kunci.
	
	Server SQL:
	
	`my $sql_qry = "SELECT column_name', character_maximum_length \ FROM INFORMATION_SCHEMA.COLUMNS \ WHERE TABLE_SCHEMA='$example_schemaInfo' \ AND TABLE_NAME='$example_table' \ AND DATA_TYPE IN ('varchar','nvarchar');";`
	
	PostgreSQL:
	
	`my $sql_qry = "SELECT c1.column_name, c1.ordinal_position \ FROM information_schema.key_column_usage AS c LEFT \ JOIN information_schema.table_constraints AS t1 \ ON t1.constraint_name = c1.constraint_name \`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`WHERE t1.table_name = $example_schemaInfo'.'$example_table' \ AND t1.constraint_type = 'PRIMARY KEY' ;";`	

Buat perubahan tambahan pada aplikasi berbasis Perl atau Python Anda untuk mendukung PostgreSQL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengkonversi konstruksi SQL Server tambahan ke PostgreSQL.	Perubahan berikut berlaku untuk semua aplikasi, terlepas dari bahasa pemrograman. • Memenuhi syarat objek database yang digunakan aplikasi Anda dengan nama skema baru dan sesuai. • Tangani operator LIKE untuk pencocokan peka huruf besar/kecil dengan fitur pemeriksaan di PostgreSQL. • Menangani fungsi spesifik database yang tidak didukung sepertiDATEDIFF,DATEADD,, GETDATECONVERT, dan CAST operator. Untuk fungsi yang kompatibel dengan PostgreSQL yang setara, lihat Fungsi SQL asli atau bawaan di bagian Informasi tambahan.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Menangani nilai Boolean dalam pernyataan perbandingan. • Menangani nilai pengembalian dari fungsi. Ini bisa berupa kumpulan rekaman, bingkai data, variabel, dan nilai Boolean. Tangani ini sesuai dengan persyaratan aplikasi Anda dan untuk mendukung PostgreSQL. • Tangani blok anonim (sepertiBEGIN TRAN) dengan fungsi PostgreSQL baru yang ditentukan pengguna. • Konversi sisipan massal untuk baris. PostgreSQL setara dengan utilitas SQL Server bulk copy bcp (), yang dipanggil dari dalam aplikasi, adalah. COPY • Mengkonversi operator penggabungan kolom. SQL Server menggunakan + untuk penggabungan string, tetapi PostgreSQL menggunakan. | |	

Tingkatkan kinerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Manfaatkan layanan AWS untuk melakukan peningkatan kinerja.	Saat bermigrasi ke AWS Cloud, Anda dapat menyempurnakan desain aplikasi dan database Anda untuk memanfaatkan layanan AWS. Misalnya, jika kueri dari aplikasi Python Anda, yang terhubung ke server database yang kompatibel dengan Aurora PostgreSQL, membutuhkan waktu lebih lama daripada kueri Microsoft SQL Server asli Anda, Anda dapat mempertimbangkan untuk membuat umpan data historis langsung ke bucket Amazon Simple Storage Service (Amazon S3) dari server Aurora, dan menggunakan bucket Amazon Athena-based Kueri SQL untuk menghasilkan laporan dan kueri data analitik untuk dasbor pengguna Anda.	Pengembang aplikasi, arsitek Cloud

Sumber daya terkait

- Perl
- Perl DBI Modul
- Python
- psycopg2

- SQLAlchemy
- Salinan Massal - PostgreSQL
- Salinan Massal - Microsoft SQL Server
- PostgreSQL
- Bekerja dengan Amazon Aurora PostgreSQL

Informasi tambahan

Baik Microsoft SQL Server dan Aurora PostgreSQL kompatibel adalah ANSI SQL-Complaint. Namun, Anda harus tetap mengetahui adanya ketidakcocokan dalam sintaks, tipe data kolom, fungsi khusus database asli, sisipan massal, dan sensitivitas huruf besar saat Anda memigrasikan aplikasi Python atau Perl dari SQL Server ke PostgreSQL.

Bagian berikut memberikan informasi lebih lanjut tentang kemungkinan inkonsistensi.

Perbandingan tipe data

Perubahan tipe data dari SQL Server ke PostgreSQL dapat menyebabkan perbedaan yang signifikan dalam data yang dihasilkan di mana aplikasi beroperasi. Untuk perbandingan tipe data, lihat tabel di situs web Sqlines.

Fungsi SQL asli atau bawaan

Perilaku beberapa fungsi berbeda antara database SQL Server dan PostgreSQL. Tabel berikut memberikan perbandingan.

Microsoft SQL Server	Deskripsi	PostgreSQL
`CAST`	Mengkonversi nilai dari satu tipe data ke yang lain.	PostgreSQL `type :: operator`
`GETDATE()`	Mengembalikan tanggal dan waktu sistem database saat ini, dalam `YYYY-MM-DD hh:mm:ss.mmm` format.	`CLOCK_TIMESTAMP`
`DATEADD`	Menambahkan time/date interval ke tanggal.	`INTERVAL`ekspresi

`CONVERT`	Mengkonversi nilai ke format data tertentu.	`TO_CHAR`
`DATEDIFF`	Mengembalikan perbedaan antara dua tanggal.	`DATE_PART`
`TOP`	Membatasi jumlah baris dalam set `SELECT` hasil.	`LIMIT/FETCH`

Blok anonim

Kueri SQL terstruktur diatur ke dalam beberapa bagian seperti deklarasi, executable, dan penanganan pengecualian. Tabel berikut membandingkan versi Microsoft SQL Server dan PostgreSQL dari blok anonim sederhana. Untuk blok anonim yang kompleks, kami sarankan Anda memanggil fungsi basis data khusus dalam aplikasi Anda.

Microsoft SQL Server

```
my $sql_qry1=
my $sql_qry2 =
my $sqlqry = "BEGIN TRAN
$sql_qry1 $sql_qry2
if @\@error !=0 ROLLBACK
TRAN
else COMIT TRAN";
```

PostgreSQL

```
my $sql_qry1=
my $sql_qry2 =
my $sql_qry = " DO \$\$
BEGIN
$header_sql $content_sql
END
\$\$";
```

Perbedaan lainnya

- Sisipan baris massal: PostgreSQL setara dengan utilitas Microsoft SQL Server bcp adalah COPY.
- Sensitivitas kasus: Nama kolom peka huruf besar/kecil di PostgreSQL, jadi Anda harus mengonversi nama kolom SQL Server Anda menjadi huruf kecil atau huruf besar. Ini menjadi faktor ketika Anda mengekstrak atau membandingkan data, atau menempatkan nama kolom dalam kumpulan hasil atau variabel. Contoh berikut mengidentifikasi kolom di mana nilai mungkin disimpan dalam huruf besar atau kecil.

```
my $sql_qry = "SELECT $record_id FROM $exampleTable WHERE LOWER($record_name) =
 \'failed transaction\'";
```

- Penggabungan: SQL Server menggunakan + sebagai operator untuk penggabungan string, sedangkan PostgreSQL menggunakan. | |
- Validasi: Anda harus menguji dan memvalidasi query SQL sebaris dan fungsi sebelum Anda menggunakannya dalam kode aplikasi untuk PostgreSQL.
- Penyertaan Perpustakaan ORM: Anda juga dapat mencari termasuk atau mengganti pustaka koneksi database yang ada dengan pustaka ORM Python seperti dan PynoMoDB. SQLAlchemy Ini akan membantu untuk dengan mudah query dan memanipulasi data dari database menggunakan paradigma berorientasi objek.

Pola migrasi berdasarkan beban kerja

Topik

- IBM
- Microsoft
- N/A
- Sumber terbuka
- Oracle
- SAP

IBM

- Migrasikan database Db2 dari Amazon ke EC2 Aurora MySQL yang kompatibel dengan menggunakan AWS DMS
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan menggunakan pengiriman log untuk mengurangi waktu pemadaman
- Migrasi Db2 untuk LUW ke EC2 Amazon dengan pemulihan bencana ketersediaan tinggi
- Bermigrasi dari IBM Db2 di Amazon EC2 ke Aurora PostgreSQL yang kompatibel dengan menggunakan AWS DMS dan AWS SCT
- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2

Microsoft

- Mempercepat penemuan dan migrasi beban kerja Microsoft ke AWS
- Ubah aplikasi Python dan Perl untuk mendukung migrasi database dari Microsoft SQL Server ke Amazon Aurora PostgreSQL Edisi yang kompatibel
- Buat CloudFormation templat AWS untuk tugas AWS DMS menggunakan Microsoft Excel dan Python
- Ekspor database Microsoft SQL Server ke Amazon S3 dengan menggunakan AWS DMS
- Menerapkan SHA1 hashing untuk data PII saat bermigrasi dari SQL Server ke PostgreSQL
- Menyerap dan memigrasikan instans EC2 Windows ke akun AWS Managed Services
- Memigrasikan antrian pesan dari Microsoft Azure Service Bus ke Amazon SQS
- Migrasi database Microsoft SQL Server dari Amazon EC2 ke Amazon DocumentDB dengan menggunakan AWS DMS
- Migrasikan database Microsoft SQL Server ke Aurora MySQL dengan menggunakan AWS DMS dan AWS SCT
- Migrasi aplikasi.NET dari Microsoft Azure App Service ke AWS Elastic Beanstalk
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan server tertaut
- Memigrasi database Microsoft SQL Server lokal ke Amazon RDS for SQL Server menggunakan metode pencadangan dan pemulihan asli
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan AWS DMS
- Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan agen ekstraksi data AWS SCT
- Memigrasi database Microsoft SQL Server lokal ke Microsoft SQL Server di Amazon yang menjalankan Linux EC2
- Migrasi data dari Microsoft Azure Blob ke Amazon S3 dengan menggunakan Rclone
- Migrasi aplikasi yang dihosting IIS ke Amazon dengan menggunakan appcmd.exe EC2
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2 menggunakan Layanan Migrasi Aplikasi
- Migrasikan sertifikat SSL Windows ke Application Load Balancer menggunakan ACM

- Menghosting ulang beban kerja lokal di AWS Cloud: daftar periksa migrasi
- Mengatasi kesalahan koneksi setelah memigrasikan Microsoft SQL Server ke AWS Cloud
- Menyiapkan infrastruktur Multi-AZ untuk SQL Server Always On FCI dengan menggunakan Amazon FSx

N/A

- Membuat proses persetujuan untuk permintaan firewall selama migrasi rehost ke AWS

Sumber terbuka

- Buat pengguna dan peran aplikasi di Aurora PostgreSQL yang kompatibel
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dengan dan menggunakan dan AWS SCT AWS DMS AWS CLI CloudFormation
- Memigrasi database MariaDB lokal ke Amazon RDS for MariaDB menggunakan alat bawaan
- Memigrasikan database MySQL lokal ke Amazon RDS for MySQL
- Memigrasikan database MySQL lokal ke Aurora MySQL
- Memigrasikan database PostgreSQL lokal ke Aurora PostgreSQL
- Memigrasikan database Couchbase Server ke Amazon EC2
- Bermigrasi dari Server WebSphere Aplikasi IBM ke Apache Tomcat di Amazon EC2 dengan Auto Scaling
- Bermigrasi dari PostgreSQL di Amazon ke Amazon RDS untuk PostgreSQL menggunakan pglogical EC2
- Migrasikan aplikasi Java lokal ke AWS menggunakan AWS App2Container
- Migrasikan database MySQL lokal ke Aurora MySQL menggunakan Percona, Amazon EFS, dan Amazon S3 XtraBackup
- Migrasikan tabel eksternal Oracle ke Amazon Aurora PostgreSQL yang kompatibel
- Mulai ulang AWS Replication Agent secara otomatis tanpa menonaktifkan SELinux setelah me-reboot server sumber RHEL
- Transportasi database PostgreSQL antara dua instans Amazon RDS DB menggunakan pg_transport

Oracle

- Mengkonversi VARCHAR2 (1) tipe data untuk Oracle ke tipe data Boolean untuk Amazon Aurora PostgreSQL
- Meniru Oracle DR dengan menggunakan database global Aurora yang kompatibel dengan PostgreSQL
- Bermigrasi secara bertahap dari Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL menggunakan Oracle SQL Developer dan AWS SCT
- Muat file BLOB ke TEXT dengan menggunakan pengkodean file di Aurora PostgreSQL yang kompatibel
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dalam mode SSL dengan menggunakan AWS DMS
- Migrasikan database Amazon RDS for Oracle ke database Akun AWS lain Wilayah AWS dan gunakan untuk replikasi yang sedang berlangsung AWS DMS
- Memigrasikan database Oracle lokal ke Amazon menggunakan Oracle Data EC2 Pump
- Memigrasi database Oracle lokal ke Amazon OpenSearch Service menggunakan Logstash
- Memigrasikan database Oracle lokal ke Amazon RDS for MySQL menggunakan AWS DMS dan AWS SCT
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle dengan menggunakan Impor Oracle Data Pump langsung melalui tautan database
- Memigrasikan database Oracle lokal ke Amazon RDS for Oracle menggunakan Oracle Data Pump
- Memigrasikan database Oracle lokal ke Amazon RDS for PostgreSQL menggunakan Oracle bystander dan AWS DMS
- Memigrasi database Oracle lokal ke Oracle di Amazon EC2
- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS untuk MariaDB menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle dari Amazon EC2 ke Amazon RDS for Oracle menggunakan AWS DMS
- Migrasikan database Oracle ke Amazon RDS for Oracle dengan menggunakan adaptor file datar Oracle GoldenGate
- Migrasikan Database Oracle ke Amazon Redshift menggunakan AWS DMS dan AWS SCT
- Migrasikan database Oracle ke Aurora PostgreSQL menggunakan AWS DMS dan AWS SCT

- Migrasikan database Oracle JD Edwards EnterpriseOne ke AWS dengan menggunakan Oracle Data Pump dan AWS DMS
- Migrasikan tabel partisi Oracle ke PostgreSQL menggunakan AWS DMS
- Migrasikan PeopleSoft database Oracle ke AWS dengan menggunakan AWS DMS
- Memigrasi data dari database Oracle lokal ke Aurora PostgreSQL
- Bermigrasi dari Amazon RDS for Oracle ke Amazon RDS untuk MySQL
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk PostgreSQL menggunakan tampilan terwujud dan AWS DMS
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Amazon RDS for PostGresQL dan AWS DMS SharePlex
- Migrasi dari Oracle Database ke Amazon RDS for PostgreSQL dengan menggunakan Oracle GoldenGate
- Bermigrasi dari Oracle di Amazon EC2 ke Amazon RDS untuk MySQL menggunakan AWS DMS dan AWS SCT
- Bermigrasi dari Oracle WebLogic ke Apache Tomcat (ToMee) di Amazon ECS
- Migrasikan indeks berbasis fungsi dari Oracle ke PostgreSQL
- Migrasi aplikasi lama dari Oracle Pro*C ke ECPG
- Migrasikan nilai Oracle CLOB ke baris individual di PostgreSQL di AWS
- Migrasikan kode kesalahan Oracle Database ke database yang kompatibel dengan Amazon Aurora PostgreSQL
- Migrasikan fungsi asli Oracle ke PostgreSQL menggunakan ekstensi
- Migrasi Oracle ke PeopleSoft Amazon RDS Kustom
- Migrasikan fungsionalitas Oracle ROWID ke PostgreSQL di AWS
- Migrasikan paket pragma Oracle SERIALLY_REUSABLE ke PostgreSQL
- Migrasikan kolom virtual yang dihasilkan dari Oracle ke PostgreSQL
- Siapkan fungsionalitas Oracle UTL_FILE pada Aurora PostgreSQL yang kompatibel
- Validasi objek database setelah bermigrasi dari Oracle ke Amazon Aurora PostgreSQL

SAP

- Memigrasikan database SAP ASE lokal ke Amazon EC2
- Bermigrasi dari SAP ASE ke Amazon RDS for SQL Server menggunakan AWS DMS
- Migrasikan SAP ASE di Amazon EC2 ke Amazon Aurora PostgreSQL yang kompatibel dengan menggunakan AWS SCT dan AWS DMS

Lebih banyak pola

- Akses Layanan AWS dari IBM z/OS dengan menginstal AWS CLI
- Menilai kesiapan aplikasi untuk migrasi ke AWS Cloud dengan menggunakan CAST Highlight
- Menilai kinerja kueri untuk memigrasi database SQL Server ke MongoDB Atlas di AWS
- Otomatiskan failover dan failback lintas wilayah dengan menggunakan DR Orchestrator Framework
- Mengotomatiskan tugas database dalam edisi SQL Server Express yang berjalan di Amazon EC2 dengan menggunakan AWS Lambda dan Penjadwal Tugas
- Membangun penampil file mainframe tingkat lanjut di AWS Cloud
- Connect ke data Application Migration Service dan kontrol pesawat melalui jaringan pribadi
- Kontainerisasi beban kerja mainframe yang telah dimodernisasi oleh Blu Age
- Mengkonversi kueri JSON Oracle ke database PostgreSQL SQL
- Mengonversi fitur temporal Teradata NORMALIZE ke Amazon Redshift SQL
- Mengonversi fitur Teradata RESET WHEN ke Amazon Redshift SQL
- Salin tabel Amazon DynamoDB di seluruh akun menggunakan AWS Backup
- Terapkan cluster Cassandra di Amazon EC2 dengan statis IPs pribadi untuk menghindari penyeimbangan kembali
- Menerapkan aplikasi multi-stack menggunakan AWS CDK dengan TypeScript
- Menerapkan instance cluster failover SQL Server di Amazon EC2 dan Amazon FSx dengan menggunakan Terraform
- Meniru array PL/SQL asosiatif Oracle di Amazon Aurora PostgreSQL dan Amazon RDS untuk PostgreSQL
- Perkirakan ukuran mesin Amazon RDS untuk database Oracle dengan menggunakan laporan AWR
- Hasilkan wawasan data dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight
- Menangani blok anonim dalam pernyataan SQL Dinamis di Aurora PostgreSQL
- Identifikasi gambar kontainer duplikat secara otomatis saat bermigrasi ke repositori Amazon ECR
- Mengatur cluster failover Microsoft SQL Server di Amazon EC2 menggunakan Windows FSx File Server
- Migrasikan beban kerja Apache Cassandra ke Amazon Keyspaces dengan menggunakan AWS Glue

- Migrasikan beban kerja kontainer Anda dari Azure Red Hat OpenShift (ARO) ke (ROSA Layanan OpenShift Red Hat di AWS)
- Bermigrasi dari Oracle 8i atau 9i ke Amazon RDS untuk menggunakan Oracle dan AWS DMS SharePlex
- Migrasi data Hadoop ke Amazon S3 menggunakan Migrator WANdisco LiveData
- Migrasi grup ketersediaan Microsoft SQL Server Always On menggunakan AWS Application Migration Service
- Migrasikan fungsi dan prosedur Oracle yang memiliki lebih dari 100 argumen ke PostgreSQL
- Migrasikan SAP HANA ke AWS menggunakan SAP HSR dengan nama host yang sama
- Migrasikan SQL Server ke AWS menggunakan grup ketersediaan terdistribusi
- Memodernisasi dan menyebarkan aplikasi mainframe menggunakan dan Terraform AWS Transform
- Modernisasi beban kerja pencetakan online mainframe di AWS dengan menggunakan Micro Focus Enterprise Server dan LRS VPSX/MFI
- Modernisasi manajemen output mainframe AWS dengan menggunakan Rocket Enterprise Server dan LRS X PageCenter
- Ubah header HTTP saat Anda bermigrasi dari F5 ke Application Load Balancer di AWS
- Kelola failover Multi-AZ untuk cluster EMR dengan menggunakan Application Recovery Controller
- Menganalisis dependensi objek untuk migrasi database sebagian dari Oracle ke PostgreSQL
- Siapkan CI/CD pipeline untuk migrasi database dengan menggunakan Terraform
- Siapkan pemulihan bencana untuk Oracle JD Edwards dengan EnterpriseOne AWS Elastic Disaster Recovery
- Sederhanakan manajemen sertifikat pribadi dengan menggunakan AWS Private CA dan AWS RAM
- Transfer z/OS data Db2 skala besar ke Amazon S3 dalam file CSV

Modernisasi

Topik

- Menganalisis dan memvisualisasikan arsitektur perangkat lunak dalam CAST Imaging
- Menilai kesiapan aplikasi untuk migrasi ke AWS Cloud dengan menggunakan CAST Highlight
- Secara otomatis mengarsipkan item ke Amazon S3 menggunakan DynamoDB TTL
- Membangun arsitektur tanpa server multi-tenant di Amazon Service OpenSearch
- Menerapkan aplikasi multi-stack menggunakan AWS CDK dengan TypeScript
- Otomatiskan penerapan aplikasi bersarang menggunakan AWS SAM
- Menerapkan isolasi penyewa SaaS untuk Amazon S3 dengan menggunakan mesin penjual otomatis token AWS Lambda
- Menerapkan pola saga tanpa server dengan menggunakan AWS Step Functions
- Mengelola aplikasi kontainer lokal dengan menyiapkan Amazon ECS Anywhere dengan AWS CDK
- Memodernisasi aplikasi Formulir Web ASP.NET di AWS
- Jalankan beban kerja yang digerakkan oleh peristiwa dan terjadwal dalam skala besar dengan AWS Fargate
- Orientasi penyewa dalam arsitektur SaaS untuk model silo menggunakan C# dan AWS CDK
- Menguraikan monolit menjadi layanan mikro dengan menggunakan CQRS dan sumber acara
- Lebih banyak pola

Menganalisis dan memvisualisasikan arsitektur perangkat lunak dalam CAST Imaging

Arpita Sinha, Amazon Web Services

James Hurrell, Perangkat Lunak Pemeran

Ringkasan

Pola ini menunjukkan bagaimana Anda dapat menggunakan CAST Imaging untuk menavigasi sistem perangkat lunak yang kompleks secara visual dan melakukan analisis yang tepat dari struktur perangkat lunak. Dengan menggunakan CAST Imaging dengan cara ini, Anda dapat membuat keputusan yang lebih tepat tentang arsitektur aplikasi Anda, terutama untuk tujuan modernisasi.

Untuk melihat arsitektur aplikasi Anda di CAST Imaging, Anda harus terlebih dahulu onboard kode sumber aplikasi Anda melalui CAST Console. Konsol kemudian menerbitkan data aplikasi Anda ke CAST Imaging, di mana Anda dapat memvisualisasikan dan menavigasi arsitektur aplikasi Anda lapis demi lapis.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Gambar Mesin Amazon (AMI) untuk Pencitraan CAST
- Instans Amazon Elastic Compute Cloud (Amazon EC2) yang menyertakan hal-hal berikut (disarankan disarankan untuk menginstal instans EC2 Amazon r5.xlarge untuk memori):
  - 4 vCPU
  - RAM 32 GB
  - Volume solid state drive (SSD) (gp3) Tujuan Umum minimum 500 GB
- Kunci lisensi CAST Console dan CAST Imaging (untuk mendapatkan kunci lisensi yang diperlukan, hubungi CAST di aws.contact-me@castsoftware.com)
- Kode sumber lengkap aplikasi yang ingin Anda analisis dalam format terkompresi (.zip)
- Microsoft Edge, Mozilla Firefox, atau Google Chrome

Arsitektur

Diagram berikut menunjukkan contoh alur kerja untuk onboarding kode sumber aplikasi melalui Konsol CAST dan kemudian melihatnya di CAST Imaging:

Diagram menunjukkan alur kerja berikut:

1. CAST menghasilkan metadata kode sumber aplikasi dengan rekayasa balik front-end, middleware, dan kode back-end.
2. Data aplikasi yang dihasilkan oleh CAST secara otomatis diimpor ke CAST Imaging, di mana ia dapat divisualisasikan dan dianalisis.

Berikut adalah cuplikan tentang cara kerja proses ini:

Alat

- CAST Imaging adalah aplikasi berbasis browser yang membantu Anda melihat dan menavigasi sistem perangkat lunak Anda secara visual, sehingga Anda dapat membuat keputusan berdasarkan informasi tentang arsitekturnya.
- CAST Console adalah aplikasi berbasis browser yang membantu Anda mengonfigurasi, menjalankan, dan mengelola analisis CAST AIP.

> **Note**
>
> CAST Imaging dan CAST Console disertakan dalam AMI untuk CAST Imaging.

Epik

Siapkan lingkungan CAST Imaging

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan konfigurasi Konsol CAST awal.	1. Buka browser web Anda dan sambungkan ke Konsol CAST dengan memasukkan URL berikut: http://localhost:8081 2. Saat diminta, masukkan kunci lisensi Konsol CAST Anda. Lalu, pilih Selanjutnya. 3. Tinjau pengaturan konfigurasi. Jika tidak ada perubahan yang diperlukan, pilih Simpan dan Selesai.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis
Jalankan konfigurasi CAST Imaging awal.	1. Buka browser web Anda dan sambungkan ke CAST Imaging dengan memasukkan URL berikut: http://localhost:8083 2. Saat diminta, masuk dengan memasukkan admin untuk nama pengguna dan kata sandi. 3. Saat diminta, masukkan kunci lisensi CAST Imaging Anda. Kemudian, pilih Perbarui untuk menyimpan kuncinya.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan CAST Extend server lokal.	(Opsional) Secara default, CAST Extend server lokal dikonfigurasi untuk berfungsi dalam mode offline. Jika ini dapat diterima, tidak diperlukan konfigurasi tambahan. Namun, jika Anda lebih suka mengkonfigurasi CAST Extend server lokal dalam online/proxy mode dengan koneksi langsung ke CAST Extend, ikuti langkah-langkah ini. **Note** Untuk CAST Extend credentials, lihat halaman pendaftaran CAST Extend. 1. Gunakan pintasan CAST Extend Admin Center di desktop untuk memuat browser web Anda dan terhubung ke server lokal CAST Extend. 2. Pilih opsi Online. 3. Masukkan kredensi CAST Extend Anda (email dan kata sandi), dan pilih Simpan untuk menyelesaikan proses.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis

Mengonboard aplikasi Anda ke CAST Imaging

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kode sumber untuk aplikasi Anda.	Simpan kode sumber aplikasi Anda dalam satu file.zip terkompresi.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis
Tambahkan aplikasi Anda ke Konsol CAST.	1. Buka browser web Anda dan sambungkan ke Konsol CAST dengan memasukkan URL berikut: http://localhost:8081 2. Saat diminta, masuk dengan memasukkan admin untuk nama pengguna dan kata sandi. 3. Pilih Tambahkan aplikasi. Kemudian, masukkan nama aplikasi dan pilih Tambah.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis
Buka wizard pengiriman kode sumber.	Temukan aplikasi yang Anda buat di Konsol CAST. Kemudian, pilih Tambahkan versi.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis
Unggah kode sumber untuk aplikasi Anda.	Lakukan salah satu tindakan berikut: • Seret dan jatuhkan file.zip yang berisi kode sumber aplikasi Anda ke wizard pengiriman kode sumber. —atau— • Pilih ikon unggah cloud. Kemudian, buka file.zip	Arsitek perangkat lunak, Pengembang, Pemimpin teknis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang berisi kode sumber aplikasi Anda.	
Mulai proses analisis.	1. Di panduan pengiriman, berikan detail versi dan tentukan opsi konfigurasi. Untuk informasi selengkapnya, lihat Orientasi standar untuk Pencitraan CAST dalam dokumentasi Pencitraan CAST. 2. Pastikan bahwa opsi Publish to CAST Imaging dipilih. Kemudian, pilih Lanjutkan. **Note** Memilih Proceed memulai proses analisis untuk kode sumber. Jendela kemajuan di Konsol CAST menunjukkan setiap langkah proses analisis, dan menampilkan pemberitahuan saat analisis selesai.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis

Verifikasi hasil analisis dan data yang dipublikasikan ke CAST Imaging

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa status dan log.	Ketika semua tindakan analisis selesai, verifikasi bahwa ada pesan sukses di jendela kemajuan. **Note** Anda dapat memeriksa log individu untuk setiap tindakan analisis segera setelah selesai. Untuk meninjau log untuk tindakan tertentu, pilih Lihat log di jendela Kemajuan.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis
Periksa detail aplikasi.	Di panel Detail aplikasi, tinjau detail tentang hasil analisis. Pastikan untuk melihat teknologi yang ditemukan dan organisasi kode sumber.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis
Verifikasi dan akses CAST Imaging.	1. Di panel Manajemen Aplikasi di Konsol CAST, verifikasi bahwa status versi aplikasi Anda diproses Pencitraan. Ikon CAST Imaging muncul. 2. Pilih ikon CAST Imaging untuk menavigasi langsung ke data aplikasi Anda di CAST Imaging.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Status yang diproses Pencitraan berarti bahwa kode sumber telah dianalisis dan diunggah ke instans CAST Imaging Anda.	

Mulai analisis aplikasi Anda dengan CAST Imaging

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke CAST Imaging.	Buka Cast Imaging dan masukkan kredensi admin default (admin/admin). Data aplikasi Anda muncul.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis
Jelajahi data aplikasi Anda di CAST Imaging.	Mulai melihat arsitektur perangkat lunak Anda dengan menggunakan fitur CAST Imaging. Untuk tutorial singkat tentang cara menggunakan fitur CAST Imaging, pilih ikon Bantuan untuk menampilkan CAST Imaging Helper. Untuk informasi selengkap nya, lihat Panduan Pengguna Pencitraan CAST.	Arsitek perangkat lunak, Pengembang, Pemimpin teknis

Sumber daya terkait

Dokumentasi Konsol CAST

- Login
- Mengkonfigurasi opsi melalui Konsol CAST

Dokumentasi Pencitraan CAST

- Orientasi aplikasi untuk CAST Imaging - prasyarat
- Tambahkan aplikasi baru untuk CAST Imaging
- Orientasi standar untuk CAST Imaging - periksa hasil
- Login
- Opsi konfigurasi - Pusat Admin GUI

Lebih banyak sumber daya tentang Pencitraan CAST di AWS

- Modernisasi Aplikasi ke AWS Dipercepat oleh CAST — Teknis (PartnerCast webinar AWS, memerlukan akun gratis)
- Modernisasi Aplikasi ke Arsitektur AWS dengan CAST Imaging (AWS workshop)
- AWS Marketplace: Pencitraan CAST
- Semua CAST pada sumber daya AWS

Menilai kesiapan aplikasi untuk migrasi ke AWS Cloud dengan menggunakan CAST Highlight

Greg Rivera, Amazon Web Services

Ringkasan

CAST Highlight adalah solusi perangkat lunak sebagai layanan (SaaS) untuk melakukan analisis portofolio aplikasi cepat. Pola ini menjelaskan cara mengonfigurasi dan menggunakan CAST Highlight untuk menilai kesiapan cloud aplikasi perangkat lunak khusus di seluruh portofolio TI organisasi, dan merencanakan modernisasi atau migrasi ke Amazon Web Services (AWS) Cloud.

CAST Highlight menghasilkan wawasan tentang kesiapan cloud aplikasi, mengidentifikasi pemblokir kode yang perlu dihapus sebelum migrasi, memperkirakan upaya untuk menghapus pemblokir ini, dan merekomendasikan layanan AWS yang dapat digunakan aplikasi individual setelah migrasi.

Pola ini menjelaskan prosedur untuk menyiapkan dan menggunakan CAST Highlight, yang terdiri dari lima langkah: penyiapan pengguna baru, manajemen aplikasi, manajemen kampanye, analisis kode sumber, dan analisis hasil. Anda harus menyelesaikan semua langkah di bagian Epik dari pola ini untuk memastikan pemindaian dan analisis aplikasi yang berhasil.

Prasyarat dan batasan

Prasyarat

- Akun CAST Highlight aktif dengan izin Manajer Portofolio.
- Setidaknya 300 MB ruang disk kosong dan memori 4 GB di komputer lokal Anda untuk menginstal CAST Highlight Local Agent.
- Microsoft Windows 8 atau yang lebih baru.
- Kode sumber aplikasi Anda harus disimpan dalam file teks yang dapat diakses dari mesin tempat Agen Lokal diinstal. Tidak ada kode sumber yang meninggalkan tempat dan semua kode dipindai secara lokal.

Arsitektur

Diagram berikut menggambarkan alur kerja untuk menggunakan CAST Highlight.

Alur kerja terdiri dari langkah-langkah berikut:

1. Masuk ke portal CAST Highlight, unduh Agen Lokal, dan instal di komputer lokal Anda. Amazon Simple Storage Service (Amazon S3) menyimpan paket instalasi Agen Lokal.
2. Pindai file kode sumber Anda dan hasilkan file hasil.
3.
> **Important**
>
> Unggah file hasil ke portal CAST Highlight. : Tidak ada kode sumber yang disertakan dalam file hasil.

4. Jawab pertanyaan survei untuk setiap aplikasi yang Anda pindai.
5. Lihat dasbor dan laporan yang tersedia di portal CAST Highlight. Amazon Relational Database Service (Amazon RDS) menyimpan pemindaian kode, hasil analisis, dan data perangkat lunak CAST Highlight.

Tumpukan teknologi

CAST Highlight mendukung teknologi berikut untuk menganalisis kesiapan cloud aplikasi:

- Java
- COBOL
- C#
- C++
- Clojure
- PHP
- JavaScript
- TypeScript
- Python
- Microsoft Transaksi-SQL
- VB.Net
- Kotlin
- Skala
- Cepat

Otomatisasi dan skala

- Penganalisis CLI dapat digunakan untuk mengotomatiskan proses analisis CAST Highlight.

Alat

Tidak ada alat yang diperlukan untuk pola ini jika semua prasyarat terpenuhi. Namun, Anda dapat memilih untuk menggunakan alat opsional, seperti utilitas manajemen kode sumber (SCM), ekstraktor kode, atau alat lain untuk mengelola file kode sumber Anda.

Epik

Penyiapan pengguna baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan akun CAST Highlight Anda dan pilih kata sandi Anda.	Semua pengguna CAST Highlight pertama kali menerima email aktivasi akun. Ikuti tautan aktivasi untuk mengaktifkan akun CAST Highlight Anda dan masukkan kata sandi untuk menyelesa ikan proses aktivasi.	N/A
Masuk ke portal CAST Highlight.	Beranda CAST Highlight muncul setelah Anda memasukkan kata sandi baru Anda. Masuk ke portal CAST Highlight dengan kredensi pengguna Anda.	N/A

Manajemen aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat catatan aplikasi.	Di portal CAST Highlight, navigasikan ke tab Kelola Aplikasi di bagian Kelola	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Portofolio. Di ubin Aplikasi di bagian atas layar, pilih Tambah.	
Pilih nama aplikasi.	Masukkan nama untuk aplikasi Anda, lalu pilih Simpan. Nama ini digunakan untuk catatan aplikasi Anda di CAST Highlight.	N/A
Ulangi langkah-langkah untuk semua aplikasi.	Ulangi langkah-langkah ini untuk setiap aplikasi yang ingin Anda pindai.	N/A

Manajemen kampanye

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kampanye.	CAST Highlight menggunakan “campaign” untuk menggambarkan serangkaian aplikasi yang akan dianalisis pada waktu tertentu. Di portal Sorotan CAST, arahkan ke tab Kelola Kampanye di bagian Kelola Portofolio. Pilih Buat Kampanye untuk meluncurkan layar pembuatan kampanye.	N/A
Masukkan nama dan pilih tanggal penutupan kampanye.	Masukkan nama untuk kampanye Anda dan pilih tanggal penutupan untuk kampanye tersebut.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important Kontributor tidak dapat mengirimkan hasil analisis aplikasi setelah tanggal penutupan kampanye.	
Putuskan untuk menyertakan pemindaian kode sumber, jawaban survei, dan lingkup domain dan aplikasi.	Pilih satu atau lebih survei standar yang digunakan untuk meningkatkan data analisis kode sumber dengan informasi kualitatif. Kategori survei adalah Dampak bisnis, upaya pemeliharaan perangkat lunak CloudReady, properti Aplikasi, dan Dampak Hijau. Pilih domain dan aplikasi yang dianalisis selama kampanye. **Important** Pastikan Anda menambahkan semua aplikasi yang ingin Anda pindai di bagian Kelola Aplikasi sebelum memulai kampanye.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan pesan peluncuran.	Sesuaikan pesan peluncuran yang akan dikirim melalui email ke semua kontributor yang terkait dengan aplikasi dalam kampanye.	N/A
Luncurkan kampanye.	Pilih Lengkap untuk meluncurkan kampanye.	N/A

Analisis kode sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh Agen Lokal Sorotan CAST.	Di portal CAST Highlight , pilih Pemindaian Aplikasi dan unduh Agen Lokal ke komputer lokal Anda.	N/A
Instal Agen Lokal.	Luncurkan program instalasi CASTHighlight Setup.exe dan ikuti petunjuk pengaturan yang muncul. Setelah Agen Lokal diinstal, Anda siap untuk menganalisis aplikasi Anda.	N/A
Tentukan ruang lingkup pemindaian kode Agen Lokal.	Analisis kode dilakukan pada tingkat file dan tidak mempertimbangkan tautan logis atau dependensi di antara file. Semua file dianggap sama dan bagian dari aplikasi. Untuk memberikan hasil yang akurat dan konsisten, siapkan	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	cakupan pemindaian kode Anda dengan menggunakan fitur pengecualian file atau folder yang tersedia di Agen Lokal.	
Sertakan paket open-source atau COTS.	(Opsional) Jika Anda ingin menyertakan paket open-source atau komersial off-the-shelf (COTS), pastikan paket tersebut disertakan dalam folder yang ingin Anda pindai. Biasanya, pustaka eksternal dikelompokkan dalam subfolder yang disebut “pihak ketiga” atau yang serupa, dan kode utama sering terletak di folder file “src/main”.	N/A
Kecualikan kelas uji.	Kelas pengujian biasanya dikecualikan dari analisis kode sumber karena umumnya bukan bagian dari aplikasi yang dikompilasi. Namun, Anda dapat memilih untuk memasukkannya ke dalam pemindaian jika diperlukan.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kecualikan folder SCM, build, dan deployment.	Untuk hasil yang lebih konsisten, Anda harus menghindari menyertak an folder SCM, build, atau deployment (misalnya, file.git atau .svn) dalam pemindaian Anda.	N/A
Sertakan file ketergantungan.	Jika Anda menginginkan wawasan tentang kerangka kerja dan dependensi yang file fisiknya bukan bagian dari folder yang Anda pindai, pastikan Anda menyertak an file dependensi (seperti file pom.xml, build.gradle, package.json, atau .vcsproj).	N/A
Panggil Agen Lokal.	Jalankan Agen Lokal di mesin Windows lokal Anda.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih folder yang berisi kode sumber Anda.	Pilih folder yang berisi kode sumber Anda. Anda dapat menambahkan beberapa folder untuk ditemukan oleh Agen Lokal. Meskipun Agen Lokal mendukung penemuan sumber melalui jalur jaringan, Anda harus memastikan folder sumber berada di komputer lokal Anda. **Important** Sebaiknya jalankan beberapa pemindaian jika ada lebih dari 10.000 file di folder sumber Anda.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai penemuan file.	Di dasbor Agen Lokal, pilih Temukan File. Agen Lokal menemukan file di folder dan subfolder Anda, dan mendeteksi teknologinya. Anda dapat memilih tombol Batal untuk membatalkan penemuan kapan saja. Setelah penemuan file selesai, Agen Lokal mencantum kan folder dan file yang ditemukan. Kolom Technolog ies menunjukkan teknologi terkait dan jumlah file. Kolom Path menunjukkan lokasi folder dan file.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbaiki konfigurasi pemindaian kode sumber.	(Opsional) Untuk menyempurnakan pemindaian Agen Lokal, Anda dapat menonaktifkan satu atau beberapa teknologi untuk folder atau file tertentu. Jika semua teknologi dinonaktifkan, folder atau file Anda akan dikecualikan dari ruang lingkup pemindaian. Untuk menonaktifkan teknologi, pilih label kuning dari teknologi yang ingin Anda nonaktifkan. Anda juga dapat memilih ikon filter saat mengarahkan kursor ke file atau folder untuk mengaitkan teknologi dengan file atau folder tertentu. Pengaturan ini disimpan dan mempercepat proses penemuan untuk folder atau file.	N/A
Mulai pemindaian kode sumber.	Setelah Anda mengkonfigurasi pemindaian Anda, pilih “Pindai File” untuk memulai proses pemindaian.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa label hijau atau abu-abu.	Setelah pemindaian kode sumber selesai, label status ditampilkan di tingkat folder dan file. Label hijau berarti bahwa file dipindai dengan benar dengan teknologi terkait. Label abu-abu berarti file tidak dipindai dan dikecuali kan. Alasan pengecualian mereka ditampilkan saat Anda mengarahkan kursor ke label setiap file. Kemungkin an alasan pengecualian file termasuk file biner, file yang tidak dapat dibaca, file yang hilang, pustaka eksternal , file yang disandikan, file yang dihasilkan, kesalahan sintaks, konten yang tidak dalam bahasa yang diharapka n, kode yang tidak sesuai dengan kriteria analisis yang cukup, file yang melebihi batas ukuran (10 MB), masalah batas waktu, atau tidak tersedianya penganalisis.	N/A
Ubah konfigurasi pemindaian dan pindai kode lagi.	(Opsional) Anda dapat mengubah pengaturan konfigurasi pemindaian Anda dan memilih Pindai File untuk memindai file lagi.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan hasil pemindaian.	Pilih Konfirmasi Hasil jika hasil pemindaian memenuhi persyaratan Anda.	N/A
Lihat kerangka kerja dan pustaka perangkat lunak yang ditemukan oleh Agen Lokal.	Lihat kerangka kerja dan pustaka perangkat lunak yang digunakan atau direferensikan oleh aplikasi Anda, dan ditemukan oleh Agen Lokal selama pemindaian kode. Anda dapat menyimpan atau mengabaikan elemen dari daftar ini dengan memilih tombol sakelar masing-masing. Pilih Konfirmasi dependensi untuk melanjutkan. **Important** Jika kerangka kerja dimatikan, itu tidak tercantum di portal CAST Highlight atau dilampirkan ke aplikasi Anda.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Simpan hasil pemindaian kode.	Agen Lokal menampilkan ringkasan hasil pemindaian kode Anda yang dikelompokkan berdasarkan teknologi. Pilih Simpan dan tentukan folder yang Anda inginkan agar hasilnya disimpan. Agen Lokal menghasilkan satu file.zip per pemindaian, yang berisi semua hasil analisis. Bergantung pada jumlah teknologi dan folder sumber root yang berbeda, Agen Lokal secara otomatis menghasilkan satu atau beberapa file.csv dengan struktur FolderNamepenamaan.technology.date.csv.	N/A
Unggah hasil pemindaian kode ke portal CAST Highlight.	Di portal CAST Highlight, pilih aplikasi yang Anda analisis di bagian Pemindaian Aplikasi. Pilih Unggah Hasil dan pilih file.csv. Anda juga dapat mengunggah file.csv satu per satu. Setelah setiap file diunggah, catatan unggahan muncul di layar Anda.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus file hasil analisis, jika diperlukan.	(Opsional) File hasil analisis dapat dihapus kapan saja selama proses pengunggahan dengan memilih ikon tempat sampah. **⚠ Important** Hanya pengguna dengan hak istimewa Manajer Portofoli o atau kontributor yang mengungga h hasil yang dapat menghapus hasilnya.	N/A
Jawab survei aplikasi.	Tombol Survei muncul pada aplikasi yang memerlukan survei. Pilih Survei, jawab pertanyaan untuk setiap bagian survei, dan pilih Kirim setelah Anda selesai. Kemajuan survei Anda ditampilkan di bagian atas layar Anda. Anda dapat mengirimkan hasil Anda setelah semua informasi wajib dikirimkan. Namun, Anda dapat memperkaya data dalam instance CAST Highlight organisasi Anda dengan menjawab semua pertanyaan.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim hasil pemindaian kode.	Setelah Anda mengunggah semua file hasil .csv untuk aplikasi dan menyelesaikan pertanyaan survei, pilih Kirim di bagian Pemindaian Aplikasi. Langkah ini diperlukan untuk menyelesaikan proses dan memastikan bahwa hasilnya tersedia di portal CAST Highlight.	N/A

Analisis hasil

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat beranda portal CAST Highlight.	Beranda portal CAST Highlight mencakup ubin yang memiliki informasi tingkat tinggi tentang portofolio aplikasi Anda, seperti kesehatan perangkat lunak CloudReady, dan skor keamanan sumber terbuka untuk seluruh portofolio Anda. Beranda juga mencakup jumlah aplikasi onboard. Untuk informasi selengkap nya tentang definisi metrik dan metodologi pengukura n CAST Highlight, lihat CAST Highlight — Metrik dan metodologi (presentasi Microsoft PowerPoint).	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat CloudReady dasbor.	Pilih CloudReady ubin untuk membuka CloudReady dasbor. Ini adalah dasbor tingkat portofolio utama untuk menilai kesiapan cloud aplikasi Anda. Ini membantu Anda merencanakan dan mengembangkan peta jalan portofolio untuk migrasi cloud Anda	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat Portofolio Advisor untuk dasbor Cloud.	Dasbor Portofolio Advisor for Cloud secara otomatis mengelompokkan aplikasi ke dalam kategori migrasi yang direkomendasikan. Segmentasi didasarkan pada karakteristik teknis dari setiap aplikasi. Faktor-fa ktor termasuk analisis kode sumber (kesiapan cloud, ketahanan perangkat lunak, dan banyak lagi) dan dampak bisnis, yang berasal dari survei. Di kanan atas, pilih Compute untuk menghasil kan rekomendasi segmentasi awal. Gelembung di bagan di bagian atas dasbor mewakili setiap aplikasi dalam portofolio, yang diatur oleh segmentasi yang direkomendasikan. Setiap aplikasi juga tercantum dalam tabel data di bawah grafik, termasuk metrik yang relevan untuk setiap aplikasi. Segmen yang mungkin direkomendasikan meliputi: • Rehost — Rekomendasi untuk mengubah konfigura si infrastruktur aplikasi untuk mengangkat dan	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menggesernya ke cloud dengan menggunakan solusi infrastruktur sebagai layanan (IaaS). • Refactor — Rekomendasi untuk melakukan modifikasi sederhana dari kode aplikasi tanpa mengubah arsitektur atau fungsionalitas sehingga dapat dimigrasikan dengan menggunakan solusi container as a service (CaaS) atau platform as a service (PaaS). • Rearchitect — Rekomendasi untuk memodifikasi kode aplikasi secara dramatis untuk meningkatkan kesehatan aplikasi dan mempersiapkannya untuk migrasi dengan menggunakan solusi PaaS atau menerapkannya sebagai aplikasi tanpa server dengan menggunakan solusi fungsi sebagai layanan (FaaS). • Membangun kembali — Rekomendasi untuk membuang kode aplikasi dan mengembangkannya lagi di cloud dengan menggunakan solusi PaaS	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	atau mengembangkannya lagi sebagai aplikasi tanpa server dengan menggunakan solusi FaaS. • Pensiun — Rekomendasi untuk membuang aplikasi sama sekali atau berpotensi menggantinya dengan alternatif perangkat lunak komersial sebagai layanan (SaaS).	
Memodifikasi rekomendasi segmentasi.	Dalam beberapa kasus, Anda mungkin memilih untuk mengubah segmen yang direkomendasikan oleh CAST Highlight. Anda dapat melakukan ini dengan menjelajah ke aplikasi di tabel data dan memilih segmen yang berbeda dari daftar drop-down di sebelah nama aplikasi. Kemudian pilih Simpan di kanan atas untuk menyimpan perubahan Anda. Anda juga dapat mengekspor data ini kapan saja dengan memilih Ekspor di kanan atas.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih aplikasi untuk dianalisis.	Pada dashboard Portofolio Advisor for Cloud, pilih gelembung aplikasi untuk menganalisis aplikasi tersebut. Pilih nama aplikasi dalam tabel setelah bagan gelembung untuk memulai analisis yang lebih dalam. Dasbor yang berbeda tersedia untuk menganalisis aplikasi individual, seperti Code Insights (pola kesehatan perangkat lunak), Tren, dan Komposisi Perangkat Lunak (risiko sumber terbuka).	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menganalisis CloudReady hasil aplikasi individu.	Pilih CloudReadytab, yang menunjukkan CloudReady skor keseluruhan aplikasi. Skor ini adalah rata-rata tertimbang berdasark an kombinasi jawaban CloudReady survei dan pemindaian CloudReady kode. Jawaban atas pertanyaan survei muncul pada tabel di bawah ubin. Pilih Pemindaian CloudRead y Kode untuk melihat hasil pemindaian kode. Ada daftar CloudReady pola yang kode aplikasi dipindai. Daftar ini mencakup kolom berikut: • Cloud Requirement adalah pola kode tertentu. • Teknologi adalah bahasa pemrograman pola. “Dampak” adalah dampak pola pada aplikasi (C = kode, F = kerangka kerja, A = arsitektur). • Kritikalitas adalah tingkat pentingnya mengatasi pola ini sebelum bermigrasi. • Kontribusi adalah bagaimana pola ini berkontri busi pada CloudReady skor keseluruhan. Jika polanya	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	hijau, itu adalah booster dan meningkatkan CloudRead y skor. Jika polanya merah, itu adalah pemblokir dan menurunkan CloudReady skor. Jika pola tidak memiliki warna, itu adalah pemblokir yang tidak terdeteksi dan meningkatkan CloudReady skor. • Penghalang jalan adalah jumlah kemunculan individu dari pola pemblokir. Pilih nomor penghalang jalan untuk menampilkan daftar file kode sumber tempat pola terdeteksi. • Est. Upaya adalah perkiraan jumlah hari yang diperluka n untuk memulihkan hambatan di setiap baris.	
Ekspor data ke Microsoft Excel.	(Opsional) Pilih Ekspor ke Excel untuk mengekspo r data untuk analisis lebih lanjut. Data hasil analisis aplikasi dapat digunakan untuk menganalisis lebih lanjut kesiapan cloud suatu aplikasi dan menentukan kode apa yang harus diperbarui sebelum migrasi.	N/A

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat rekomendasi.	Pilih Rekomendasi di samping Pemindaian CloudReady Kode untuk melihat layar Rekomendasi Layanan Cloud. Ini mengidentifikasi layanan AWS yang dapat diadopsi aplikasi berdasarkan karakteri stiknya. Ulangi langkah ini untuk melihat rekomendasi untuk semua aplikasi yang Anda analisis.	N/A

Sumber daya terkait

Manajemen kampanye

- CAST Highlight Foundation Pelatihan Sertifikasi Bagian 3: Konfigurasi Portofolio (video)

Analisis kode sumber

- CAST Highlight Foundation Pelatihan Sertifikasi Bagian 4: Analisis Aplikasi (video)

Sumber daya lainnya

- Sorotan CAST di AWS Marketplace
- AWS dan CAST: Mempercepat modernisasi aplikasi
- CAST Highlight - Dokumentasi, tutorial produk, dan alat pihak ketiga
- CAST Sorotan - Demo Produk Kesiapan Cloud (video)
- Modernisasi Portofolio Aplikasi dengan CAST Highlight (AWS workshop)

Secara otomatis mengarsipkan item ke Amazon S3 menggunakan DynamoDB TTL

Tabby Ward, Amazon Web Services

Ringkasan

Pola ini menyediakan langkah-langkah untuk menghapus data lama dari tabel Amazon DynamoDB dan mengarsipkannya ke bucket Amazon Simple Storage Service (Amazon S3) di Amazon Web Services (AWS) tanpa harus mengelola armada server.

Pola ini menggunakan Amazon DynamoDB Time to Live (TTL) untuk secara otomatis menghapus item lama dan Amazon DynamoDB Streams untuk menangkap item kedaluwarsa TTL. Kemudian menghubungkan DynamoDB Streams ke AWS Lambda, yang menjalankan kode tanpa menyediakan atau mengelola server apa pun.

Saat item baru ditambahkan ke aliran DynamoDB, fungsi Lambda dimulai dan menulis data ke aliran pengiriman Amazon Data Firehose. Firehose menyediakan solusi sederhana dan terkelola sepenuhnya untuk memuat data sebagai arsip ke Amazon S3.

DynamoDB sering digunakan untuk menyimpan data time series, seperti data klik-aliran halaman web atau data Internet of Things (IoT) dari sensor dan perangkat yang terhubung. Daripada menghapus item yang jarang diakses, banyak pelanggan ingin mengarsipkannya untuk tujuan audit. TTL menyederhanakan pengarsipan ini dengan secara otomatis menghapus item berdasarkan atribut timestamp.

Item yang dihapus oleh TTL dapat diidentifikasi di DynamoDB Streams, yang menangkap urutan modifikasi tingkat item yang diurutkan waktu dan menyimpan urutan dalam log hingga 24 jam. Data ini dapat dikonsumsi oleh fungsi Lambda dan diarsipkan dalam bucket Amazon S3 untuk mengurangi biaya penyimpanan. Untuk mengurangi biaya lebih lanjut, aturan siklus hidup Amazon S3 dapat dibuat untuk mentransisikan data secara otomatis (segera setelah dibuat) ke kelas penyimpanan dengan biaya terendah.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- AWS Command Line Interface (AWS CLI) 1.7 atau yang lebih baru, diinstal dan dikonfigurasi di macOS, Linux, atau Windows.

- Python 3.7 atau yang lebih baru.
- Boto3, diinstal dan dikonfigurasi. Jika Boto3 belum diinstal, jalankan `python -m pip install boto3` perintah untuk menginstalnya.

Arsitektur

Tumpukan teknologi

- Amazon DynamoDB
- Amazon DynamoDB Streams
- Amazon Data Firehose
- AWS Lambda
- Amazon S3

1. Item dihapus oleh TTL.
2. Pemicu aliran DynamoDB memanggil fungsi prosesor aliran Lambda.
3. Fungsi Lambda menempatkan catatan dalam aliran pengiriman Firehose dalam format batch.
4. Catatan data diarsipkan dalam bucket S3.

Alat

- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat terpadu untuk mengelola layanan AWS Anda.
- Amazon DynamoDB - Amazon DynamoDB adalah database nilai kunci dan dokumen yang memberikan kinerja milidetik satu digit pada skala apa pun.
- Amazon DynamoDB Time to Live (TTL) - Amazon DynamoDB TTL membantu Anda menentukan stempel waktu per item untuk menentukan kapan item tidak lagi diperlukan.
- Amazon DynamoDB Streams - Amazon DynamoDB Streams menangkap urutan modifikasi tingkat item yang diurutkan waktu di tabel DynamoDB apa pun dan menyimpan informasi ini dalam log hingga 24 jam.
- Amazon Data Firehose — Amazon Data Firehose adalah cara termudah untuk memuat data streaming secara andal ke dalam data lake, penyimpanan data, dan layanan analitik.

- AWS Lambda — AWS Lambda menjalankan kode tanpa perlu menyediakan atau mengelola server. Anda hanya membayar untuk waktu komputasi yang Anda konsumsi.
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri.

Kode

Kode untuk pola ini tersedia di item GitHub Arsip ke S3 menggunakan repositori DynamoDB TTL.

Epik

Siapkan tabel DynamoDB, TTL, dan aliran DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tabel DynamoDB.	Gunakan AWS CLI untuk membuat tabel di DynamoDB yang disebut. `Reservation` Pilih unit kapasitas baca acak (RCU) dan unit kapasitas tulis (WCU), dan berikan tabel Anda dua atribut: `ReservationID` dan. `ReservationDate` `aws dynamodb create-table \ --table-name Reservation \ --attribute-definitions AttributeName=ReservationID,AttributeType=S AttributeName=ReservationDate,AttributeType=N \ --key-schema AttributeName=ReservationID`	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>,KeyType=HASH
	 AttributeName=Rese
	rvationDate,KeyTyp
	e=RANGE \
	--provisioned-th
	roughput ReadCapac
	ityUnits=100,Write
	CapacityUnits=100</pre>
	`ReservationDate` adalah stempel waktu epoch yang akan digunakan untuk menyalakan TTL.	

Nyalakan DynamoDB TTL.	Gunakan AWS CLI untuk mengaktifkan DynamoDB TTL untuk atribut. `ReservationDate`
	<pre>aws dynamodb update-ti
	me-to-live \
	--table-name Reservati
	on\
	  --time-to-live-spe
	cification Enabled=t
	rue,AttributeName=
	ReservationDate</pre>
		Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Nyalakan aliran DynamoDB.	Gunakan AWS CLI untuk mengaktifkan aliran DynamoDB untuk `Reservation` tabel dengan menggunakan jenis aliran. `NEW_AND_OLD_IMAGES`
	
	`aws dynamodb update-table \`
	`--table-name Reservation \`
	`  --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES`
	
	Aliran ini akan berisi catatan untuk item baru, item yang diperbarui, item yang dihapus, dan item yang dihapus oleh TTL. Catatan untuk item yang dihapus oleh TTL berisi atribut metadata tambahan untuk membedakannya dari item yang dihapus secara manual. `userIdentity` Bidang untuk penghapusan TTL menunjukkan bahwa layanan DynamoDB melakukan tindakan penghapusan.
	
	Dalam pola ini, hanya item yang dihapus oleh TTL yang diarsipkan, tetapi Anda hanya dapat mengarsipkan catatan
		Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di mana `eventName` ada `REMOVE` dan `userIdentity` berisi `principalId` sama dengan. `dynamodb.amazonaws.com`	

Buat dan konfigurasikan bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Gunakan AWS CLI untuk membuat bucket S3 tujuan di Wilayah AWS Anda, ganti `us-east-1` dengan Region dan amzn-s3- demo-destination-bucket dengan nama bucket Anda. `aws s3api create-bucket \ --bucket amzn-s3-demo-destination-bucket \ --region us-east-1` Pastikan nama bucket S3 Anda unik secara global, karena namespace dibagikan oleh semua akun AWS.	Arsitek cloud, Pengembang aplikasi
Buat kebijakan siklus hidup 30 hari untuk bucket S3.	1. Masuk ke AWS Management Console dan buka konsol Amazon S3. 2. Pilih bucket S3 yang berisi data dari Firehose.	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di bucket S3, pilih tab Manajemen, dan pilih Tambahkan aturan siklus hidup. 4. Masukkan nama aturan Anda di kotak dialog Aturan Siklus Hidup, dan konfigura sikan aturan siklus hidup 30 hari untuk bucket Anda.	

Membuat aliran pengiriman Firehose

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan aliran pengiriman Firehose.	Unduh dan edit contoh `CreateFireHoseToS3.py` kode dari GitHub repositori. Kode ini ditulis dengan Python dan menunjukkan cara membuat aliran pengirima n Firehose dan peran AWS Identity and Access Management (IAM). Peran IAM akan memiliki kebijakan yang dapat digunakan oleh Firehose untuk menulis ke bucket S3 tujuan. Untuk menjalankan skrip, gunakan argumen perintah dan baris perintah berikut.	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Argumen 1=<Your_S3_bucket_ARN> , yang merupakan Amazon Resource Name (ARN) untuk bucket yang Anda buat sebelumnya Argumen 2= Nama Firehose Anda (Pilot ini `firehose_to_s3_stream` menggunakan.) Argumen 3= Nama peran IAM Anda (Pilot ini menggunak an`firehose_to_s3` .) `python CreateFireHoseToS3.py <Your_S3_Bucket_ARN> firehose_to_s3_stream firehose_to_s3` Jika peran IAM yang ditentukan tidak ada, skrip akan membuat peran asumsi dengan kebijakan hubungan tepercaya, serta kebijakan yang memberikan izin Amazon S3 yang memadai. Untuk contoh kebijakan ini, lihat bagian Informasi tambahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi aliran pengiriman Firehose.	Jelaskan aliran pengiriman Firehose dengan menggunakan AWS CLI untuk memverifikasi bahwa aliran pengiriman berhasil dibuat. `aws firehose describe-delivery-stream --delivery-stream-name firehose_to_s3_stream`	Arsitek cloud, Pengembang aplikasi

Buat fungsi Lambda untuk memproses aliran pengiriman Firehose

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan kepercayaan untuk fungsi Lambda.	Buat file kebijakan kepercayaan dengan informasi berikut. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }`	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Ini memberi izin fungsi Anda untuk mengakses sumber daya AWS.	
Buat peran eksekusi untuk fungsi Lambda.	Untuk membuat peran eksekusi, jalankan kode berikut. `aws iam create-role --role-name lambda-ex --assume-role-policy-document file://TrustPolicy.json`	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tambahkan izin ke peran.	Untuk menambahkan izin ke peran, gunakan `attach-policy-to-role` perintah.
	`aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole`
	`aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole`
	`aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/AmazonKinesisFirehoseFullAccess`
	`aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/IAMFullAccess`
		Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat fungsi Lambda.	Kompres `LambdaStreamProcessor.py` file dari repositori kode dengan menjalankan perintah berikut.
	`zip function.zip LambdaStreamProcessor.py`
	Saat Anda membuat fungsi Lambda, Anda akan memerlukan peran eksekusi Lambda ARN. Untuk mendapatkan ARN, jalankan kode berikut.
	`aws iam get-role \ --role-name lambda-ex`
	Untuk membuat fungsi Lambda, jalankan kode berikut.
	`# Review the environment variables and replace them with your values.`
	`aws lambda create-function --function-name LambdaStreamProcessor \ --zip-file fileb://function.zip --handler LambdaStreamProcessor.handler --runtime python3.8 \`
		Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--role {Your Lamda Execution Role ARN}\ --environment Variables="{firehose_name=firehose_to_s3_stream,bucket_arn = <Your_S3_bucket_ARN>,iam_role_name = firehose_to_s3, batch_size=400}"`	
Konfigurasikan pemicu fungsi Lambda.	Gunakan AWS CLI untuk mengonfigurasi pemicu (DynamoDB Streams), yang memanggil fungsi Lambda. Ukuran batch 400 adalah untuk menghindari masalah konkurensi Lambda. `aws lambda create-event-source-mapping --function-name LambdaStreamProcessor \ --batch-size 400 --starting-position LATEST \ --event-source-arn <Your Latest Stream ARN From DynamoDB Console>`	Arsitek cloud, Pengembang aplikasi

Uji fungsionalitasnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan item dengan cap waktu kedaluwarsa ke tabel Reservasi.	Untuk menguji fungsiona litas, tambahkan item dengan stempel waktu epoch kedaluwarsa ke tabel. `Reservation` TTL akan secara otomatis menghapus item berdasarkan stempel waktu. Fungsi Lambda dimulai pada aktivitas DynamoDB Stream, dan memfilter acara untuk mengidentifikasi aktivitas atau item yang dihapus. `REMOVE` Kemudian menempatkan catatan dalam aliran pengirima n Firehose dalam format batch. Aliran pengiriman Firehose mentransfer item ke bucket S3 tujuan dengan awalan. `firehosetos3exampl e/year=current year/ month=current month/ day=current day/hour= current hour/` **Important** Untuk mengoptim alkan pengambil an data, konfigura sikan Amazon S3	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan `ErrorOutputPrefix` dan `Prefix` yang dirinci di bagian Informasi tambahan.	

Bersihkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus semua sumber daya.	Hapus semua sumber daya untuk memastikan bahwa Anda tidak dikenakan biaya untuk layanan apa pun yang tidak Anda gunakan.	Arsitek cloud, Pengembang aplikasi

Sumber daya terkait

- Mengelola siklus hidup penyimpanan
- Kelas Penyimpanan Amazon S3
- Dokumentasi AWS SDK untuk Python (Boto3)

Informasi tambahan

Membuat dan mengonfigurasi aliran pengiriman Firehose — Contoh kebijakan

Dokumen contoh kebijakan hubungan tepercaya Firehose

```
firehose_assume_role = {
        'Version': '2012-10-17',
        'Statement': [
            {
                'Sid': '',
                'Effect': 'Allow',
```

```
                'Principal': {
                    'Service': 'firehose.amazonaws.com'
                },
                'Action': 'sts:AssumeRole'
            }
        ]
    }
```

Contoh kebijakan izin S3

```
s3_access = {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "",
                "Effect": "Allow",
                "Action": [
                    "s3:AbortMultipartUpload",
                    "s3:GetBucketLocation",
                    "s3:GetObject",
                    "s3:ListBucket",
                    "s3:ListBucketMultipartUploads",
                    "s3:PutObject"
                ],
                "Resource": [
                    "{your s3_bucket ARN}/*",
                    "{Your s3 bucket ARN}"
                ]
            }
        ]
    }
```

Uji fungsionalitas - konfigurasi Amazon S3

Konfigurasi Amazon S3 dengan yang berikut ini `Prefix` dan `ErrorOutputPrefix` dipilih untuk mengoptimalkan pengambilan data.

prefix

```
firehosetos3example/year=! {timestamp: yyyy}/month=! {timestamp:MM}/day=!
 {timestamp:dd}/hour=!{timestamp:HH}/
```

Firehose pertama kali membuat folder dasar yang disebut `firehosetos3example` langsung di bawah bucket S3. Kemudian mengevaluasi ekspresi`!{timestamp:yyyy}`,,`!{timestamp:MM}`, dan `!{timestamp:HH}` ke tahun`!{timestamp:dd}`, bulan, hari, dan jam menggunakan DateTimeFormatterformat Java.

Misalnya, perkiraan timestamp kedatangan 1604683577 dalam waktu zaman Unix mengevaluasi,,, dan. `year=2020` `month=11` `day=06` `hour=05` Oleh karena itu, lokasi di Amazon S3, tempat catatan data dikirimkan, dievaluasi. `firehosetos3example/year=2020/month=11/day=06/hour=05/`

ErrorOutputPrefix

```
firehosetos3erroroutputbase/!{firehose:random-string}/!{firehose:error-output-type}/!
{timestamp:yyyy/MM/dd}/
```

`ErrorOutputPrefix`Hasilnya dalam folder dasar yang dipanggil `firehosetos3erroroutputbase` langsung di bawah bucket S3. Ekspresi `!{firehose:random-string}` mengevaluasi ke string acak 11 karakter seperti. `ztWxkdg3Thg` Lokasi untuk objek Amazon S3 tempat catatan gagal dikirimkan dapat dievaluasi. `firehosetos3erroroutputbase/ztWxkdg3Thg/processing-failed/2020/11/06/`

Membangun arsitektur tanpa server multi-tenant di Amazon Service OpenSearch

Tabby Ward dan Nisha Gambhir, Amazon Web Services

Ringkasan

Amazon OpenSearch Service adalah layanan terkelola yang memudahkan penerapan, pengoperasian, dan skala Elasticsearch, yang merupakan mesin pencari dan analitik sumber terbuka yang populer. OpenSearch Layanan menyediakan pencarian teks gratis serta konsumsi dan dasbor mendekati waktu nyata untuk streaming data seperti log dan metrik.

Penyedia perangkat lunak sebagai layanan (SaaS) sering menggunakan OpenSearch Layanan untuk mengatasi berbagai kasus penggunaan, seperti mendapatkan wawasan pelanggan dengan cara yang terukur dan aman sekaligus mengurangi kompleksitas dan waktu henti.

Menggunakan OpenSearch Layanan di lingkungan multi-penyewa memperkenalkan serangkaian pertimbangan yang memengaruhi partisi, isolasi, penyebaran, dan pengelolaan solusi SaaS Anda. Penyedia SaaS harus mempertimbangkan cara menskalakan cluster Elasticsearch mereka secara efektif dengan beban kerja yang terus berubah. Mereka juga perlu mempertimbangkan bagaimana kondisi tetangga yang berjenjang dan bising dapat memengaruhi model partisi mereka.

Pola ini meninjau model yang digunakan untuk mewakili dan mengisolasi data penyewa dengan konstruksi Elasticsearch. Selain itu, pola berfokus pada arsitektur referensi tanpa server sederhana sebagai contoh untuk menunjukkan pengindeksan dan pencarian menggunakan OpenSearch Layanan di lingkungan multi-penyewa. Ini mengimplementasikan model partisi data pool, yang berbagi indeks yang sama di antara semua penyewa sambil mempertahankan isolasi data penyewa. Pola ini menggunakan AWS layanan berikut: Amazon API Gateway, AWS Lambda, Amazon Simple Storage Service (Amazon S3), dan Service. OpenSearch

Untuk informasi selengkapnya tentang model kumpulan dan model partisi data lainnya, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Command Line Interface (AWS CLI) versi 2.x, diinstal dan dikonfigurasi di macOS, Linux, atau Windows

- Python versi 3.9
- pip3 — Kode sumber Python disediakan sebagai file.zip untuk digunakan dalam fungsi Lambda. Jika Anda ingin menggunakan kode secara lokal atau menyesuaikannya, ikuti langkah-langkah berikut untuk mengembangkan dan mengkompilasi ulang kode sumber:
  1. Hasilkan `requirements.txt` file dengan menjalankan perintah berikut di direktori yang sama dengan skrip Python: `pip3 freeze > requirements.txt`
  2. Instal dependensi: `pip3 install -r requirements.txt`

Batasan

- Kode ini berjalan dengan Python, dan saat ini tidak mendukung bahasa pemrograman lainnya.
- Aplikasi sampel tidak menyertakan dukungan AWS Cross-region atau Disaster Recovery (DR).
- Pola ini dimaksudkan untuk tujuan demonstrasi saja. Ini tidak dimaksudkan untuk digunakan dalam lingkungan produksi.

Arsitektur

Diagram berikut menggambarkan arsitektur tingkat tinggi dari pola ini. Arsitekturnya meliputi:

- Lambda untuk mengindeks dan menanyakan konten
- OpenSearch Layanan untuk melakukan pencarian
- API Gateway untuk menyediakan interaksi API dengan pengguna
- Amazon S3 untuk menyimpan data mentah (tidak diindeks)
- Amazon CloudWatch untuk memantau log
- AWS Identity and Access Management (IAM) untuk membuat peran dan kebijakan penyewa

Otomatisasi dan skala

Untuk kesederhanaan, pola menggunakan AWS CLI untuk menyediakan infrastruktur dan untuk menyebarkan kode sampel. Anda dapat membuat CloudFormation template atau AWS Cloud Development Kit (AWS CDK) skrip untuk mengotomatiskan pola.

Alat

Layanan AWS

- AWS CLIadalah alat terpadu untuk mengelola Layanan AWS dan sumber daya dengan menggunakan perintah di shell baris perintah Anda.
- Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- API Gateway adalah Layanan AWS untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs pada skala apa pun.
- Amazon S3 adalah layanan penyimpanan objek yang memungkinkan Anda menyimpan dan mengambil sejumlah informasi kapan saja, dari mana saja di web.
- OpenSearch Layanan adalah layanan yang dikelola sepenuhnya yang memudahkan Anda untuk menerapkan, mengamankan, dan menjalankan Elasticsearch dengan biaya efektif dalam skala besar.

Kode

Lampiran menyediakan file sampel untuk pola ini. Ini termasuk:

- `index_lambda_package.zip`— Fungsi Lambda untuk mengindeks data di OpenSearch Layanan dengan menggunakan model pool.
- `search_lambda_package.zip`— Fungsi Lambda untuk mencari data di OpenSearch Layanan.
- `Tenant-1-data`— Sampel data mentah (tidak diindeks) untuk Penyewa - 1.
- `Tenant-2-data`— Sampel data mentah (tidak diindeks) untuk Penyewa-2.

> **Important**
>
> Cerita dalam pola ini mencakup contoh AWS CLI perintah yang diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^).

> **Note**
>
> Dalam AWS CLI perintah, ganti semua nilai dalam kurung sudut (<>) dengan nilai yang benar.

Epik

Buat dan konfigurasikan bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat ember S3.	Buat ember S3 di. Wilayah AWS Bucket ini akan menyimpan data penyewa yang tidak diindeks untuk aplikasi sampel. Pastikan nama bucket S3 unik secara global, karena namespace dibagikan oleh semua orang. Akun AWS
	
	Untuk membuat bucket S3, Anda dapat menggunakan perintah AWS CLI create-bucket sebagai berikut:
	
	`aws s3api create-bucket \`
	`  --bucket <tenantrawdata> \`
	`  --region <your-AWS-Region>`
	
	di `tenantrawdata` mana nama bucket S3. (Anda dapat menggunakan nama unik apa
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pun yang mengikuti pedoman penamaan bucket.)	

Membuat dan mengkonfigurasi cluster Elasticsearch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat domain OpenSearch Layanan.	Jalankan AWS CLI create-elasticsearch-domainperintah untuk membuat domain OpenSearch Layanan: `aws es create-elasticsearch-domain \ --domain-name vpc-cli-example \ --elasticsearch-version 7.10 \ --elasticsearch-cluster-config InstanceType=t3.medium.elasticsearch,InstanceCount=1 \ --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=10 \ --domain-endpoint-options "{\"EnforceHTTPS\": true}" \ --encryption-at-rest-options "{\"Enabled\": true}" \ --node-to-node-encryption-options "{\"Enabled\": true}" \ --advanced-security-options "{\"Enabl`	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>ed\": true, \"Interna
	lUserDatabaseEnabled
	\": true, \
	    \"MasterUserOption
	s\": {\"MasterUserName
	\": \"KibanaUser\", \
	    \"MasterUserPasswo
	rd\": \"NewKiba
	naPassword@123\"}}" \
	  --vpc-options
	 "{\"SubnetIds\":
	 [\"<subnet-id>\"],
	 \"SecurityGroupIds\":
	 [\"<sg-id>\"]}" \
	  --access-policies
	 "{\"Version\":
	 \"2012-10-17\",
	 \"Statement\":
	 [{ \"Effect\":
	 \"Allow\", \
	    \"Principal\":
	 {\"AWS\": \"*\" },
	 \"Action\":\"es:*\",
	 \
	    \"Resource\":
	 \"arn:aws:es:<regi
	on>:<account-id>:d
	omain\/vpc-cli-exa
	mple\/*\" }] }"</pre> Jumlah instans diatur ke 1 karena domain adalah untuk tujuan pengujian. Anda perlu mengaktifkan kontrol akses berbutir halus dengan menggunakan `advanced-security-options` parameter, karena detailnya	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tidak dapat diubah setelah domain dibuat. Perintah ini membuat nama pengguna master (`KibanaUser`) dan kata sandi yang dapat Anda gunakan untuk masuk ke konsol Kibana. Karena domain merupakan bagian dari virtual private cloud (VPC), Anda harus memastikan bahwa Anda dapat menjangkau instance Elasticsearch dengan menentukan kebijakan akses yang akan digunakan. Untuk informasi selengkap nya, lihat Meluncurkan domain OpenSearch Layanan Amazon Anda dalam VPC dalam AWS dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan host benteng.	Siapkan instans Windows Amazon Elastic Compute Cloud (Amazon EC2) sebagai host bastion untuk mengakses konsol Kibana. Grup keamanan Elasticsearch harus mengizinkan lalu lintas dari grup EC2 keamanan Amazon. Untuk petunjuk, lihat posting blog Mengontrol Akses Jaringan ke EC2 Instans Menggunakan Server Bastion.
	
	Ketika host bastion telah disiapkan, dan Anda memiliki grup keamanan yang terkait dengan instance yang tersedia, gunakan AWS CLI authorize-security-group-ingressperintah untuk menambahkan izin ke grup keamanan Elasticsearch untuk mengizinkan port 443 dari grup keamanan Amazon EC2 (bastion host).
	
	`aws ec2 authorize-security-group-ingress \`
	`  --group-id <SecurityGroupIdfElasticSearch> \`
	`  --protocol tcp \`
	`  --port 443 \`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--source-group <SecurityGroupIdfBashionHostEC2>`	

Buat dan konfigurasikan fungsi indeks Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat peran eksekusi Lambda.	Jalankan perintah AWS CLI create-role untuk memberikan akses dan sumber daya fungsi indeks Lambda: Layanan AWS
	`aws iam create-role \ --role-name index-lambda-role \ --assume-role-policy-document file://lambda_assume_role.json`
	di mana `lambda_assume_role.json` adalah dokumen JSON yang memberikan `AssumeRole` izin ke fungsi Lambda, sebagai berikut:
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": {`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Service": "lambda.amazonaws.com"
	},
	"Action": "sts:AssumeRole"
	}
	]
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lampirkan kebijakan terkelola ke peran Lambda.	Jalankan AWS CLI attach-role-policyperintah untuk melampirkan kebijakan terkelola ke peran yang dibuat pada langkah sebelumny a. Kedua kebijakan ini memberikan izin peran untuk membuat elastic network interface dan menulis log ke CloudWatch Log.
	
	`aws iam attach-role-policy \ --role-name index-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole`
	
	`aws iam attach-role-policy \ --role-name index-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat kebijakan untuk memberikan izin fungsi indeks Lambda untuk membaca objek S3.	Jalankan perintah AWS CLI create-policy untuk `s3:GetObject` memberikan izin fungsi indeks Lambda untuk membaca objek di bucket S3:
	
	`aws iam create-policy \ --policy-name s3-permission-policy \ --policy-document file://s3-policy.json`
	
	File tersebut `s3-policy.json` adalah dokumen JSON yang ditunjukkan di bawah ini yang memberikan `s3:GetObject` izin untuk memungkinkan akses baca ke objek S3. Jika Anda menggunakan nama yang berbeda saat membuat bucket S3, berikan nama bucket yang benar di `Resource` bagian ini:
	
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:GetObject",`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Resource
	": "arn:aws:s3:::<ten
	antrawdata>/*"
	}
	]
	}</pre>	

Lampirkan kebijakan izin Amazon S3 ke peran eksekusi Lambda.	Jalankan AWS CLI attach-role-policyperintah untuk melampirkan kebijakan izin Amazon S3 yang Anda buat di langkah sebelumnya ke peran eksekusi Lambda:
	<pre>aws iam attach-role-
	policy \
	--role-name index-lam
	bda-role \
	--policy-arn
	<PolicyARN></pre>
	di `PolicyARN` mana Nama Sumber Daya Amazon (ARN) dari kebijakan izin Amazon S3. Anda bisa mendapatkan nilai ini dari output dari perintah sebelumnya.
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi indeks Lambda.	Jalankan perintah AWS CLI create-function untuk membuat fungsi indeks Lambda, yang akan mengakses Service: OpenSearch `aws lambda create-function \ --function-name index-lambda-function \ --zip-file fileb://index_lambda_package.zip \ --handler lambda_index.lambda_handler \ --runtime python3.9 \ --role "arn:aws:iam::account-id:role/index-lambda-role" \ --timeout 30 \ --vpc-config "{\"SubnetIds\": [\"<subnet-id1\>", \"<subnet-id2>\"], \ \"SecurityGroupIds\": [\"<sg-1>\"]}"`	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Izinkan Amazon S3 memanggil fungsi indeks Lambda.	Jalankan perintah AWS CLI add-permission untuk memberi Amazon S3 izin untuk memanggil fungsi indeks Lambda: `aws lambda add-permission \ --function-name index-lambda-function \ --statement-id s3-permissions \ --action lambda:InvokeFunction \ --principal s3.amazonaws.com \ --source-arn "arn:aws:s3:::<tenantrawdata>" \ --source-account "<account-id>"`	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan pemicu Lambda untuk acara Amazon S3.	Jalankan AWS CLI put-bucket-notification-configuration perintah untuk mengirim notifikasi ke fungsi indeks Lambda saat peristiwa Amazon `ObjectCreated` S3 terdeteksi. Fungsi indeks berjalan setiap kali objek diunggah ke bucket S3. `aws s3api put-bucket-notification-configuration \ --bucket <tenantrawdata> \ --notification-configuration file://s3-trigger.json` File tersebut `s3-trigger.json` adalah dokumen JSON di folder saat ini yang menambahkan kebijakan sumber daya ke fungsi Lambda saat peristiwa Amazon `ObjectCreated` S3 terjadi.	Arsitek cloud, Administrator Cloud

Buat dan konfigurasikan fungsi pencarian Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran eksekusi Lambda.	Jalankan perintah AWS CLI create-role untuk memberikan akses dan sumber daya fungsi	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	pencarian Lambda: Layanan AWS
	
	`aws iam create-role \ --role-name search-lambda-role \ --assume-role-policy-document file://lambda_assume_role.json`
	
	di mana `lambda_assume_role.json` adalah dokumen JSON di folder saat ini yang memberikan `AssumeRole` izin ke fungsi Lambda, sebagai berikut:
	
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lampirkan kebijakan terkelola ke peran Lambda.	Jalankan AWS CLI attach-role-policyperintah untuk melampirkan kebijakan terkelola ke peran yang dibuat pada langkah sebelumny a. Kedua kebijakan ini memberikan izin peran untuk membuat elastic network interface dan menulis log ke CloudWatch Log.
	
	`aws iam attach-role-policy \ --role-name search-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole`
	
	`aws iam attach-role-policy \ --role-name search-lambda-role \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi pencarian Lambda.	Jalankan perintah AWS CLI create-function untuk membuat fungsi pencarian Lambda, yang akan mengakses Service: OpenSearch `aws lambda create-function \ --function-name search-lambda-function \ --zip-file fileb://search_lambda_package.zip \ --handler lambda_search.lambda_handler \ --runtime python3.9 \ --role "arn:aws:iam::account-id:role/search-lambda-role" \ --timeout 30 \ --vpc-config "{\"SubnetIds\": [\"<subnet-id1\>", \"<subnet-id2>\"], \ \"SecurityGroupIds\": [\"<sg-1>\"]}"`	Arsitek cloud, Administrator Cloud

Buat dan konfigurasikan peran penyewa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM penyewa.	Jalankan perintah AWS CLI create-role untuk membuat dua peran penyewa yang	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	akan digunakan untuk menguji fungsionalitas pencarian:
	
	`aws iam create-role \ --role-name Tenant-1-role \ --assume-role-policy-document file://assume-role-policy.json`
	
	`aws iam create-role \ --role-name Tenant-2-role \ --assume-role-policy-document file://assume-role-policy.json`
	
	File tersebut `assume-role-policy.json` adalah dokumen JSON di folder saat ini yang memberikan `AssumeRole` izin untuk peran eksekusi Lambda:
	
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "<Lambda execution role for index function>", "AWS": "<Lambda execution role for search function>"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>},
	"Action":
	"sts:AssumeRole"
	}
	]
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat kebijakan IAM penyewa.	Jalankan perintah AWS CLI create-policy untuk membuat kebijakan penyewa yang memberikan akses ke operasi Elasticsearch:
	`aws iam create-policy \ --policy-name tenant-policy \ --policy-document file://policy.json`
	File tersebut `policy.json` adalah dokumen JSON di folder saat ini yang memberikan izin di Elasticsearch:
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "es:ESHttpDelete", "es:ESHttpGet", "es:ESHttpHead", "es:ESHttpPost", "es:ESHttpPut", "es:ESHttpPatch" ],`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"Resource": ["<ARN of Elasticsearch domain created earlier>"] }] }</pre>	

Lampirkan kebijakan IAM penyewa ke peran penyewa.	Jalankan AWS CLI attach-role-policyperintah untuk melampirkan kebijakan IAM penyewa ke dua peran penyewa yang Anda buat di langkah sebelumnya:
	<pre>aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/tenant-policy \ --role-name Tenant-1-role
	
	aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/tenant-policy \ --role-name Tenant-2-role</pre>
	Kebijakan ARN berasal dari output dari langkah sebelumnya.
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat kebijakan IAM untuk memberikan izin Lambda untuk mengambil peran.	Jalankan perintah AWS CLI create-policy untuk membuat kebijakan bagi Lambda untuk mengambil peran penyewa:
	`aws iam create-policy \ --policy-name assume-tenant-role-policy \ --policy-document file://lambda_policy.json`
	File tersebut `lambda_policy.json` adalah dokumen JSON di folder saat ini yang memberikan izin untuk: `AssumeRole`
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "<ARN of tenant role created earlier>" } ] }`
	Untuk`Resource`, Anda dapat menggunakan karakter wildcard untuk menghindari
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pembuatan kebijakan baru untuk setiap penyewa.	

Buat kebijakan IAM untuk memberikan izin peran indeks Lambda untuk mengakses Amazon S3.	Jalankan perintah AWS CLI create-policy untuk memberikan izin peran indeks Lambda untuk mengakses objek di bucket S3:
	`aws iam create-policy \ --policy-name s3-permission-policy \ --policy-document file://s3_lambda_policy.json`
	File tersebut `s3_lambda_policy.json` adalah dokumen kebijakan JSON berikut di folder saat ini:
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::tenantrawdata/*" } ] }`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lampirkan kebijakan ke peran eksekusi Lambda.	Jalankan AWS CLI attach-role-policyperintah untuk melampirkan kebijakan yang dibuat pada langkah sebelumnya ke indeks Lambda dan peran eksekusi pencarian yang Anda buat sebelumnya:
	
	`aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/assume-tenant-role-policy \ --role-name index-lambda-role`
	
	`aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/assume-tenant-role-policy \ --role-name search-lambda-role`
	
	`aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/s3-permission-policy \ --role-name index-lambda-role`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Kebijakan ARN berasal dari output dari langkah sebelumnya.	

Membuat dan mengonfigurasi API penelusuran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat REST API di API Gateway.	Jalankan AWS CLI create-rest-api perintah untuk membuat sumber daya REST API: `aws apigateway create-rest-api \ --name Test-Api \ --endpoint-configuration "{ \"types\": [\"REGIONAL\"] }"` Untuk jenis konfigurasi titik akhir, Anda dapat menentukan `EDGE` alih-alih `REGIONAL` menggunakan lokasi tepi, bukan lokasi tertentu Wilayah AWS. Perhatikan nilai `id` bidang dari output perintah. Ini adalah ID API yang akan Anda gunakan dalam perintah berikutnya.	Arsitek cloud, Administrator Cloud
Buat sumber daya untuk API pencarian.	Sumber daya API pencarian memulai fungsi pencarian Lambda dengan nama sumber daya. `search` (Anda tidak perlu membuat API untuk	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	fungsi indeks Lambda, karena ini berjalan secara otomatis saat objek diunggah ke bucket S3.)
	
	1. Jalankan perintah AWS CLI get-resources untuk mendapatkan ID induk untuk jalur root:
	
	`aws apigateway get-resources \ --rest-api-id <API-ID>`
	
	Perhatikan nilai bidang ID. Anda akan menggunakan ID induk ini di perintah berikutnya.
	
	`{ "items": [ { "id": "zpsri964ck", "path": "/" } ] }`
	
	2. Jalankan perintah AWS CLI create-resource untuk membuat resource untuk API pencarian. Untuk `parent-id`, tentukan ID dari perintah sebelumnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws apigateway create-resource \ --rest-api-id <API-ID> \ --parent-id <Parent-ID> \ --path-part search`	
Buat metode GET untuk API pencarian.	Jalankan perintah AWS CLI put-method untuk membuat `GET` metode untuk API pencarian: `aws apigateway put-method \ --rest-api-id <API-ID> \ --resource-id <ID from the previous command output> \ --http-method GET \ --authorization-type "NONE" \ --no-api-key-required` Untuk `resource-id` , tentukan ID dari output `create-resource` perintah.	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat respons metode untuk API penelusuran.	Jalankan AWS CLI put-method-responseperintah untuk menambahkan respons metode untuk API pencarian: `aws apigateway put-method-response \ --rest-api-id <API-ID> \ --resource-id <ID from the create-resource command output> \ --http-method GET \ --status-code 200 \ --response-models "{\"application/json\": \"Empty\"}"` Untuk`resource-id` , tentukan ID dari output dari `create-resource` perintah sebelumnya.	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan integrasi Lambda proxy untuk API pencarian.	Jalankan perintah AWS CLI put-integration untuk mengatur integrasi dengan fungsi pencarian Lambda: `aws apigateway put-integration \ --rest-api-id <API-ID> \ --resource-id <ID from the create-resource command output> \ --http-method GET \ --type AWS_PROXY \ --integration-http-method GET \ --uri arn:aws:apigateway:region:lambda:path/2015-03-31/functions/arn:aws:lambda:<region>:<account-id>:function:<function-name>/invocations` Untuk`resource-id`, tentukan ID dari `create-resource` perintah sebelumnya.	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan izin API Gateway untuk memanggil fungsi pencarian Lambda.	Jalankan perintah AWS CLI add-permission untuk memberikan izin API Gateway untuk menggunakan fungsi pencarian: `aws lambda add-permission \ --function-name <function-name> \ --statement-id apigateway-get \ --action lambda:InvokeFunction \ --principal apigateway.amazonaws.com \ --source-arn "arn:aws:execute-api:<region>:<account-id>:api-id/*/GET/search"` Ubah `source-arn` jalur jika Anda menggunakan nama sumber daya API yang berbeda, bukan`search`.	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan API pencarian.	Jalankan perintah AWS CLI create-deployment untuk membuat sumber daya panggung bernama: `dev`
	
	`aws apigateway create-deployment \`
	`  --rest-api-id <API-ID> \`
	`  --stage-name dev`
	
	Jika Anda memperbarui API, Anda dapat menggunakan AWS CLI perintah yang sama untuk menerapkannya kembali ke tahap yang sama.
		Arsitek cloud, Administrator Cloud

Buat dan konfigurasikan peran Kibana

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke konsol Kibana.	1. Temukan tautan ke Kibana di dasbor domain Anda di konsol OpenSearch Layanan. URL dalam bentuk:`<domain-endpoint>/_plugin/kibana/` . 2. Gunakan host bastion yang Anda konfigurasikan di epik pertama untuk mengakses konsol Kibana. 3. Masuk ke konsol Kibana dengan menggunakan	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nama pengguna dan kata sandi master dari langkah sebelumnya, saat Anda membuat domain OpenSearch Layanan. 4. Ketika diminta untuk memilih penyewa, pilih Pribadi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan konfigurasikan peran Kibana.	Untuk memberikan isolasi data dan memastikan bahwa satu penyewa tidak dapat mengambil data penyewa lain, Anda perlu menggunak an keamanan dokumen, yang memungkinkan penyewa untuk mengakses hanya dokumen yang berisi ID penyewa mereka. 1. Di konsol Kibana, di panel navigasi, pilih Keamanan, Peran. 2. Buat peran penyewa baru. 3. Setel izin cluster `keindices_all` , yang memberikan izin buat, baca, perbarui, dan hapus (CRUD) pada indeks Layanan. OpenSearch 4. Batasi izin indeks ke indeks. `tenant-data` (Nama indeks harus cocok dengan nama dalam fungsi pencarian dan indeks Lambda.) 5. Tetapkan izin indeks `keindices_all` , untuk memungkinkan pengguna melakukan semua operasi terkait indeks. (Anda dapat membatasi operasi untuk akses yang lebih terperinci,	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tergantung pada kebutuhan Anda.) 6. Untuk keamanan tingkat dokumen, gunakan kebijakan berikut untuk memfilter dokumen berdasarkan ID penyewa, untuk menyediakan isolasi data bagi penyewa dalam indeks bersama: `{ "bool": { "must": { "match": { "TenantId": "Tenant-1" } } } }` Nama indeks, properti, dan nilai peka huruf besar/kecil.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memetakan pengguna ke peran.	1. Pilih tab Pengguna yang dipetakan untuk peran tersebut, lalu pilih Pengguna peta. 2. Di bagian peran Backend, tentukan ARN dari peran penyewa IAM yang Anda buat sebelumnya, lalu pilih Peta. Ini memetakan peran penyewa IAM ke peran Kibana sehingga pencarian khusus penyewa mengembalikan data untuk penyewa itu saja. Misalnya, jika nama peran IAM untuk Penyewa-1 adalah`Tenant-1-Role` , tentukan ARN untuk `Tenant-1-Role` (dari epik Buat dan konfigura sikan peran penyewa) di kotak peran Backend untuk peran Penyewa-1 Kibana. 3. Ulangi langkah 1 dan 2 untuk Penyewa-2. Kami menyarankan Anda mengotomatiskan pembuatan peran penyewa dan Kibana pada saat orientasi penyewa.	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat indeks data penyewa.	Di panel navigasi, di bawah Manajemen, pilih Alat Pengembang, lalu jalankan perintah berikut. Perintah ini membuat `tenant-data` indeks untuk menentukan pemetaan untuk `TenantId` properti.
	`PUT /tenant-data`
	`{`
	`  "mappings": {`
	`  "properties": {`
	`  "TenantId":`
	`  { "type": "keyword"}`
	`  }`
	`  }`
	`}`
		Arsitek cloud, Administrator Cloud

Buat titik akhir VPC untuk Amazon S3 dan AWS STS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat titik akhir VPC untuk Amazon S3.	Jalankan AWS CLI create-vpc-endpointperintah untuk membuat titik akhir VPC untuk Amazon S3. Titik akhir memungkinkan fungsi indeks Lambda di VPC untuk mengakses Amazon S3.
	`aws ec2 create-vpc-endpoint \`
	`  --vpc-id <VPC-ID> \`
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--service-name com.amazonaws.us-east-1.s3 \
	--route-table-ids <route-table-ID></pre>
	Untuk`vpc-id`, tentukan VPC yang Anda gunakan untuk fungsi indeks Lambda. Untuk`service-name` , gunakan URL yang benar untuk titik akhir Amazon S3. Untuk`route-table-ids` , tentukan tabel rute yang terkait dengan titik akhir VPC.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat titik akhir VPC untuk. AWS STS	Jalankan AWS CLI create-vpc-endpointperintah untuk membuat titik akhir VPC untuk AWS Security Token Service ().AWS STS Titik akhir memungkinkan indeks Lambda dan fungsi pencarian di VPC untuk mengakses. AWS STS Fungsi digunakan AWS STS ketika mereka mengambil peran IAM. `aws ec2 create-vpc-endpoint \ --vpc-id <VPC-ID> \ --vpc-endpoint-type Interface \ --service-name com.amazonaws.us-east-1.sts \ --subnet-id <subnet-ID> \ --security-group-id <security-group-ID>` Untuk`vpc-id`, tentukan VPC yang Anda gunakan untuk indeks Lambda dan fungsi pencarian. Untuk`subnet-id` , berikan subnet di mana titik akhir ini harus dibuat. Untuk`security-group-id`, tentukan grup keamanan untuk mengaitkan titik akhir ini dengan. (Ini bisa sama	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan kelompok keamanan yang digunakan Lambda.)	

Uji multi-tenancy dan isolasi data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file Python untuk fungsi indeks dan pencarian.	1. Dalam `index_lambda_package.zip` file, edit `lamba_index.py` file untuk memperbarui Akun AWS ID, Wilayah AWS, dan informasi titik akhir Elasticsearch. 2. Dalam `search_lambda_package.zip` file, edit `lambda_search.py` file untuk memperbarui Akun AWS ID, Wilayah AWS, dan informasi titik akhir Elasticsearch. Anda bisa mendapatkan titik akhir Elasticsearch dari tab Ikhtisar konsol Layanan. OpenSearch Ini memiliki format`<AWS-Region>.es.amazonaws.com` .	Arsitek cloud, Pengembang aplikasi
Perbarui kode Lambda.	Gunakan AWS CLI update-function-codeperintah untuk memperbarui kode Lambda	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dengan perubahan yang Anda buat pada file Python:
	
	`aws lambda update-function-code \`
	`  --function-name index-lambda-function \`
	`  --zip-file fileb://index_lambda_package.zip`
	
	`aws lambda update-function-code \`
	`  --function-name search-lambda-function \`
	`  --zip-file fileb://search_lambda_package.zip`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Unggah data mentah ke bucket S3.	Gunakan perintah AWS CLI cp untuk mengunggah data objek Tenant-1 dan Tenant-2 ke `tenantrawdata` bucket (tentukan nama bucket S3 yang Anda buat untuk tujuan ini):
	`aws s3 cp tenant-1-data s3://tenantrawdata`
	`aws s3 cp tenant-2-data s3://tenantrawdata`
	Bucket S3 diatur untuk menjalankan fungsi indeks Lambda setiap kali data diunggah sehingga dokumen diindeks di Elasticsearch.
		Arsitek cloud, Administrator Cloud

Cari data dari konsol Kibana.	Di konsol Kibana, jalankan kueri berikut: `GET tenant-data/_search` Kueri ini menampilkan semua dokumen yang diindeks di Elasticsearch. Dalam hal ini, Anda akan melihat dua dokumen terpisah untuk Tenant-1 dan Tenant-2.	Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Uji API pencarian dari API Gateway.	1. Di konsol API Gateway, buka API pencarian, pilih `GET` metode di dalam sumber penelusuran, lalu pilih Uji.
	2. Di jendela pengujian, berikan string kueri berikut (peka huruf besar/kecil) untuk ID penyewa, lalu pilih Uji.
	`TenantId=Tenant-1`
	Fungsi Lambda mengirimkan kueri ke OpenSearch Layanan yang memfilter dokumen penyewa berdasarkan keamanan tingkat dokumen. Metode mengembalikan dokumen milik Tenant-1.
	3. Ubah string kueri menjadi:
	`TenantId=Tenant-2`
	Query ini mengembalikan dokumen milik Tenant-2.
	
	Untuk ilustrasi layar, lihat bagian Informasi tambahan.
		Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pembersihan sumber daya	Bersihkan semua sumber daya yang Anda buat untuk mencegah biaya tambahan ke akun Anda.	AWS DevOps, arsitek Cloud, Administrator Cloud

Sumber daya terkait

- AWS SDK untuk Python (Boto)
- AWS Lambda dokumentasi
- Dokumentasi API Gateway
- Dokumentasi Amazon S3
- Dokumentasi OpenSearch Layanan Amazon
  - Kontrol akses berbutir halus di Layanan Amazon OpenSearch
  - Membuat aplikasi pencarian dengan Amazon OpenSearch Service
  - Meluncurkan domain OpenSearch Layanan Amazon Anda dalam VPC

Informasi tambahan

Model partisi data

Ada tiga model partisi data umum yang digunakan dalam sistem multi-tenant: silo, pool, dan hybrid. Model yang Anda pilih tergantung pada kepatuhan, tetangga yang bising, operasi, dan kebutuhan isolasi lingkungan Anda.

Model silo

Dalam model silo, setiap data penyewa disimpan di area penyimpanan yang berbeda di mana tidak ada percampuran data penyewa. Anda dapat menggunakan dua pendekatan untuk mengimplementasikan model silo dengan OpenSearch Layanan: domain per penyewa dan indeks per penyewa.

- Domain per penyewa — Anda dapat menggunakan domain OpenSearch Layanan terpisah (identik dengan cluster Elasticsearch) per penyewa. Menempatkan setiap penyewa di domainnya sendiri memberikan semua manfaat yang terkait dengan memiliki data dalam konstruksi mandiri.

Namun, pendekatan ini memperkenalkan tantangan manajemen dan kelincahan. Sifatnya yang terdistribusi membuat lebih sulit untuk mengumpulkan dan menilai kesehatan operasional dan aktivitas penyewa. Ini adalah opsi mahal yang mengharuskan setiap domain OpenSearch Layanan memiliki tiga node master dan dua node data untuk beban kerja produksi seminimal mungkin.

- Indeks per penyewa — Anda dapat menempatkan data penyewa dalam indeks terpisah dalam kluster Layanan. OpenSearch Dengan pendekatan ini, Anda menggunakan pengenal penyewa saat membuat dan memberi nama indeks, dengan mengawali pengenal penyewa ke nama indeks. Pendekatan indeks per penyewa membantu Anda mencapai tujuan silo Anda tanpa memperkenalkan cluster yang benar-benar terpisah untuk setiap penyewa. Namun, Anda mungkin mengalami tekanan memori jika jumlah indeks bertambah, karena pendekatan ini membutuhkan lebih banyak pecahan, dan node master harus menangani lebih banyak alokasi dan penyeimbangan kembali.

Isolasi dalam model silo — Dalam model silo, Anda menggunakan kebijakan IAM untuk mengisolasi domain atau indeks yang menyimpan data setiap penyewa. Kebijakan ini mencegah satu penyewa mengakses data penyewa lain. Untuk menerapkan model isolasi silo, Anda dapat membuat kebijakan berbasis sumber daya yang mengontrol akses ke sumber daya penyewa Anda. Ini sering merupakan kebijakan akses domain yang menentukan tindakan mana yang dapat dilakukan prinsipal pada sub-sumber daya domain, termasuk indeks Elasticsearch dan. APIs Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan yang diizinkan atau ditolak pada domain, indeks, atau dalam Layanan. APIs OpenSearch `Action`Elemen kebijakan IAM menjelaskan tindakan atau tindakan spesifik yang diizinkan atau ditolak oleh kebijakan, dan `Principal` elemen tersebut menentukan akun, pengguna, atau peran yang terpengaruh.

Kebijakan contoh berikut memberikan akses penuh Penyewa-1 (seperti yang ditentukan oleh`es:*`) ke sub-sumber daya pada domain saja. `tenant-1` Bagian tambahan `/*` dalam `Resource` elemen menunjukkan bahwa kebijakan ini berlaku untuk sub-sumber daya domain, bukan untuk domain itu sendiri. Ketika kebijakan ini berlaku, penyewa tidak diizinkan untuk membuat domain baru atau mengubah setelan pada domain yang sudah ada.

```
{
```

```
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::<aws-account-id>:user/Tenant-1"
         },
         "Action": "es:*",
         "Resource": "arn:aws:es:<Region>:<account-id>:domain/tenant-1/*"
      }
   ]
}
```

Untuk mengimplementasikan model silo penyewa per indeks, Anda perlu memodifikasi kebijakan sampel ini untuk lebih membatasi Penyewa-1 ke indeks atau indeks yang ditentukan, dengan menentukan nama indeks. Kebijakan sampel berikut membatasi Penyewa-1 ke indeks. `tenant-index-1`

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Tenant-1"
         },
         "Action": "es:*",
         "Resource": "arn:aws:es:<Region>:<account-id>:domain/test-domain/tenant-
index-1/*"
      }
   ]
}
```

Model kolam

Dalam model pool, semua data penyewa disimpan dalam indeks dalam domain yang sama. Pengenal penyewa disertakan dalam data (dokumen) dan digunakan sebagai kunci partisi, sehingga Anda dapat menentukan data mana yang menjadi milik penyewa mana. Model ini mengurangi overhead manajemen. Mengoperasikan dan mengelola indeks gabungan lebih mudah dan lebih efisien daripada mengelola beberapa indeks. Namun, karena data penyewa bercampur dalam indeks

yang sama, Anda kehilangan isolasi penyewa alami yang disediakan oleh model silo. Pendekatan ini mungkin juga menurunkan kinerja karena efek tetangga yang bising.

Isolasi penyewa dalam model kolam — Secara umum, isolasi penyewa menantang untuk diterapkan dalam model kolam renang. Mekanisme IAM yang digunakan dengan model silo tidak memungkinkan Anda untuk menggambarkan isolasi berdasarkan ID penyewa yang disimpan dalam dokumen Anda.

Pendekatan alternatif adalah dengan menggunakan dukungan kontrol akses halus (FGAC) yang disediakan oleh Open Distro untuk Elasticsearch. FGAC memungkinkan Anda mengontrol izin pada tingkat indeks, dokumen, atau bidang. Dengan setiap permintaan, FGAC mengevaluasi kredensi pengguna dan mengautentikasi pengguna atau menolak akses. Jika FGAC mengautentikasi pengguna, FGAC mengambil semua peran yang dipetakan ke pengguna tersebut dan menggunakan set lengkap izin untuk menentukan cara menangani permintaan.

Untuk mencapai isolasi yang diperlukan dalam model gabungan, Anda dapat menggunakan keamanan tingkat dokumen, yang memungkinkan Anda membatasi peran pada subset dokumen dalam indeks. Peran contoh berikut membatasi kueri ke Penyewa-1. Dengan menerapkan peran ini ke Penyewa-1, Anda dapat mencapai isolasi yang diperlukan.

```
{
   "bool": {
     "must": {
       "match": {
         "tenantId": "Tenant-1"
       }
     }
   }
 }
```

Model hibrida

Model hibrida menggunakan kombinasi model silo dan kolam renang di lingkungan yang sama untuk menawarkan pengalaman unik untuk setiap tingkat penyewa (seperti tingkatan gratis, standar, dan premium). Setiap tingkatan mengikuti profil keamanan yang sama yang digunakan dalam model pool.

Isolasi penyewa dalam model hybrid — Dalam model hybrid, Anda mengikuti profil keamanan yang sama seperti pada model pool, di mana menggunakan model keamanan FGAC pada tingkat

dokumen menyediakan isolasi penyewa. Meskipun strategi ini menyederhanakan manajemen cluster dan menawarkan kelincahan, ini memperumit aspek lain dari arsitektur. Misalnya, kode Anda memerlukan kompleksitas tambahan untuk menentukan model mana yang terkait dengan setiap penyewa. Anda juga harus memastikan bahwa kueri penyewa tunggal tidak memenuhi seluruh domain dan menurunkan pengalaman untuk penyewa lain.

Pengujian di API Gateway

Jendela uji untuk kueri Penyewa-1

Jendela uji untuk kueri Penyewa-2

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menerapkan aplikasi multi-stack menggunakan AWS CDK dengan TypeScript

Dr. Rahul Sharad Gaikwad, Amazon Web Services

Ringkasan

Pola ini menyediakan step-by-step pendekatan untuk penerapan aplikasi di Amazon Web Services (AWS) menggunakan AWS Cloud Development Kit (AWS CDK) dengan. TypeScript Sebagai contoh, pola tersebut menyebarkan aplikasi analitik real-time tanpa server.

Pola membangun dan menyebarkan aplikasi tumpukan bersarang. CloudFormation Tumpukan AWS induk memanggil tumpukan turunan, atau bersarang,. Setiap tumpukan anak membangun dan menerapkan sumber daya AWS yang ditentukan dalam tumpukan. CloudFormation AWS CDK Toolkit, perintah antarmuka baris perintah (CLI)`cdk`, adalah antarmuka utama untuk tumpukan. CloudFormation

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Virtual Private Cloud (VPC) dan subnet yang ada
- AWS CDK Toolkit diinstal dan dikonfigurasi
- Seorang pengguna dengan izin administrator dan satu set kunci akses.
- Node.js
- Antarmuka Baris Perintah AWS (AWS CLI)

Batasan

- Karena AWS CDK menggunakan AWS CloudFormation, aplikasi AWS CDK tunduk pada kuota CloudFormation layanan. Untuk informasi selengkapnya, lihat CloudFormation kuota AWS.

Versi produk

Pola ini telah dibangun dan diuji menggunakan alat dan versi berikut.

- Perangkat AWS CDK 1.83.0

- Node.js 14.13.0
- npm 7.0.14

Pola tersebut harus berfungsi dengan versi AWS CDK atau npm apa pun. Perhatikan bahwa Node.js versi 13.0.0 hingga 13.6.0 tidak kompatibel dengan AWS CDK.

Arsitektur

Tumpukan teknologi target

- Konsol AWS Amplify
- Amazon API Gateway
- AWS CDK
- Amazon CloudFront
- Amazon Cognito
- Amazon DynamoDB
- Amazon Data Firehose
- Amazon Kinesis Data Streams
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)

Arsitektur target

Diagram berikut menunjukkan penerapan aplikasi multi-tumpukan menggunakan AWS CDK dengan. TypeScript

Diagram berikut menunjukkan arsitektur contoh aplikasi real-time tanpa server.

Alat

Alat

- AWS Amplify Console adalah pusat kendali untuk penerapan web dan aplikasi seluler fullstack di AWS. Hosting Konsol Amplify menyediakan alur kerja berbasis git untuk hosting aplikasi

web nirserver fullstack dengan deployment kontinu. UI Admin adalah antarmuka visual untuk pengembang web dan seluler frontend untuk membuat dan mengelola backend aplikasi di luar konsol AWS.

- Amazon API Gateway adalah layanan AWS untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs pada skala apa pun.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS CDK Toolkit adalah kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan aplikasi AWS CDK Anda. Perintah `cdk` CLI adalah alat utama untuk berinteraksi dengan aplikasi AWS CDK Anda. Ini menjalankan aplikasi Anda, menginterogasi model aplikasi yang Anda tentukan, dan memproduksi serta menerapkan CloudFormation template AWS yang dihasilkan oleh AWS CDK.
- Amazon CloudFront adalah layanan web yang mempercepat distribusi konten web statis dan dinamis, seperti.html, .css, .js, dan file gambar. CloudFront mengirimkan konten Anda melalui jaringan pusat data di seluruh dunia yang disebut lokasi tepi untuk latensi yang lebih rendah dan peningkatan kinerja.
- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk web dan aplikasi seluler Anda. Pengguna Anda dapat masuk secara langsung atau melalui pihak ketiga.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi dengan skalabilitas yang mulus.
- Amazon Data Firehose adalah layanan yang dikelola sepenuhnya untuk mengirimkan data streaming real-time ke tujuan seperti Amazon S3, Amazon Redshift, OpenSearch Amazon Service, Splunk, dan titik akhir HTTP kustom atau titik akhir HTTP apa pun yang dimiliki oleh penyedia layanan pihak ketiga yang didukung.
- Amazon Kinesis Data Streams adalah layanan untuk mengumpulkan dan memproses aliran besar catatan data secara real time.
- AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Kode

Kode untuk pola ini terlampir.

Epik

Instal AWS CDK Toolkit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS CDK Toolkit.	Untuk menginstal AWS CDK Toolkit secara global, jalankan perintah berikut. `npm install -g aws-cdk`	DevOps
Verifikasi versinya.	Untuk memverifikasi versi AWS CDK Toolkit, jalankan perintah berikut. `cdk --version`	DevOps

Mengatur kredensi AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan kredensil.	Untuk mengatur kredensil, jalankan `aws configure` perintah dan ikuti petunjuknya.
	`$aws configure`
	`AWS Access Key ID`
	`[None]:`
	`AWS Secret Access Key`
	`[None]: your_secr`
	`et_access_key`
	`Default region name`
	`[None]:`
		DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Default output format [None]:`	

Unduh kode proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kode proyek terlampir.	Untuk informasi selengkapnya tentang direktori dan struktur file, lihat bagian Informasi tambahan.	DevOps

Bootstrap lingkungan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap lingkungan.	Untuk menerapkan CloudFormation template AWS ke akun dan Wilayah AWS yang ingin Anda gunakan, jalankan perintah berikut. `cdk bootstrap <account>/<Region>` Untuk informasi selengkapnya, lihat dokumentasi AWS.	DevOps

Membangun dan menyebarkan proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bangun proyek.	Untuk membangun kode proyek, jalankan `npm run build` perintah.	DevOps
Deploy proyek.	Untuk menyebarkan kode proyek, jalankan `cdk deploy` perintah.	

Verifikasi output

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi pembuatan tumpukan.	Di AWS Management Console, pilih CloudFormation. Di tumpukan untuk proyek, verifikasi bahwa tumpukan induk dan dua tumpukan anak telah dibuat.	DevOps

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim data ke Kinesis Data Streams.	Konfigurasikan Akun AWS Anda untuk mengirim data ke Kinesis Data Streams menggunakan Amazon Kinesis Data Generator (KDG). Untuk informasi selengkapnya, lihat Amazon Kinesis Data Generator.	DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna Amazon Cognito.	Untuk membuat pengguna Amazon Cognito, unduh templat cognito-setup.json CloudFormation dari bagian Buat Pengguna Amazon Cognito di halaman bantuan Kinesis Data Generator. Mulai template, lalu masukkan Nama Pengguna dan Kata Sandi Amazon Cognito Anda. Tab Output mencantumkan URL Kinesis Data Generator.	DevOps
Masuk ke Kinesis Data Generator	Untuk masuk ke KDG, gunakan kredenal Amazon Cognito yang Anda berikan dan URL Kinesis Data Generator.	DevOps
Uji aplikasi.	Di KDG, di Rekam template, Template 1, tempel kode uji dari bagian Informasi tambahan, dan pilih Kirim data.	DevOps
Uji API Gateway.	Setelah data tertelan, uji API Gateway dengan menggunakan `GET` metode untuk mengambil data.	DevOps

Sumber daya terkait

Referensi

- AWS Cloud Development Kit

- AWS CDK aktif GitHub
- Bekerja dengan tumpukan bersarang
- Contoh contoh AWS - Analisis real-time tanpa server

Informasi tambahan

Direktori dan detail file

Pola ini mengatur tiga tumpukan berikut.

- `parent-cdk-stack.ts`— Tumpukan ini bertindak sebagai tumpukan induk dan memanggil dua aplikasi anak sebagai tumpukan bersarang.
- `real-time-analytics-poc-stack.ts`— Tumpukan bersarang ini berisi infrastruktur dan kode aplikasi.
- `real-time-analytics-web-stack.ts`— Tumpukan bersarang ini hanya berisi kode aplikasi web statis.

File penting dan fungsinya

- `bin/real-time-analytics-poc.ts`— Titik masuk aplikasi AWS CDK. Ini memuat semua tumpukan yang didefinisikan di bawah`lib/`.
- `lib/real-time-analytics-poc-stack.ts`— Definisi stack (`real-time-analytics-poc`) aplikasi AWS CDK.
- `lib/real-time-analytics-web-stack.ts`— Definisi stack (`real-time-analytics-web-stack`) aplikasi AWS CDK.
- `lib/parent-cdk-stack.ts`— Definisi stack (`parent-cdk`) aplikasi AWS CDK.
- `package.json`— manifes modul npm, yang mencakup nama aplikasi, versi, dan dependensi.
- `package-lock.json`— Dipelihara oleh npm.
- `cdk.json`— Toolkit untuk menjalankan aplikasi.
- `tsconfig.json`- TypeScript Konfigurasi proyek.
- `.gitignore`— Daftar file yang harus dikecualikan Git dari kontrol sumber.
- `node_modules`— Dipelihara oleh npm; termasuk dependensi proyek.

Bagian kode berikut di tumpukan induk memanggil aplikasi turunan sebagai tumpukan AWS CDK bersarang.

```
import * as cdk from '@aws-cdk/core';
import { Construct, Stack, StackProps } from '@aws-cdk/core';
import { RealTimeAnalyticsPocStack } from './real-time-analytics-poc-stack';
import { RealTimeAnalyticsWebStack } from './real-time-analytics-web-stack';


export class CdkParentStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);


    new RealTimeAnalyticsPocStack(this, 'RealTimeAnalyticsPocStack');
    new RealTimeAnalyticsWebStack(this, 'RealTimeAnalyticsWebStack');
  }
}
```

Kode untuk pengujian

```
session={{date.now('YYYYMMDD')}}|sequence={{date.now('x')}}|
reception={{date.now('x')}}|instrument={{random.number(9)}}|
l={{random.number(20)}}|price_0={{random.number({"min":10000,
 "max":30000})}}|price_1={{random.number({"min":10000, "max":30000})}}|
price_2={{random.number({"min":10000, "max":30000})}}|
price_3={{random.number({"min":10000, "max":30000})}}|
price_4={{random.number({"min":10000, "max":30000})}}|
price_5={{random.number({"min":10000, "max":30000})}}|
price_6={{random.number({"min":10000, "max":30000})}}|
price_7={{random.number({"min":10000, "max":30000})}}|
price_8={{random.number({"min":10000, "max":30000})}}|
```

Menguji API Gateway

Di konsol API Gateway, uji API Gateway dengan menggunakan `GET` metode ini.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Otomatiskan penerapan aplikasi bersarang menggunakan AWS SAM

Dr. Rahul Sharad Gaikwad, Ishwar Chathaiwale, Dmitry Gulin, dan Tabby Ward, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS), AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka yang menyediakan sintaks singkatan untuk mengekspresikan APIs fungsi, database, dan pemetaan sumber peristiwa. Dengan hanya beberapa baris untuk setiap sumber daya, Anda dapat menentukan aplikasi yang Anda inginkan dan memodelkannya dengan menggunakan YAMAL. Selama penerapan, SAM mengubah dan memperluas sintaks SAM menjadi sintaks CloudFormation AWS yang dapat Anda gunakan untuk membangun aplikasi tanpa server lebih cepat.

AWS SAM menyederhanakan pengembangan, penerapan, dan pengelolaan aplikasi tanpa server di platform AWS. Ini menyediakan kerangka kerja standar, penyebaran yang lebih cepat, kemampuan pengujian lokal, manajemen sumber daya, Integrasi tanpa batas dengan Alat Pengembangan, dan komunitas yang mendukung. Fitur-fitur ini menjadikannya alat yang berharga untuk membangun aplikasi tanpa server secara efisien dan efektif.

Pola ini menggunakan template AWS SAM untuk mengotomatiskan penerapan aplikasi bersarang. Aplikasi bersarang adalah aplikasi dalam aplikasi lain. Aplikasi induk memanggil aplikasi anak mereka. Ini adalah komponen yang digabungkan secara longgar dari arsitektur tanpa server.

Menggunakan aplikasi bersarang, Anda dapat dengan cepat membangun arsitektur tanpa server yang sangat canggih dengan menggunakan kembali layanan atau komponen yang ditulis dan dipelihara secara independen tetapi disusun menggunakan AWS SAM dan Serverless Application Repository. Aplikasi bersarang membantu Anda membangun aplikasi yang lebih kuat, menghindari pekerjaan duplikat, dan memastikan konsistensi dan praktik terbaik di seluruh tim dan organisasi Anda. Untuk mendemonstrasikan aplikasi bersarang, pola tersebut menerapkan contoh aplikasi keranjang belanja tanpa server AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Virtual Private Cloud (VPC) dan subnet

- Lingkungan pengembangan terintegrasi seperti Visual Studio Code (untuk informasi selengkapnya, lihat Alat untuk Dibangun di AWS)
- Pustaka roda Python diinstal menggunakan pip install wheel, jika belum diinstal

Batasan

- Jumlah maksimum aplikasi yang dapat di-nest dalam aplikasi nirserver adalah 200.
- Jumlah maksimum parameter untuk aplikasi bersarang dapat memiliki 60.

Versi produk

- Solusi ini dibangun di atas antarmuka baris perintah AWS SAM (AWS SAM CLI) versi 1.21.1, tetapi arsitektur ini harus bekerja dengan versi AWS SAM CLI yang lebih baru.

Arsitektur

Tumpukan teknologi target

- Amazon API Gateway
- AWS SAM
- Amazon Cognito
- Amazon DynamoDB
- AWS Lambda
- Antrean Amazon Simple Queue Service (Amazon SQS)

Arsitektur target

Diagram berikut menunjukkan bagaimana permintaan pengguna dibuat ke layanan belanja dengan menelepon APIs. Permintaan pengguna, termasuk semua informasi yang diperlukan, dikirim ke Amazon API Gateway dan otorisasi Amazon Cognito, yang melakukan mekanisme otentikasi dan otorisasi untuk. APIs

Ketika item ditambahkan, dihapus, atau diperbarui di DynamoDB, peristiwa dimasukkan ke DynamoDB Streams, yang pada gilirannya memulai fungsi Lambda. Untuk menghindari penghapusan langsung item lama sebagai bagian dari alur kerja sinkron, pesan dimasukkan ke antrian SQS, yang memulai fungsi pekerja untuk menghapus pesan.

Dalam pengaturan solusi ini, AWS SAM CLI berfungsi sebagai antarmuka untuk tumpukan CloudFormation AWS. Template AWS SAM secara otomatis menyebarkan aplikasi bersarang. Template SAM induk memanggil template anak, dan CloudFormation tumpukan induk menyebarkan tumpukan anak. Setiap tumpukan anak membangun sumber daya AWS yang ditentukan dalam CloudFormation templat AWS SAM.

1. Bangun dan gunakan tumpukan.
2. CloudFormation Tumpukan Auth berisi Amazon Cognito.
3. CloudFormation Tumpukan Produk berisi fungsi Lambda dan Amazon API Gateway
4. CloudFormation Tumpukan Belanja berisi fungsi Lambda, Amazon API Gateway, antrean SQS, dan database Amazon DynamoDB.

Alat

Alat

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka yang membantu Anda membangun aplikasi tanpa server di AWS Cloud.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.

Kode

Kode untuk pola ini tersedia di repositori GitHub AWS SAM Nested Stack Sample.

Epik

Instal AWS SAM CLI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal AWS SAM CLI.	Untuk menginstal AWS SAM CLI, lihat instruksi dalam dokumentasi AWS SAM.	DevOps insinyur

Siapkan kredensil AWS.	Untuk menetapkan kredensi AWS sehingga AWS SAM CLI dapat melakukan panggilan ke layanan AWS atas nama Anda, jalankan `aws configure` perintah dan ikuti petunjuknya.
	`$aws configure`
	`AWS Access Key ID`
	`[None]: <your_acc`
	`ess_key_id>`
	`AWS Secret Access Key`
	`[None]: your_secr`
	`et_access_key`
	`Default region name`
	`[None]:`
	`Default output format`
	`[None]:`
	Untuk informasi selengkapnya tentang menyiapkan kredensil Anda, lihat Otentikasi dan akses kredensil.
		DevOps insinyur

Inisialisasi proyek AWS SAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori kode AWS SAM.	1. Kloning repositori sampel tumpukan bersarang aws sam untuk pola ini dengan memasukkan perintah berikut.
	`git clone https://github.com/aws-samples/aws-sam-nested-stack-sample.git`
	2. Arahkan ke direktori kloning dengan memasukkan perintah berikut.
	`cd aws-sam-nested-stack-sample`
		DevOps insinyur

Menyebarkan template untuk menginisialisasi proyek.	Untuk menginisialisasi proyek, jalankan `SAM init` perintah. Ketika diminta untuk memilih sumber template, pilih`Custom Template Location`.	DevOps insinyur

Kompilasi dan buat kode template SAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau template aplikasi AWS SAM.	Tinjau template untuk aplikasi bersarang. Contoh ini menggunakan template aplikasi bersarang berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `auth.yaml` — Template ini menyiapkan sumber daya terkait otentikasi, seperti Amazon Cognito dan AWS Systems Manager Parameter Store. • `product-mock.yaml` — Template ini menyebarkan sumber daya terkait produk, seperti fungsi Lambda dan Amazon API Gateway. • `shoppingcart-service.yaml` — Template ini menyiapkan sumber daya terkait keranjang belanja, seperti AWS Identity and Access Management (IAM), tabel DynamoDB, dan fungsi Lambda.	
Tinjau template induk.	Tinjau template yang akan memanggil template aplikasi bersarang. Dalam contoh ini, template induk adalah `template.yml` . Semua aplikasi terpisah bersarang di template `template.yml` induk tunggal.	DevOps insinyur
Kompilasi dan buat kode template AWS SAM.	Menggunakan AWS SAM CLI, jalankan perintah berikut. `sam build`	DevOps insinyur

Terapkan template AWS SAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan aplikasi.	Untuk meluncurkan kode template SAM yang membuat CloudFormation tumpukan aplikasi bersarang dan menyebarkan kode di lingkungan AWS, jalankan perintah berikut. `sam deploy --guided --stack-name shopping-cart-nested-stack --capabilities CAPABILITY_IAM CAPABILITY_AUTO_EXPAND` Perintah akan meminta dengan beberapa pertanyaan. Jawab semua pertanyaan dengany.	DevOps insinyur

Verifikasi penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi tumpukan.	Untuk meninjau CloudFormation tumpukan AWS dan sumber daya AWS yang ditentukan dalam templat AWS SAM, lakukan hal berikut: 1. Masuk ke AWS Management Console, dan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	navigasikan ke CloudForm ationkonsol. 2. Verifikasi bahwa tumpukan induk dan anak terdaftar. Dalam contoh ini, `sam-shopping-cart` adalah tumpukan induk yang memanggil tumpukan Auth, Produk, dan Belanja bersarang. Tumpukan produk memberikan tautan URL Product API Gateway sebagai output.	

Sumber daya terkait

Referensi

- AWS Model Aplikasi Tanpa Server (AWS SAM)
- AWS SAM aktif GitHub
- Serverless Shopping Cart Microservice (aplikasi contoh AWS)

Tutorial dan video

- Membangun Aplikasi Tanpa Server
- AWS Online Tech Talks: Pembuatan dan Penerapan Aplikasi Tanpa Server dengan AWS SAM

Informasi tambahan

Setelah semua kode ada, contoh memiliki struktur direktori berikut:

- sam_stacks — Folder ini berisi layer. `shared.py` Lapisan adalah arsip file yang berisi pustaka, runtime kustom, atau dependensi lainnya. Dengan lapisan, Anda dapat menggunakan pustaka dalam fungsi Anda tanpa perlu memasukkannya ke dalam paket penerapan.
- product-mock-service— Folder ini berisi semua fungsi dan file Lambda terkait produk.
- shopping-cart-service- Folder ini berisi semua fungsi dan file Lambda terkait belanja.

Menerapkan isolasi penyewa SaaS untuk Amazon S3 dengan menggunakan mesin penjual otomatis token AWS Lambda

Tabby Ward, Thomas Davis, dan Sravan Periyathambi, Amazon Web Services

Ringkasan

Aplikasi SaaS multitenant harus menerapkan sistem untuk memastikan bahwa isolasi penyewa dipertahankan. Saat menyimpan data penyewa pada AWS sumber daya yang sama—seperti saat beberapa penyewa menyimpan data dalam ember Amazon Simple Storage Service (Amazon S3) S3—Anda harus memastikan bahwa akses lintas penyewa tidak dapat terjadi. Token vending machine (TVMs) adalah salah satu cara untuk menyediakan isolasi data penyewa. Mesin-mesin ini menyediakan mekanisme untuk mendapatkan token sambil mengabstraksi kompleksitas bagaimana token ini dihasilkan. Pengembang dapat menggunakan TVM tanpa memiliki pengetahuan rinci tentang bagaimana menghasilkan token.

Pola ini mengimplementasikan TVM dengan menggunakan. AWS Lambda TVM menghasilkan token yang terdiri dari kredenal layanan token keamanan sementara (STS) yang membatasi akses ke data penyewa SaaS tunggal dalam bucket S3.

TVMs, dan kode yang disediakan dengan pola ini, biasanya digunakan dengan klaim yang berasal dari JSON Web Tokens (JWTs) untuk mengaitkan permintaan AWS sumber daya dengan kebijakan cakupan penyewa (IAM AWS Identity and Access Management). Anda dapat menggunakan kode dalam pola ini sebagai dasar untuk mengimplementasikan aplikasi SaaS yang menghasilkan kredensi STS sementara cakupan berdasarkan klaim yang diberikan dalam token JWT.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) versi 1.19.0 atau yang lebih baru, diinstal dan dikonfigurasi di macOS, Linux, atau Windows. Atau, Anda dapat menggunakan AWS CLI versi 2.1 atau yang lebih baru.

Batasan

- Kode ini berjalan di Java dan saat ini tidak mendukung bahasa pemrograman lainnya.
- Aplikasi sampel tidak menyertakan dukungan AWS Cross-region atau disaster recovery (DR).

- Pola ini menunjukkan bagaimana TVM Lambda untuk aplikasi SaaS dapat menyediakan akses penyewa cakupan. Pola ini tidak dimaksudkan untuk digunakan di lingkungan produksi tanpa pengujian keamanan tambahan sebagai bagian dari aplikasi atau kasus penggunaan spesifik Anda.

Arsitektur

Tumpukan teknologi target

- AWS Lambda
- Amazon S3
- IAM
- AWS Security Token Service (AWS STS)

Arsitektur target

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Security Token Service (AWS STS) membantu Anda meminta kredensil hak istimewa terbatas sementara untuk pengguna.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Kode

Kode sumber untuk pola ini tersedia sebagai lampiran dan termasuk file-file berikut:

- `s3UploadSample.jar`menyediakan kode sumber untuk fungsi Lambda yang mengunggah dokumen JSON ke bucket S3.
- `tvm-layer.zip`menyediakan pustaka Java yang dapat digunakan kembali yang memasok token (kredensi sementara STS) untuk fungsi Lambda untuk mengakses bucket S3 dan mengunggah dokumen JSON.
- `token-vending-machine-sample-app.zip`menyediakan kode sumber yang digunakan untuk membuat artefak dan instruksi kompilasi ini.

Untuk menggunakan file-file ini, ikuti instruksi di bagian selanjutnya.

Epik

Tentukan nilai variabel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan nilai variabel.	Implementasi pola ini mencakup beberapa nama variabel yang harus digunakan secara konsisten. Tentukan nilai yang harus digunakan untuk setiap variabel, dan berikan nilai itu ketika diminta dalam langkah selanjutnya. • `<AWS Account ID>`— 12 digit ID akun yang terkait dengan Akun AWS Anda menerapkan pola ini. Untuk informasi tentang cara menemukan Akun AWS ID Anda, lihat Menggunakan alias untuk Akun AWS ID Anda dalam dokumentasi IAM.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `<AWS Region>`— Di Wilayah AWS mana Anda menerapkan pola ini. Untuk informasi selengkapnya Wilayah AWS, lihat Wilayah dan Zona Ketersediaan di AWS situs web. • `<sample-tenant-name>` — Nama penyewa untuk digunakan dalam aplikasi. Kami menyarankan Anda hanya menggunakan karakter alfanumerik dalam nilai ini untuk kesederhanaan, tetapi Anda dapat menggunakan nama yang valid untuk kunci objek S3. • `<sample-tvm-role-name>` — Nama peran IAM yang melekat pada fungsi Lambda yang menjalankan TVM dan aplikasi sampel. Nama peran adalah string yang terdiri dari karakter alfanumerik huruf besar dan kecil tanpa spasi. Anda juga dapat menyertakan salah satu karakter berikut: garis bawah (_), tanda plus (+), tanda sama dengan (=), koma (,), titik (.), di tanda (@), dan tanda hubung (-). Nama peran harus unik dalam akun tersebut.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• `<sample-app-role-name>` — Nama peran IAM yang diasumsikan oleh fungsi Lambda ketika menghasilkan kredenal STS sementara yang terbatas. Nama peran adalah string yang terdiri dari karakter alfanumerik huruf besar dan kecil tanpa spasi. Anda juga dapat menyertakan salah satu karakter berikut: garis bawah (_), tanda plus (+), tanda sama dengan (=), koma (,), titik (.), di tanda (@), dan tanda hubung (-). Nama peran harus unik dalam akun tersebut.
	• `<sample-app-function-name>` — Nama fungsi Lambda. Ini adalah string yang panjangnya hingga 64 karakter.
	• `<sample-app-bucket-name>` — Nama bucket S3 yang harus diakses dengan izin yang dicakup oleh penyewa tertentu. Nama ember S3:
	  • Panjangnya harus antara 3 dan 63 karakter.
	  • Harus terdiri dari hanya huruf kecil, angka, titik (.), dan tanda hubung (-).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Harus dimulai dan diakhiri dengan huruf atau angka. • Tidak boleh diformat sebagai alamat IP (misalnya, 192.168.5.4). • Harus unik di dalam partisi. Partisi adalah pengelompokan Wilayah. AWS saat ini memiliki tiga partisi: `aws` untuk Wilayah Standar, `aws-cn` untuk Wilayah Tiongkok, dan `aws-us-gov` untuk. AWS GovCloud (US) Regions	

Buat Bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 untuk aplikasi sampel.	Gunakan AWS CLI perintah berikut untuk membuat bucket S3. Berikan `<sample-app-bucket-name>` nilai dalam cuplikan kode: `aws s3api create-bucket --bucket <sample-app-bucket-name>` Aplikasi sampel Lambda mengunggah file JSON ke bucket ini.	Administrator awan

Buat peran dan kebijakan IAM TVM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran TVM.	Gunakan salah satu AWS CLI perintah berikut untuk membuat peran IAM. Berikan `<sample-tvm-role-name>` nilai dalam perintah. Untuk shell macOS atau Linux: `aws iam create-role \ --role-name <sample-tvm-role-name> \ --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "sts:AssumeRole" ], "Principal": { "Service": [ "lambda.amazonaws.com" ] }, "Condition": { "StringEquals": {`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"aws:SourceAccount":
	"<AWS Account ID>"
	}
	}
	}
	]
	}'</pre>
	Untuk baris perintah Windows:
	<pre>aws iam create-role ^
	--role-name <sample-t
	vm-role-name> ^
	--assume-role-policy-
	document "{\"Versi
	on\": \"2012-10
	-17\", \"Statement
	\": [{\"Effect\":
	\"Allow\", \"Action\
	": [\"sts:AssumeRole
	\"], \"Principal
	\": {\"Service\":
	[\"lambda.amazonaw
	s.com\"]}, \"Conditi
	on\": {\"StringEquals
	\": {\"aws:SourceAccou
	nt\": \"<AWS Account
	ID>\"}}}]}"</pre>
	Contoh aplikasi Lambda mengasumsikan peran ini saat aplikasi dipanggil. Kemampuan untuk mengambil peran aplikasi dengan kebijakan cakupan memberikan izin kode yang lebih luas untuk mengakses bucket S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat kebijakan peran TVM sebaris.	Gunakan salah satu AWS CLI perintah berikut untuk membuat kebijakan IAM. Berikan<sample-tvm-role-name> , **<AWS Account ID>**, dan <sample-app-role-name> nilai dalam perintah.
	
	Untuk shell macOS atau Linux:
	
	`aws iam put-role-policy \ --role-name <sample-tvm-role-name> \ --policy-name assume-app-role \ --policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>" } ]}'`
	
	Untuk baris perintah Windows:
	
	`aws iam put-role-policy ^`
		Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--role-name <sample-tvm-role-name> ^ --policy-name assume-app-role ^ --policy-document "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow\", \"Action\": \"sts:AssumeRole\", \"Resource\": \"arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>\"}]}"` Kebijakan ini melekat pada peran TVM. Ini memberi kode kemampuan untuk mengambil peran aplikasi, yang memiliki izin yang lebih luas untuk mengakses bucket S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Lampirkan kebijakan Lambda yang dikelola.	Gunakan AWS CLI perintah berikut untuk melampirkan kebijakan `AWSLambdaBasicExecutionRole` IAM. Berikan `<sample-tvm-role-name>` nilai dalam perintah:
	`aws iam attach-role-policy \ --role-name <sample-tvm-role-name> \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole`
	Untuk baris perintah Windows:
	`aws iam attach-role-policy ^ --role-name <sample-tvm-role-name> ^ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole`
	Kebijakan terkelola ini dilampirkan pada peran TVM untuk mengizinkan Lambda mengirim log ke Amazon. CloudWatch
		Administrator awan

Buat peran dan kebijakan aplikasi IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran aplikasi.	Gunakan salah satu AWS CLI perintah berikut untuk membuat peran IAM. Berikan<sample-app-role-name> ,<AWS Account ID>, dan <sample-tvm-role-name> nilai dalam perintah. Untuk shell macOS atau Linux: `aws iam create-role \ --role-name <sample-app-role-name> \ --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name>" }, "Action": "sts:AssumeRole" } ]}'` Untuk baris perintah Windows:	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws iam create-role ^ --role-name <sample-app-role-name> ^ --assume-role-policy-document "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow\",\"Principal\": {\"AWS\": \"arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name>\"},\"Action\": \"sts:AssumeRole\" }]}"` Aplikasi sampel Lambda mengasumsikan peran ini dengan kebijakan cakupan untuk mendapatkan akses berbasis penyewa ke bucket S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan peran aplikasi inline.	Gunakan salah satu AWS CLI mmands berikut untuk membuat kebijakan IAM. Berikan `<sample-app-role-name>` dan `<sample-app-bucket-name>` nilai dalam perintah. Untuk shell macOS atau Linux: `aws iam put-role-policy \ --role-name <sample-app-role-name> \ --policy-name s3-bucket-access \ --policy-document '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"], "Resource": "arn:aws:s3:::<sample-app-bucket-name>/*" }, { "Effect": "Allow",`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Action":
	["s3:ListBucket"],
	"Resource
	": "arn:aws:s3:::<sam
	ple-app-bucket-name>"
	}
	]}'</pre>
	Untuk baris perintah Windows:
	<pre>aws iam put-role-policy
	^
	--role-name <sample-a
	pp-role-name> ^
	--policy-name s3-bucket
	-access ^
	--policy-documen
	t "{\"Version\":
	\"2012-10-17\",
	\"Statement\":
	[{\"Effect\": \"Allow
	\", \"Action\":
	[\"s3:PutObject\",
	\"s3:GetObject\",
	\"s3:DeleteObject\
	\"], \"Resource\":
	\"arn:aws:s3:::<sa
	mple-app-bucket-na
	me>/*\"}, {\"Effect\":
	\"Allow\", \"Action\
	\": [\"s3:ListBucket
	\"], \"Resource\":
	\"arn:aws:s3:::<sa
	mple-app-bucket-name>
	\"}]}"</pre>
	Kebijakan ini dilampirkan pada peran aplikasi. Ini menyediakan akses luas ke objek di bucket S3. Saat aplikasi	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sampel mengasumsikan peran tersebut, izin ini akan dicakup oleh penyewa tertentu dengan kebijakan TVM yang dihasilkan secara dinamis.	

Buat aplikasi sampel Lambda dengan TVM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh file sumber yang dikompilasi.	Unduh `tvm-layer.zip` file `s3UploadSample.jar` dan, yang disertakan sebagai lampiran. Kode sumber yang digunakan untuk membuat artefak dan instusi kompilasi ini disediakan di. `token-vending-machine-sample-app.zip`	Administrator awan
Buat layer Lambda.	Gunakan AWS CLI perintah berikut untuk membuat layer Lambda, yang membuat TVM dapat diakses oleh Lambda. **Note** Jika Anda tidak menjalankan perintah ini dari lokasi di mana Anda mengunduh `tvm-layer.zip` , berikan jalur yang benar ke `tvm-layer`	Administrator cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`.zip` dalam `--zip-file` parameter.
	
	`aws lambda publish-layer-version \ --layer-name sample-tokenvending-machine \ --compatible-runtimes java11 \ --zip-file fileb://tvm-layer.zip`
	
	Untuk baris perintah Windows:
	
	`aws lambda publish-layer-version ^ --layer-name sample-tokenvending-machine ^ --compatible-runtimes java11 ^ --zip-file fileb://tvm-layer.zip`
	
	Perintah ini membuat lapisan Lambda yang berisi pustaka TVM yang dapat digunakan kembali.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda.	Gunakan AWS CLI perintah berikut untuk membuat fungsi Lambda. Berikan`<sample-app-function-name>` ,`<AWS Account ID>`,`<AWS Region>`,`<sample-tvm-role-name>` ,`<sample-app-bucket-name>` , dan `<sample-app-role-name>` nilai dalam perintah. **Note** Jika Anda tidak menjalankan perintah ini dari lokasi di mana Anda mengunduh `s3UploadSample.jar` , berikan jalur yang benar ke `s3UploadSample.jar` dalam `--zip-file` parameter. `aws lambda create-function \ --function-name <sample-app-function-name> \ --timeout 30 \ --memory-size 256 \ --runtime java11 \`	Administrator cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--role arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name> \`
	`--handler com.amazon.aws.s3UploadSample.App \`
	`--zip-file fileb://s3UploadSample.jar \`
	`--layers arn:aws:lambda:<AWS Region>:<AWS Account ID>:layer:sample-token-vending-machine:1 \`
	`--environment "Variables={S3_BUCKET=<sample-app-bucket-name>, ROLE=arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>}"`
	
	Untuk baris perintah Windows:
	
	`aws lambda create-function ^`
	`--function-name <sample-app-function-name> ^`
	`--timeout 30 ^`
	`--memory-size 256 ^`
	`--runtime java11 ^`
	`--role arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name> ^`
	`--handler com.amazon.aws.s3UploadSample.App ^`
	`--zip-file fileb://s3UploadSample.jar ^`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--layers arn:aws:lambda:<AWS Region>:<AWS Account ID>:layer:sample-token-vending-machine:1 ^ --environment "Variables={S3_BUCKET=<sample-app-bucket-name>,ROLE=arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>}"` Perintah ini menciptakan fungsi Lambda dengan kode aplikasi sampel dan lapisan TVM terpasang. Ini juga menetapkan dua variabel lingkungan: `S3_BUCKET` dan`ROLE`. Aplikasi sampel menggunakan variabel-variabel ini untuk menentukan peran yang akan diasumsikan dan bucket S3 untuk mengunggah dokumen JSON.	

Uji aplikasi sampel dan TVM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memanggil aplikasi sampel Lambda.	Gunakan salah satu AWS CLI perintah berikut untuk memulai aplikasi sampel Lambda dengan muatan yang diharapkan. Berikan `<sample-app-functi`	Administrator cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	on-name> dan <sample-tenant-name> nilai dalam perintah.
	
	Untuk shell macOS dan Linux:
	
	`aws lambda invoke \ --function <sample-app-function-name> \ --invocation-type RequestResponse \ --payload '{"tenant": "<sample-tenant-name>"}' \ --cli-binary-format raw-in-base64-out response.json`
	
	Untuk baris perintah Windows:
	
	`aws lambda invoke ^ --function <sample-app-function-name> ^ --invocation-type RequestResponse ^ --payload "{\"tenant\": \"<sample-tenant-name>\"}" ^ --cli-binary-format raw-in-base64-out response.json`
	
	Perintah ini memanggil fungsi Lambda dan mengembalikan hasilnya dalam dokumen. response.json Pada banyak sistem berbasis Unix, Anda dapat mengubah	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`response.json /dev/stdout` untuk menampilkan hasil langsung ke shell Anda tanpa membuat file lain. **Note** Mengubah `<sample-tenant-name>` nilai dalam pemanggilan berikutnya dari fungsi Lambda ini mengubah lokasi dokumen JSON dan izin yang diberikan token.	
Lihat bucket S3 untuk melihat objek yang dibuat.	Jelajahi bucket S3 (`<sample-app-bucket-name>`) yang Anda buat sebelumnya. Bucket ini berisi awalan objek S3 dengan nilai. `<sample-tenant-name>` Di bawah awalan itu, Anda akan menemukan dokumen JSON bernama dengan UUID. Memanggil aplikasi sampel beberapa kali menambahkan lebih banyak dokumen JSON.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat log untuk aplikasi sampel di CloudWatch Log.	Lihat log yang terkait dengan fungsi Lambda yang dinamai `<sample-app-function-name>` di CloudWatch Log. Untuk petunjuk, lihat Mengirim log fungsi Lambda ke CloudWatch Log dalam dokumentasi Lambda. Anda dapat melihat kebijakan cakupan penyewa yang dihasilkan oleh TVM di log ini. Kebijakan cakupan penyewa ini memberikan izin untuk aplikasi sampel ke Amazon S3,, dan PutObjectGetObject DeleteObjectListBucket APIs, tetapi hanya untuk awalan objek yang terkait dengannya. `<sample-tenant-name>` Dalam pemanggilan aplikasi sampel berikutnya, jika Anda mengubah`<sample-tenant-name>` , TVM memperbarui kebijakan cakupan agar sesuai dengan penyewa yang disediakan dalam muatan pemanggilan. Kebijakan yang dihasilkan secara dinamis ini menunjukkan bagaimana akses cakupan penyewa dapat dipertahankan dengan TVM di aplikasi SaaS. Fungsionalitas TVM disediakan dalam lapisan Lambda	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sehingga dapat dilampirkan ke fungsi Lambda lain yang digunakan oleh aplikasi tanpa harus mereplikasi kode. Untuk ilustrasi kebijakan yang dihasilkan secara dinamis, lihat bagian Informasi tambahan.	

Sumber daya terkait

- Mengisolasi Penyewa dengan Kebijakan IAM yang Dihasilkan Secara Dinamis (posting blog)
- Menerapkan Kebijakan Isolasi yang Dihasilkan Secara Dinamis di Lingkungan SaaS (posting blog)
- SaaS di AWS

Informasi tambahan

Log berikut menunjukkan kebijakan yang dihasilkan secara dinamis yang dihasilkan oleh kode TVM dalam pola ini. Dalam tangkapan layar ini, `<sample-app-bucket-name>` adalah `DOC-EXAMPLE-BUCKET` dan yang `<sample-tenant-name>` ada`test-tenant-1`. Kredensi STS yang dikembalikan oleh kebijakan cakupan ini tidak dapat melakukan tindakan apa pun pada objek di bucket S3 kecuali untuk objek yang terkait dengan awalan key objek. `test-tenant-1`

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menerapkan pola saga tanpa server dengan menggunakan AWS Step Functions

Tabby Ward, Joe Kern, dan Rohan Mehta, Amazon Web Services

Ringkasan

Dalam arsitektur layanan mikro, tujuan utamanya adalah membangun komponen terpisah dan independen untuk mempromosikan kelincahan, fleksibilitas, dan waktu yang lebih cepat untuk memasarkan aplikasi Anda. Sebagai hasil dari decoupling, setiap komponen microservice memiliki lapisan persistensi datanya sendiri. Dalam arsitektur terdistribusi, transaksi bisnis dapat menjangkau beberapa layanan mikro. Karena layanan mikro ini tidak dapat menggunakan transaksi atomisitas, konsistensi, isolasi, daya tahan (ACID) tunggal, Anda mungkin berakhir dengan transaksi sebagian. Dalam hal ini, diperlukan beberapa logika kontrol untuk membatalkan transaksi yang telah diproses. Pola saga terdistribusi biasanya digunakan untuk tujuan ini.

Pola saga adalah pola manajemen kegagalan yang membantu membangun konsistensi dalam aplikasi terdistribusi dan mengkoordinasikan transaksi antara beberapa layanan mikro untuk menjaga konsistensi data. Ketika Anda menggunakan pola saga, setiap layanan yang melakukan transaksi menerbitkan peristiwa yang memicu layanan berikutnya untuk melakukan transaksi berikutnya dalam rantai. Ini berlanjut sampai transaksi terakhir dalam rantai selesai. Jika transaksi bisnis gagal, saga mengatur serangkaian transaksi kompensasi yang membatalkan perubahan yang dilakukan oleh transaksi sebelumnya.

Pola ini menunjukkan cara mengotomatiskan penyiapan dan penerapan aplikasi sampel (yang menangani reservasi perjalanan) dengan teknologi tanpa server seperti AWS Step Functions, AWS Lambda, dan Amazon DynamoDB. Aplikasi sampel juga menggunakan Amazon API Gateway dan Amazon Simple Notification Service (Amazon SNS) untuk mengimplementasikan koordinator eksekusi saga. Pola ini dapat diterapkan dengan kerangka infrastruktur sebagai kode (IAc) seperti AWS Cloud Development Kit (AWS CDK), AWS Serverless Application Model (AWS SAM), atau Terraform.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Izin untuk membuat CloudFormation tumpukan AWS. Untuk informasi selengkapnya, lihat Mengontrol akses dalam CloudFormation dokumentasi.

- Kerangka kerja IAC pilihan Anda (AWS CDK, AWS SAM, atau Terraform) dikonfigurasi dengan akun AWS Anda sehingga Anda dapat menggunakan kerangka kerja CLI untuk menyebarkan aplikasi.
- NodeJS, digunakan untuk membangun aplikasi dan menjalankannya secara lokal.
- Editor kode pilihan Anda (seperti Visual Studio Code, Sublime, atau Atom).

Versi produk

- NodeJS versi 14
- AWS CDK versi 2.37.1
- AWS SAM versi 1.71.0
- Terraform versi 1.3.7

Batasan

Sumber acara adalah cara alami untuk menerapkan pola orkestrasi saga dalam arsitektur layanan mikro di mana semua komponen digabungkan secara longgar dan tidak memiliki pengetahuan langsung satu sama lain. Jika transaksi Anda melibatkan sejumlah kecil langkah (tiga hingga lima), pola saga mungkin sangat cocok. Namun kompleksitas meningkat dengan jumlah layanan mikro dan jumlah langkah.

Pengujian dan debugging dapat menjadi sulit ketika Anda menggunakan desain ini, karena Anda harus menjalankan semua layanan untuk mensimulasikan pola transaksi.

Arsitektur

Arsitektur target

Arsitektur yang diusulkan menggunakan AWS Step Functions untuk membangun pola saga untuk memesan penerbangan, memesan penyewaan mobil, dan memproses pembayaran untuk liburan.

Diagram alur kerja berikut menggambarkan aliran khas sistem reservasi perjalanan. Alur kerja terdiri dari pemesanan perjalanan udara (" ReserveFlight "), memesan mobil (" ReserveCarRental "), memproses pembayaran (" ProcessPayment "), konfirmasi reservasi penerbangan (" ConfirmFlight "), dan konfirmasi penyewaan mobil (" ConfirmCarRental ") diikuti dengan pemberitahuan keberhasilan ketika langkah-langkah ini selesai. Namun, jika sistem menemukan kesalahan dalam menjalankan salah satu transaksi ini, itu mulai gagal mundur. Misalnya, kesalahan dengan pemrosesan

pembayaran (” ProcessPayment “) memicu pengembalian uang (” RefundPayment “), yang kemudian memicu pembatalan mobil sewaan dan penerbangan (” CancelRentalReservation "dan" CancelFlightReservation “), yang mengakhiri seluruh transaksi dengan pesan kegagalan.

Pola ini menyebarkan fungsi Lambda terpisah untuk setiap tugas yang disorot dalam diagram serta tiga tabel DynamoDB untuk penerbangan, penyewaan mobil, dan pembayaran. Setiap fungsi Lambda membuat, memperbarui, atau menghapus baris di tabel DynamoDB masing-masing, tergantung pada apakah transaksi dikonfirmasi atau digulung kembali. Pola ini menggunakan Amazon SNS untuk mengirim pesan teks (SMS) ke pelanggan, memberi tahu mereka tentang transaksi yang gagal atau berhasil.

Otomatisasi dan skala

Anda dapat membuat konfigurasi untuk arsitektur ini dengan menggunakan salah satu kerangka kerja IAc. Gunakan salah satu tautan berikut untuk IAc pilihan Anda.

- Terapkan dengan AWS CDK
- Terapkan dengan AWS SAM
- Terapkan dengan Terraform

Alat

Layanan AWS

- AWS Step Functions adalah layanan orkestrasi tanpa server yang memungkinkan Anda menggabungkan fungsi AWS Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis. Melalui konsol grafis Step Functions, Anda melihat alur kerja aplikasi Anda sebagai serangkaian langkah berbasis peristiwa.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi dengan skalabilitas yang mulus. Anda dapat menggunakan DynamoDB untuk membuat tabel basis data yang dapat menyimpan dan mengambil data dalam jumlah berapa pun, dan melayani tingkat lalu lintas permintaan apapun.
- AWS Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.

- Amazon API Gateway adalah layanan AWS untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs pada skala apa pun.
- Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak untuk mendefinisikan sumber daya aplikasi cloud Anda dengan menggunakan bahasa pemrograman yang sudah dikenal seperti TypeScript,, JavaScript Python, Java, dan C#/Net.
- AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka untuk membangun aplikasi tanpa server. Ini menyediakan sintaks singkatan untuk mengekspresikan fungsi, database APIs, dan pemetaan sumber peristiwa.

Kode

Kode untuk contoh aplikasi yang menunjukkan pola saga, termasuk template IAC (AWS CDK, AWS SAM, atau Terraform), fungsi Lambda, dan tabel DynamoDB dapat ditemukan di tautan berikut. Ikuti instruksi dalam epik pertama untuk menginstal ini.

- Terapkan dengan AWS CDK
- Terapkan dengan AWS SAM
- Terapkan dengan Terraform

Epik

Instal paket, kompilasi, dan bangun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal paket NPM.	Buat direktori baru, arahkan ke direktori itu di terminal, dan kloning GitHub repositori pilihan Anda dari bagian Kode sebelumnya dalam pola ini. Di folder root yang memiliki `package.json` file, jalankan perintah berikut untuk	Pengembang, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengunduh dan menginstal semua paket Node Package Manager (NPM): `npm install`	
Kompilasi skrip.	Di folder root, jalankan perintah berikut untuk menginstruksikan TypeScript transpiler untuk membuat semua file yang diperlukan JavaScript : `npm run build`	Pengembang, arsitek Cloud
Perhatikan perubahan dan kompilasi ulang.	Di folder root, jalankan perintah berikut di jendela terminal terpisah untuk melihat perubahan kode, dan kompilasi kode ketika mendeteksi perubahan: `npm run watch`	Pengembang, arsitek Cloud
Jalankan pengujian unit (hanya AWS CDK).	Jika Anda menggunakan AWS CDK, di folder root, jalankan perintah berikut untuk melakukan pengujian unit Jest: `npm run test`	Pengembang, arsitek Cloud

Menerapkan sumber daya ke akun AWS target

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan tumpukan demo ke AWS.	**Important**
	Aplikasi ini AWS Region-agnostik. Jika Anda menggunakan profil, Anda harus mendeklarasikan Region secara eksplisit di profil AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) atau melalui variabel lingkungan AWS CLI.
	
	Di folder root, jalankan perintah berikut untuk membuat perakitan penerapan dan menerapkannya ke akun AWS default dan Wilayah.
	
	AWS CDK:
	
	`cdk bootstrap`
	`cdk deploy`
	
	AWS SAM:
	
	`sam build`
	`sam deploy --guided`
		Pengembang, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Terraform:
	`terraform init`
	`terraform apply`
	Langkah ini mungkin memakan waktu beberapa menit untuk menyelesa ikannya. Perintah ini menggunakan kredensya l default yang dikonfigurasi untuk AWS CLI.
	Perhatikan URL API Gateway yang ditampilkan di konsol setelah penerapan selesai. Anda akan memerlukan informasi ini untuk menguji alur eksekusi saga.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bandingkan tumpukan yang diterapkan dengan status saat ini.	Di folder root, jalankan perintah berikut untuk membandingkan tumpukan yang diterapkan dengan status saat ini setelah membuat perubahan pada kode sumber:
	AWS CDK:
	`cdk diff`
	AWS SAM:
	`sam deploy`
	Terraform:
	`terraform plan`
		Pengembang, arsitek Cloud

Uji alur eksekusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji alur eksekusi saga.	Arahkan ke URL API Gateway yang Anda catat di langkah sebelumnya, saat Anda menerapkan tumpukan. URL ini memicu mesin status untuk memulai. Untuk informasi selengkapnya tentang cara memanipulasi aliran mesin status dengan meneruskan parameter URL	Pengembang, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang berbeda, lihat bagian Informasi tambahan. Untuk melihat hasilnya, masuk ke AWS Management Console dan navigasikan ke konsol Step Functions. Di sini, Anda dapat melihat setiap langkah dari mesin saga state. Anda juga dapat melihat tabel DynamoDB untuk melihat catatan yang disisipkan, diperbarui, atau dihapus. Jika Anda sering menyegarkan layar, Anda dapat melihat perubahan status transaksi dari `pending` ke `confirmed` . Anda dapat berlangganan topik SNS dengan memperbarui kode dalam `stateMachine.ts` file dengan nomor ponsel Anda untuk menerima pesan SMS setelah reservasi berhasil atau gagal. Untuk informasi selengkapnya, lihat Amazon SNS di bagian Informasi tambahan.	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Pembersihan sumber daya	Untuk membersihkan sumber daya yang digunakan untuk aplikasi ini, Anda dapat menggunakan salah satu perintah berikut.
	AWS CDK:
	`cdk destroy`
	AWS SAM:
	`sam delete`
	Terraform:
	`terraform destroy`
		Pengembang aplikasi, arsitek Cloud

Sumber daya terkait

Makalah teknis

- Menerapkan Layanan Mikro di AWS
- Lensa Aplikasi Tanpa Server

Dokumentasi layanan AWS

- Memulai dengan AWS CDK
- Memulai AWS SAM
- AWS Step Functions
- Amazon DynamoDB
- AWS Lambda

- Amazon API Gateway
- Amazon SNS

Tutorial

- Lokakarya Langsung untuk Komputasi Tanpa Server

Informasi tambahan

Kode

Untuk tujuan pengujian, pola ini menerapkan API Gateway dan fungsi Lambda pengujian yang memicu mesin status Step Functions. Dengan Step Functions, Anda dapat mengontrol fungsionalitas sistem reservasi perjalanan dengan meneruskan `run_type` parameter untuk meniru kegagalan di "ReserveFlightReserveCarRental," "ProcessPayment," "ConfirmFlight," dan "ConfirmCarRental."

Fungsi `saga` Lambda (`sagaLambda.ts`) mengambil input dari parameter kueri di URL API Gateway, membuat objek JSON berikut, dan meneruskannya ke Step Functions untuk dieksekusi:

```
let input = {
"trip_id": tripID, //  value taken from query parameter, default is AWS request ID
"depart_city": "Detroit",
"depart_time": "2021-07-07T06:00:00.000Z",
"arrive_city": "Frankfurt",
"arrive_time": "2021-07-09T08:00:00.000Z",
"rental": "BMW",
"rental_from": "2021-07-09T00:00:00.000Z",
"rental_to": "2021-07-17T00:00:00.000Z",
"run_type": runType // value taken from query parameter, default is "success"
};
```

Anda dapat bereksperimen dengan aliran yang berbeda dari mesin status Step Functions dengan meneruskan parameter URL berikut:

- Eksekusi Berhasil – https://{api gateway url}
- Penerbangan Cadangan Gagal – https://{api gateway url}? RunType= failFlightsReservation
- Konfirmasikan Penerbangan Gagal – https://{api gateway url}? RunType= failFlightsConfirmation
- Cadangan Sewa Mobil Gagal – https://{api gateway url}? RunType= failCarRental Reservasi
- Konfirmasikan Gagal Sewa Mobil – https://{api gateway url}? RunType= Konfirmasi failCarRental

- Proses Pembayaran Gagal – https://{api gateway url}? runType=Kegagalan pembayaran
- Lulus ID Perjalanan – https://{api gateway url}? tripid= {secara default, ID trip akan menjadi ID permintaan AWS}

Template IAc

Repositori tertaut menyertakan templat IAC yang dapat Anda gunakan untuk membuat seluruh contoh aplikasi reservasi perjalanan.

- Terapkan dengan AWS CDK
- Terapkan dengan AWS SAM
- Terapkan dengan Terraform

Tabel DynamoDB

Berikut adalah model data untuk penerbangan, penyewaan mobil, dan tabel pembayaran.

```
Flight Data Model:
 var params = {
      TableName: process.env.TABLE_NAME,
      Item: {
        'pk' : {S: event.trip_id},
        'sk' : {S: flightReservationID},
        'trip_id' : {S: event.trip_id},
        'id': {S: flightReservationID},
        'depart_city' : {S: event.depart_city},
        'depart_time': {S: event.depart_time},
        'arrive_city': {S: event.arrive_city},
        'arrive_time': {S: event.arrive_time},
        'transaction_status': {S: 'pending'}
      }
    };

Car Rental Data Model:
var params = {
      TableName: process.env.TABLE_NAME,
      Item: {
        'pk' : {S: event.trip_id},
        'sk' : {S: carRentalReservationID},
        'trip_id' : {S: event.trip_id},
```

```
        'id': {S: carRentalReservationID},
        'rental': {S: event.rental},
        'rental_from': {S: event.rental_from},
        'rental_to': {S: event.rental_to},
        'transaction_status': {S: 'pending'}
      }
    };

Payment Data Model:
var params = {
      TableName: process.env.TABLE_NAME,
      Item: {
        'pk' : {S: event.trip_id},
        'sk' : {S: paymentID},
        'trip_id' : {S: event.trip_id},
        'id': {S: paymentID},
        'amount': {S: "750.00"}, // hard coded for simplicity as implementing any
 monetary transaction functionality is beyond the scope of this pattern
        'currency': {S: "USD"},
        'transaction_status': {S: "confirmed"}
      }
    };
```

Fungsi Lambda

Fungsi-fungsi berikut akan dibuat untuk mendukung aliran dan eksekusi mesin state di Step Functions:

- Reserve Flights: Menyisipkan catatan ke dalam tabel DynamoDB Flights dengan `pending` jumlah, `transaction_status` untuk memesan penerbangan.
- Konfirmasi Penerbangan: Memperbarui catatan di tabel DynamoDB Flights, untuk `transaction_status` disetel `confirmed` ke, untuk mengonfirmasi penerbangan.
- Batalkan Reservasi Penerbangan: Menghapus catatan dari tabel DynamoDB Flights, untuk membatalkan penerbangan yang tertunda.
- Reserve Car Rentals: Menyisipkan catatan ke dalam tabel CarRentals DynamoDB dengan, untuk `transaction_status` memesan `pending` sewa mobil.
- Konfirmasikan Penyewaan Mobil: Memperbarui catatan di tabel CarRentals DynamoDB, untuk `transaction_status` mengatur ke, `confirmed` untuk mengkonfirmasi sewa mobil.
- Batalkan Reservasi Penyewaan Mobil: Menghapus catatan dari tabel CarRentals DynamoDB, untuk membatalkan sewa mobil yang tertunda.

- Proses Pembayaran: Menyisipkan catatan ke dalam tabel Pembayaran DynamoDB untuk pembayaran.
- Batalkan Pembayaran: Menghapus catatan dari tabel Pembayaran DynamoDB untuk pembayaran.

Amazon SNS

Aplikasi sampel membuat topik dan langganan berikut untuk mengirim pesan SMS dan memberi tahu pelanggan tentang pemesanan yang berhasil atau gagal. Jika Anda ingin menerima pesan teks saat menguji aplikasi sampel, perbarui langganan SMS dengan nomor telepon Anda yang valid di file definisi mesin negara.

Cuplikan AWS CDK (tambahkan nomor telepon di baris kedua dari kode berikut):

```
const topic = new  sns.Topic(this, 'Topic');
topic.addSubscription(new subscriptions.SmsSubscription('+11111111111'));
const snsNotificationFailure = new tasks.SnsPublish(this ,'SendingSMSFailure', {
topic:topic,
integrationPattern: sfn.IntegrationPattern.REQUEST_RESPONSE,
message: sfn.TaskInput.fromText('Your Travel Reservation Failed'),
});

const snsNotificationSuccess = new tasks.SnsPublish(this ,'SendingSMSSuccess', {
topic:topic,
integrationPattern: sfn.IntegrationPattern.REQUEST_RESPONSE,
message: sfn.TaskInput.fromText('Your Travel Reservation is Successful'),
});
```

Cuplikan AWS SAM (ganti +1111111111 string dengan nomor telepon Anda yang valid):

```
  StateMachineTopic11111111111:
    Type: 'AWS::SNS::Subscription'
    Properties:
      Protocol: sms
      TopicArn:
        Ref: StateMachineTopic
      Endpoint: '+11111111111'
    Metadata:
      'aws:sam:path': SamServerlessSagaStack/StateMachine/Topic/+11111111111/Resource
```

Cuplikan Terraform (ganti +111111111 string dengan nomor telepon Anda yang valid):

```
resource "aws_sns_topic_subscription" "sms-target" {
  topic_arn = aws_sns_topic.topic.arn
  protocol  = "sms"
  endpoint  = "+11111111111"
}
```

Reservasi yang berhasil

Alur berikut menggambarkan reservasi yang berhasil dengan “ReserveFlight,” “ReserveCarRental,” dan "ProcessPayment" diikuti oleh "ConfirmFlight" dan "ConfirmCarRental.” Pelanggan diberitahu tentang pemesanan yang berhasil melalui pesan SMS yang dikirim ke pelanggan topik SNS.

Reservasi gagal

Aliran ini adalah contoh kegagalan dalam pola saga. Jika, setelah memesan penerbangan dan penyewaan mobil, ProcessPayment "" gagal, langkah-langkah dibatalkan dalam urutan terbalik. Reservasi dirilis, dan pelanggan diberitahu tentang kegagalan melalui pesan SMS yang dikirim ke pelanggan topik SNS.

Mengelola aplikasi kontainer lokal dengan menyiapkan Amazon ECS Anywhere dengan AWS CDK

Dr. Rahul Sharad Gaikwad, Amazon Web Services

Ringkasan

Amazon ECS Anywhere adalah perpanjangan dari Amazon Elastic Container Service (Amazon ECS). Anda dapat menggunakan ECS Anywhere untuk menerapkan tugas Amazon ECS asli di lingkungan lokal atau yang dikelola pelanggan. Fitur ini membantu mengurangi biaya dan mengurangi orkestrasi dan operasi kontainer lokal yang kompleks. Anda dapat menggunakan ECS Anywhere untuk menyebarkan dan menjalankan aplikasi kontainer di lingkungan lokal dan cloud. Ini menghilangkan kebutuhan tim Anda untuk mempelajari beberapa domain dan keahlian, atau untuk mengelola perangkat lunak yang kompleks sendiri.

Pola ini menunjukkan langkah-langkah untuk menyiapkan ECS Anywhere dengan menggunakan tumpukan AWS Cloud Development Kit (AWS CDK).

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. (Lihat Menginstal, memperbarui, dan menghapus instalan AWS CLI dalam dokumentasi AWS CLI.)
- AWS CDK Toolkit, diinstal dan dikonfigurasi. (Lihat AWS CDK Toolkit dalam dokumentasi AWS CDK, dan ikuti petunjuk untuk menginstal versi 2 secara global.)
- Manajer paket node (npm), diinstal dan dikonfigurasi untuk AWS CDK di. TypeScript (Lihat Mengunduh dan menginstal Node.js dan npm di dokumentasi npm.)

Batasan

- Untuk batasan dan pertimbangan, lihat Instans eksternal (Amazon ECS Anywhere) di dokumentasi Amazon ECS.

Versi produk

- AWS CDK Toolkit versi 2

- npm versi 7.20.3 atau yang lebih baru
- Node.js versi 16.6.1 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- AWS CloudFormation
- AWS CDK
- Amazon ECS Anywhere
- AWS Identity and Access Management (IAM)

Arsitektur target

Diagram berikut menggambarkan arsitektur sistem tingkat tinggi pengaturan ECS Anywhere menggunakan AWS CDK dengan TypeScript, seperti yang diterapkan oleh pola ini.

1. Saat Anda menerapkan tumpukan AWS CDK, itu membuat CloudFormation tumpukan di AWS.
2. CloudFormation Stack menyediakan kluster Amazon ECS dan sumber daya AWS terkait.
3. Untuk mendaftarkan instans eksternal dengan kluster Amazon ECS, Anda harus menginstal AWS Systems Manager Agent (Agen SSM) di mesin virtual (VM) dan mendaftarkan VM sebagai instans terkelola AWS Systems Manager.
4. Anda juga harus menginstal agen penampung Amazon ECS dan Docker di VM Anda untuk mendaftarkannya sebagai instance eksternal dengan cluster Amazon ECS.
5. Ketika instans eksternal terdaftar dan dikonfigurasi dengan cluster Amazon ECS, instans dapat menjalankan beberapa kontainer di VM Anda, yang terdaftar sebagai instance eksternal.

Otomatisasi dan skala

GitHub Repositori yang disediakan dengan pola ini menggunakan AWS CDK sebagai alat infrastruktur sebagai kode (IAc) untuk membuat konfigurasi untuk arsitektur ini. AWS CDK membantu Anda mengatur sumber daya dan menyiapkan ECS Anywhere.

Alat

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.

Kode

Kode sumber untuk pola ini tersedia di GitHub, di repositori Sampel CDK Amazon ECS Anywhere. Untuk mengkloning dan menggunakan repositori, ikuti instruksi di bagian selanjutnya.

Epik

Verifikasi konfigurasi AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi versi AWS CDK.	Verifikasi versi AWS CDK Toolkit dengan menjalankan perintah berikut: `cdk --version` Pola ini membutuhkan AWS CDK versi 2. Jika Anda memiliki AWS CDK versi sebelumnya, ikuti petunjuk dalam dokumentasi AWS CDK untuk memperbaruinya.	DevOps insinyur
Siapkan kredensil AWS.	Untuk mengatur kredensil, jalankan `aws configure` perintah dan ikuti petunjuknya: `$aws configure`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`AWS Access Key ID [None]: <your-access-key-ID>`
	`AWS Secret Access Key [None]: <your-secret-access-key>`
	`Default region name [None]: <your-Region-name>`
	`Default output format [None]:`	

Bootstrap lingkungan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode AWS CDK.	Kloning repositori GitHub kode untuk pola ini dengan menggunakan perintah: `git clone https://github.com/aws-samples/amazon-ecs-anywhere-cdk-samples.git`	DevOps insinyur
Bootstrap lingkungan.	Untuk menerapkan CloudFormation template AWS ke akun dan Wilayah AWS yang ingin Anda gunakan, jalankan perintah berikut: `cdk bootstrap <account-number>/<Region>`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkap nya, lihat Bootstrapping di dokumentasi AWS CDK.	

Membangun dan menyebarkan proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal dependensi paket dan kompilasi TypeScript file.	Instal dependensi paket dan kompilasi TypeScript file dengan menjalankan perintah berikut:
	`$cd amazon-ecs-anywher e-cdk-samples`
	`$npm install`
	`$npm fund`
	Perintah ini menginstal semua paket dari repositori sampel.
	Important
	Jika Anda mendapatk an kesalahan tentang paket yang hilang, gunakan salah satu perintah berikut:
	`$npm ci`
	— atau —
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`$npm install -g @aws-cdk/<package_name>` Untuk informasi selengkapnya, lihat npm ci dan npm install di dokumentasi npm.	
Bangun proyek.	Untuk membangun kode proyek, jalankan perintah: `npm run build` Untuk informasi selengkap nya tentang membangun dan menerapkan project, lihat aplikasi AWS CDK pertama Anda di dokumentasi AWS CDK.	DevOps insinyur
Deploy proyek.	Untuk menyebarkan kode proyek, jalankan perintah: `cdk deploy`	DevOps insinyur
Verifikasi pembuatan dan output tumpukan.	Buka CloudFormation konsol AWS di https://console.aws.amazon.com/cloudformation, dan pilih tumpukan. `EcsAnywhereStack` Tab Output menunjukkan perintah untuk dijalankan di VM eksternal Anda.	DevOps insinyur

Menyiapkan mesin lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan VM Anda dengan menggunakan Vagrant.	Untuk tujuan demonstrasi, Anda dapat menggunakan HashiCorp Vagrant untuk membuat VM. Vagrant adalah utilitas open-source untuk membangun dan memelihar a lingkungan pengemban gan perangkat lunak virtual portabel. Buat Vagrant VM dengan menjalank an `vagrant up` perintah dari direktori root tempat Vagrantfile ditempatkan. Untuk informasi lebih lanjut, lihat dokumentasi Vagrant.	DevOps insinyur
Daftarkan VM Anda sebagai instance eksternal.	1. Masuk ke Vagrant VM dengan menggunakan perintah. `vagrant ssh` Untuk informasi lebih lanjut, lihat dokumentasi Vagrant. 2. Buat kode aktivasi dan ID yang dapat Anda gunakan untuk mendaftarkan VM Anda dengan AWS Systems Manager dan untuk mengaktif kan instans eksternal Anda. Output dari perintah ini termasuk `ActivationId` dan `ActivationCode` nilai: `aws ssm create-ac tivation --iam-role`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	EcsAnywhereInstanc eRole | tee ssm-activ ation.json 3. Ekspor ID aktivasi dan nilai kode: export ACTIVATIO N_ID=<activation-ID> export ACTIVATIO N_CODE=<activation- code> 4. Unduh skrip penginstalan ke server lokal atau VM Anda: curl -o "ecs-anywhere- install.sh" "https:// amazon-ecs-agent.s 3.amazonaws.com/ec s-anywhere-install -latest.sh" && sudo chmod +x ecs-anywhere- install.sh 5. Jalankan skrip penginstalan di server lokal atau VM Anda: sudo ./ecs-anywhere-ins tall.sh \ --cluster test-ecs- anywhere \ --activation-id $ACTIVATION_ID \ --activation-code $ACTIVATION_CODE \ --region <Region>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang menyiapkan dan mendaftarkan VM Anda, lihat Mendaftarkan instans eksternal ke klaster di dokumentasi Amazon ECS.	

Verifikasi status ECS Anywhere dan VM eksternal.	Untuk memverifikasi apakah kotak virtual Anda terhubung ke bidang kontrol Amazon ECS dan berjalan, gunakan perintah berikut:
	`aws ssm describe-instance-information`
	`aws ecs list-container-instances --cluster $CLUSTER_NAME`
		DevOps insinyur

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan dan hapus sumber daya.	Setelah Anda menelusuri pola ini, Anda harus menghapus sumber daya yang Anda buat untuk menghindari biaya lebih lanjut. Untuk membersihkan, jalankan perintah: `cdk destroy`	DevOps insinyur

Sumber daya terkait

- Dokumentasi Amazon ECS Anywhere
- Demo Amazon ECS Anywhere
- Sampel Workshop Amazon ECS Anywhere

Memodernisasi aplikasi Formulir Web ASP.NET di AWS

Vijai Anand Ramalingam dan Sreelaxmi Pai, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah untuk memodernisasi aplikasi ASP.NET Web Forms lama dan monolit dengan mem-porting ke ASP.NET Core di AWS.

Porting aplikasi ASP.NET Web Forms ke ASP.NET Core membantu Anda memanfaatkan kinerja, penghematan biaya, dan ekosistem Linux yang kuat. Namun, ini bisa menjadi upaya manual yang signifikan. Dalam pola ini, aplikasi lama dimodernisasi secara bertahap dengan menggunakan pendekatan bertahap, dan kemudian dikemas di AWS Cloud.

Pertimbangkan aplikasi monolit warisan untuk keranjang belanja. Mari kita asumsikan bahwa itu dibuat sebagai aplikasi ASP.NET Web Forms dan terdiri dari halaman.aspx dengan file code-behind (). `aspx.cs` Proses modernisasi terdiri dari langkah-langkah berikut:

1. Pecahkan monolit menjadi layanan mikro dengan menggunakan pola dekomposisi yang sesuai. Untuk informasi selengkapnya, lihat panduan Menguraikan monolit menjadi layanan mikro di situs web AWS Prescriptive Guidance.
2. Port aplikasi ASP.NET Web Forms (.NET Framework) lama Anda ke ASP.NET Core di.NET 5 atau yang lebih baru. Dalam pola ini, Anda menggunakan Porting Assistant untuk.NET untuk memindai aplikasi ASP.NET Web Forms Anda dan mengidentifikasi ketidakcocokan dengan ASP.NET Core. Ini mengurangi upaya porting manual.
3. Kembangkan kembali layer UI Formulir Web dengan menggunakan React. Pola ini tidak mencakup pembangunan kembali UI. Untuk petunjuk, lihat Membuat Aplikasi React Baru di dokumentasi React.
4. Kembangkan kembali file kode di belakang Formulir Web (antarmuka bisnis) sebagai API web ASP.NET Core. Pola ini menggunakan NDepend laporan untuk membantu mengidentifikasi file dan dependensi yang diperlukan.
5. Tingkatkan shared/common proyek, seperti Logika Bisnis dan Akses Data, dalam aplikasi lama Anda ke .NET 5 atau yang lebih baru dengan menggunakan Porting Assistant untuk.NET.
6. Tambahkan layanan AWS untuk melengkapi aplikasi Anda. Misalnya, Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan mengakses log aplikasi Anda, dan AWS Systems Manager untuk menyimpan setelan aplikasi Anda.

7. Kontainerisasi aplikasi ASP.NET Core yang dimodernisasi. Pola ini membuat file Docker yang menargetkan Linux di Visual Studio dan menggunakan Docker Desktop untuk mengujinya secara lokal. Langkah ini mengasumsikan bahwa aplikasi lama Anda sudah berjalan di instans Windows Amazon Elastic Compute Cloud (Amazon) lokal atau Amazon EC2. Untuk informasi selengkapnya, lihat pola Menjalankan container ASP.NET Core API Docker pada instance Amazon EC2 Linux.
8. Terapkan aplikasi inti ASP.NET yang dimodernisasi ke Amazon Elastic Container Service (Amazon ECS). Pola ini tidak mencakup langkah penerapan. Untuk petunjuk, lihat Amazon ECS Workshop.

> **Note**
>
> Pola ini tidak mencakup pengembangan UI, modernisasi database, atau langkah penerapan kontainer.

Prasyarat dan batasan

Prasyarat

- Visual Studio atau Visual Studio Code, diunduh dan diinstal.
- Akses ke akun AWS menggunakan AWS Management Console dan AWS Command Line Interface (AWS CLI) versi 2. (Lihat petunjuk untuk mengonfigurasi AWS CLI.)
- AWS Toolkit for Visual Studio (lihat petunjuk penyiapan).
- Docker Desktop, diunduh dan diinstal.
- .NET SDK, diunduh dan diinstal.
- NDepend alat, diunduh dan diinstal. Untuk menginstal NDepend ekstensi untuk Visual Studio, jalankan `NDepend.VisualStudioExtension.Installer` (lihat instruksi). Anda dapat memilih Visual Studio 2019 atau 2022, tergantung pada kebutuhan Anda.
- Porting Assistant untuk.NET, diunduh dan diinstal.

Arsitektur

Memodernisasi aplikasi keranjang belanja

Diagram berikut menggambarkan proses modernisasi untuk aplikasi keranjang belanja ASP.NET warisan.

Arsitektur target

Diagram berikut menggambarkan arsitektur aplikasi keranjang belanja modern di AWS. Web ASP.NET Core APIs dikerahkan ke cluster Amazon ECS. Layanan pencatatan dan konfigurasi disediakan oleh Amazon CloudWatch Logs dan AWS Systems Manager.

Alat

Layanan AWS

- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat untuk menjalankan, menghentikan, dan mengelola kontainer di klaster. Anda dapat menjalankan tugas dan layanan Anda pada infrastruktur tanpa server yang dikelola oleh AWS Fargate. Atau, untuk kontrol lebih besar atas infrastruktur Anda, Anda dapat menjalankan tugas dan layanan Anda pada sekelompok EC2 instance yang Anda kelola.
- Amazon CloudWatch Logs — Amazon CloudWatch Logs memusatkan log dari semua sistem, aplikasi, dan layanan AWS yang Anda gunakan. Anda dapat melihat dan memantau log, mencari kode atau pola kesalahan tertentu, memfilternya berdasarkan bidang tertentu, atau mengarsipkannya dengan aman untuk analisis masa depan.
- AWS Systems Manager – AWS Systems Manager adalah layanan AWS yang dapat Anda gunakan untuk melihat dan mengontrol infrastruktur di AWS. Menggunakan konsol Systems Manager, Anda dapat melihat data operasional dari beberapa layanan AWS dan mengotomatiskan tugas operasional di seluruh sumber daya AWS Anda. Systems Manager membantu Anda menjaga keamanan dan kepatuhan dengan memindai instans dan pelaporan terkelola Anda (atau mengambil tindakan korektif) pada setiap pelanggaran kebijakan yang terdeteksi.

Alat

- Visual Studio atau Visual Studio Code — Alat untuk membangun aplikasi.NET, web APIs, dan program lainnya.
- AWS Toolkit for Visual Studio — Ekstensi untuk Visual Studio yang membantu mengembangkan, men-debug, dan menyebarkan aplikasi.NET yang menggunakan layanan AWS.
- Docker Desktop — Alat yang menyederhanakan pembuatan dan penerapan aplikasi kontainer.

- NDepend— Analyzer yang memantau kode.NET untuk dependensi, masalah kualitas, dan perubahan kode.
- Porting Assistant for .NET — Alat analisis yang memindai kode.NET untuk mengidentifikasi ketidakcocokan dengan.NET Core dan untuk memperkirakan upaya migrasi.

Epik

Port aplikasi lama Anda ke .NET 5 atau versi yang lebih baru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tingkatkan aplikasi lawasan.NET Framework Anda ke .NET 5.	Anda dapat menggunakan Porting Assistant untuk.NET untuk mengonversi aplikasi ASP.NET Web Forms lama Anda ke .NET 5 atau yang lebih baru. Ikuti petunjuk di Porting Assistant untuk dokumentasi .NET.	Pengembang aplikasi
Hasilkan NDepend laporan.	Saat Anda memoderni sasi aplikasi Formulir Web ASP.NET Anda dengan menguraikannya menjadi layanan mikro, Anda mungkin tidak memerlukan semua file.cs dari aplikasi lama. Anda dapat menggunakan NDepend untuk membuat laporan untuk file code-behind (.cs) apa pun, untuk mendapatkan semua penelepon dan pemanggil . Laporan ini membantu Anda mengidentifikasi dan hanya menggunakan file yang diperlukan dalam layanan mikro Anda.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Setelah Anda menginstal NDepend (lihat bagian Prasyarat), buka solusi (file.sln) untuk aplikasi lama Anda di Visual Studio dan ikuti langkah-langkah berikut: 1. Bangun aplikasi lama di Visual Studio. 2. Pada bilah menu Visual Studio, pilih NDepend, Lampirkan NDepend proyek baru ke solusi VS saat ini. 3. Pilih Analisis rakitan .NET. 4. Ketika analisis selesai, navigasikan ke proyek di Solution Explorer. Klik kanan file di belakang kode apa pun (misalnya ,`listproducts.aspx.cs`) yang ingin Anda buat laporannya, lalu pilih Tampilkan pada Grafik Ketergantungan. 5. Di bilah navigasi, pilih Penelepon dan pemanggil, lalu pilih Edit kueri kode. 6. Di panel Query and Rules Edit, pilih panah unduh, lalu pilih Ekspor ke Excel. Proses ini menghasilkan laporan untuk file di belakang kode yang mencantum	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kan semua penelepon dan pemanggil. Untuk informasi selengkapnya tentang grafik ketergantungan, lihat NDepend dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat solusi.NET 5 baru.	Untuk membuat struktur.NET 5 (atau yang lebih baru) baru untuk web ASP.NET Core Anda yang dimodernisasi: APIs 1. Buka Visual Studio. 2. Buat solusi kosong baru. 3. Buat proyek baru yang menargetkan .NET 5 (atau yang lebih baru), berdasarkan aplikasi lama Anda. Untuk contoh proyek lama dan baru untuk aplikasi keranjang belanja, lihat bagian Informasi tambahan. 4. Gunakan NDepend laporan dari langkah sebelumnya untuk mengidentifikasi semua file yang diperlukan. Salin file-file ini dari aplikasi yang Anda tingkatkan sebelumnya dan tambahkan ke solusi baru. 5. Bangun solusinya dan perbaiki semua masalah. Untuk informasi selengkapnya tentang membuat proyek dan solusi, lihat dokumentasi Visual Studio.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Saat Anda membangun solusi dan memverifikasi fungsionalitas, Anda mungkin mengident ifikasi beberapa file tambahan yang akan ditambahkan ke solusi, selain file yang NDepend diidentif ikasi.	

Perbarui kode aplikasi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Implementasikan web APIs dengan ASP.NET Core.	Mari kita asumsikan bahwa salah satu layanan mikro yang Anda identifikasi dalam aplikasi keranjang belanja monolit lama Anda adalah Produk. Anda membuat proyek API web ASP.NET Core baru untuk Produk di epik sebelumnya. Pada langkah ini, Anda mengident ifikasi dan memodernisasi semua formulir web (halaman. aspx) yang terkait dengan Produk. Mari kita asumsikan bahwa Produk terdiri dari empat bentuk web, seperti	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang diilustrasikan sebelumnya di bagian Arsitektur: • Daftar Produk • Lihat Produk • Tambah/Edit Produk • Hapus Produk Anda harus menganalisis setiap formulir web, mengidentifikasi semua permintaan yang dikirim ke database untuk melakukan beberapa logika, dan mendapatkan tanggapan. Anda dapat mengimplementasikan setiap permintaan sebagai titik akhir API web. Mengingat bentuk webnya, Produk dapat memiliki titik akhir yang mungkin berikut: • `/api/products` • `/api/products/{id}` • `/api/products/add` • `/api/products/update/{id}` • `/api/products/delete/{id}` Seperti disebutkan sebelumnya, Anda juga dapat menggunakan kembali semua proyek lain yang Anda	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	upgrade ke .NET 5, termasuk Logika Bisnis, Akses Data, dan shared/common proyek.	
Konfigurasikan CloudWatch Log Amazon.	Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan mengakses log aplikasi Anda. Anda dapat mencatat data ke Amazon CloudWatch Logs menggunakan AWS SDK. Anda juga dapat menginteg rasikan aplikasi.NET dengan CloudWatch Log dengan menggunakan framework logging .NET populer seperti NLog, Log4Net, dan ASP.NET Core logging framework. Untuk informasi selengkap nya tentang langkah ini, lihat posting blog Amazon CloudWatch Logs dan .NET Logging Frameworks.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan AWS Systems Manager Parameter Store.	Anda dapat menggunakan AWS Systems Manager Parameter Store untuk menyimpan setelan aplikasi seperti string koneksi secara terpisah dari kode aplikasi Anda. NuGet Paket Amazon.Extensions. Configuration. SystemsManagermenyederhanakan cara aplikasi Anda memuat pengaturan ini dari AWS Systems Manager Parameter Store ke dalam sistem konfigurasi.NET Core. Untuk informasi selengkapnya tentang langkah ini, lihat posting blog penyedia konfigurasi.NET Core untuk AWS Systems Manager.	Pengembang aplikasi

Tambahkan otentikasi dan otorisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan cookie bersama untuk otentikasi.	Memodernisasi aplikasi monolit warisan adalah proses berulang dan membutuhkan monolit dan versi modernnya untuk hidup berdampingan. Anda dapat menggunakan cookie bersama untuk mencapai otentikasi tanpa	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	batas antara kedua versi. Aplikasi ASP.NET lama terus memvalidasi kredensi pengguna dan mengeluar kan cookie sementara aplikasi ASP.NET Core yang dimodernisasi memvalidasi cookie. Untuk instruksi dan kode sampel, lihat GitHub proyek sampel.	

Bangun dan jalankan kontainer secara lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gambar Docker dengan menggunakan Visual Studio.	Pada langkah ini, Anda membuat file Docker dengan menggunakan Visual Studio for .NET Core web API. 1. Buka Visual Studio. 2. Di Solution Explorer, dari menu konteks (klik kanan) proyek Anda, pilih Add, Docker Support. 3. Pilih Linux sebagai sistem operasi target. Visual Studio membuat file Docker untuk proyek Anda. Untuk contoh file Docker, lihat Visual Studio Container	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Tools for Docker di situs web Microsoft.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bangun dan jalankan wadah dengan menggunakan Docker Desktop.	Sekarang Anda dapat membangun, membuat dan menjalankan wadah di Docker Desktop.
	1. Buka jendela Prompt Perintah. Arahkan ke folder solusi tempat file Docker berada. Jalankan perintah berikut untuk membuat image Docker:
	`docker build -t aspnetcorewebapiimage -f Dockerfile .`
	2. Jalankan perintah berikut untuk melihat semua gambar Docker:
	`docker images`
	3. Jalankan perintah berikut untuk membuat dan menjalankan wadah:
	`docker run -d -p 8080:80 --name aspnetcorewebapicontainer aspnetcorewebapiimage`
	4. Buka Docker Desktop, lalu pilih Containers/Apps. Anda dapat melihat wadah baru bernama `aspnetcor`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`ewebapicontainer` running.	

Sumber daya terkait

- Jalankan wadah API Docker web ASP.NET Core di instans Amazon EC2 Linux (AWS Prescriptive Guidance)
- Lokakarya Amazon ECS
- Lakukan blue/green penerapan ECS dengan menggunakan CodeDeploy AWS (dokumentasi CloudFormation AWS) CloudFormation
- Memulai dengan NDepend (NDepend dokumentasi)
- Asisten Porting untuk .NET

Informasi tambahan

Tabel berikut memberikan contoh proyek sampel untuk aplikasi keranjang belanja lama dan proyek yang setara dalam aplikasi ASP.NET Core Anda yang dimodernisasi.

Solusi warisan:

Nama proyek	Template proyek	Kerangka target
Antarmuka Bisnis	Perpustakaan Kelas	.NET Framework
BusinessLogic	Perpustakaan Kelas	.NET Framework
WebApplication	Aplikasi Web Kerangka ASP.NET	.NET Framework
UnitTests	NUnit Proyek Uji	.NET Framework
Dibagikan -> Umum	Perpustakaan Kelas	.NET Framework
Bersama -> Kerangka	Perpustakaan Kelas	.NET Framework

Solusi baru:

Nama proyek	Template proyek	Kerangka target
BusinessLogic	Perpustakaan Kelas	.NET 5.0
<WebAPI>	API Web Inti ASP.NET	.NET 5.0
<WebAPI>. UnitTests	NUnit 3 Proyek Uji	.NET 5.0
Dibagikan -> Umum	Perpustakaan Kelas	.NET 5.0
Bersama -> Kerangka	Perpustakaan Kelas	.NET 5.0

Jalankan beban kerja yang digerakkan oleh peristiwa dan terjadwal dalam skala besar dengan AWS Fargate

HARI OHM PRASATH RAJAGOPAL, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menjalankan beban kerja terjadwal dan berbasis peristiwa pada skala besar di Amazon Web Services (AWS) Cloud dengan menggunakan AWS Fargate.

Dalam kasus penggunaan pola ini diatur, kode dipindai untuk informasi sensitif AWS, seperti nomor akun AWS dan kredensil, setiap kali permintaan tarik dikirimkan. Permintaan tarik memulai fungsi Lambda. Fungsi Lambda memanggil tugas Fargate yang menangani pemindaian kode. Lambda dimulai setiap kali permintaan tarik baru diajukan. Jika pemindaian menemukan informasi sensitif, Amazon Simple Notification Service (Amazon SNS) mengirimkan hasil pemindaian dalam pesan email.

Pola ini sangat membantu dalam kasus penggunaan bisnis berikut:

- Jika bisnis Anda harus menjalankan banyak beban kerja terjadwal dan berbasis peristiwa yang tidak dapat dijalankan oleh AWS Lambda karena keterbatasan sekitar runtime (batas 15 menit) atau memori
- Jika Anda ingin AWS mengelola instans yang disediakan untuk beban kerja ini

Saat Anda menggunakan pola ini, Anda memiliki opsi untuk membuat virtual private cloud (VPC) baru. Pola ini juga menggunakan AWS CodeCommit.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS CodeCommit untuk menghosting basis kode dan membuat permintaan tarik
- AWS Command Line Interface (AWS CLI) versi 1.7 atau yang lebih baru, diinstal dan dikonfigurasi di macOS, Linux, atau Windows
- Beban kerja berjalan dalam wadah
- Apache Maven yang dapat dieksekusi diatur di classpath

Arsitektur

Aliran keseluruhan mencakup langkah-langkah berikut.

1. Setiap kali permintaan tarik baru dikirimkan CodeCommit, fungsi Lambda dimulai. Fungsi Lambda mendengarkan melalui acara `CodeCommit Pull Request State Change` melalui Amazon. EventBridge
2. Fungsi Lambda mengirimkan tugas Fargate baru dengan parameter lingkungan berikut untuk memeriksa kode dan memindainya.

   ```
   RUNNER # <<TaskARN>>
   SNS_TOPIC # <<SNSTopicARN>>
   SUBNET # <<Subnet in which Fargate task gets launched>>
   ```

   Jika pemindaian menemukan informasi sensitif dalam kode, Fargate mendorong pesan baru ke topik Amazon SNS.
3. Pelanggan SNS membaca pesan dari topik dan mengirim pesan email.

Teknologi

- AWS CodeCommit
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon Elastic Container Service (Amazon ECS)
- Amazon EventBridge
- AWS Fargate
- AWS Lambda
- Amazon SNS
- Docker

Alat

Alat

- AWS CLI — AWS Command Line Interface (CLI) adalah alat terpadu untuk mengelola layanan AWS Anda.
- AWS CodeCommit — AWS CodeCommit adalah layanan kontrol sumber terkelola penuh yang menampung repositori berbasis Git yang aman. Dengan menggunakan CodeCommit, tim dapat berkolaborasi dalam kode dalam lingkungan yang aman dan sangat skalabel.
- Amazon ECR - Amazon Elastic Container Registry (Amazon ECR) adalah registri terkelola penuh yang dapat digunakan pengembang untuk menyimpan, mengelola, dan menyebarkan gambar kontainer Docker.
- Amazon ECS - Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang sangat skalabel dan cepat. Anda dapat menggunakan Amazon ECS untuk menjalankan, menghentikan, dan mengelola kontainer di klaster.
- AWS Fargate — AWS Fargate adalah teknologi yang dapat Anda gunakan dengan Amazon ECS untuk menjalankan container tanpa harus mengelola server atau cluster instans Amazon. EC2
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan (juga dikenal sebagai produsen dan konsumen). Penerbit berkomunikasi secara asinkron dengan pelanggan dengan mengirim pesan ke topik, yang merupakan titik akses logis dan saluran komunikasi. Klien yang berlangganan topik SNS menerima pesan yang dipublikasikan menggunakan protokol yang didukung, seperti Lambda, email, pemberitahuan push seluler, dan pesan teks seluler (SMS).
- Docker — Docker membantu Anda membangun, menguji, dan mengirimkan aplikasi dalam paket yang disebut kontainer.
- Klien Git — Baris perintah atau alat desktop untuk memeriksa artefak yang diperlukan
- Maven - Apache Maven adalah alat manajemen proyek untuk mengelola pembangunan, pelaporan, dan dokumentasi proyek secara terpusat.

Epik

Siapkan repositori lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kodenya.	Di bagian Lampiran, unduh file.zip dan ekstrak file.	Pengembang, administrator sistem AWS
Siapkan repo.	Jalankan `mvn clean install` di folder root.	Pengembang, administrator sistem AWS

Buat gambar Amazon ECR dan dorong gambar

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori Amazon ECR dan masuk.	Buka konsol Amazon ECR. Di panel navigasi, pilih Repositori, lalu pilih Buat repositori. Untuk bantuan tentang ini dan cerita lainnya, lihat bagian Sumber daya terkait.	Pengembang, administrator sistem AWS
Dorong gambar kontainer Anda.	Buka repositori, pilih Lihat perintah push, dan masuk ke Docker. Setelah Anda masuk, jalankan perintah, dengan substitusi yang diperlukan, yang ada di bawah Dorong gambar kontainer di bagian Informasi tambahan. Ini mengunggah gambar kontainer Docker yang digunakan untuk melakukan pemindaian kode. Setelah unggahan selesai, salin URL	Pengembang, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	build terbaru di repositori Amazon ECR.	

Buat CodeCommit repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodeCommit repositori.	Untuk membuat CodeCommit repositori AWS baru, jalankan perintah di bawah Buat CodeCommit repositori di bagian Informasi tambahan.	Pengembang, administrator sistem AWS

Buat VPC (opsional)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Jika Anda ingin menggunak an VPC baru daripada yang sudah ada, jalankan perintah di bawah Buat VPC di bagian Informasi tambahan. Skrip AWS Cloud Development Kit (AWS CDK) akan menampilk an VPC dan subnet yang dibuat. IDs	Pengembang, administrator sistem AWS

Buat klaster Amazon ECS dan tugas Fargate

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster dan tugas.	Untuk membuat klaster Amazon ECS dan definisi	Pengembang, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tugas Fargate, jalankan perintah di bawah Buat klaster dan tugas di bagian Informasi tambahan. Pastikan ID VPC yang benar dan URI repo Amazon ECR diteruskan sebagai parameter saat menjalankan skrip shell. Skrip membuat definisi tugas Fargate yang menunjuk ke gambar Docker (bertanggung jawab untuk pemindaian). Script kemudian membuat pekerjaan dan peran eksekusi terkait.	
Verifikasi cluster Amazon ECS.	Buka konsol Amazon ECS. Di panel navigasi, pilih Clusters, dan pilih cluster Amazon ECS yang baru dibuat bernama Fargate-Job-Cluster. Setelah ini, pilih Definisi tugas di panel navigasi, dan konfirmasikan bahwa ada definisi tugas baru dengan awalan `awscdkfargateecsTaskDef` .	Pengembang, administrator sistem AWS

Buat topik SNS dan pelanggan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat sebuah topik SNS.	Untuk membuat topik SNS, jalankan perintah di bawah Buat topik SNS di bagian Informasi tambahan. Setelah	Pengembang, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pembuatan berhasil, perhatikan `SNS ARN`, yang digunakan pada langkah berikutnya.	
Buat pelanggan SNS.	Untuk membuat pelanggan email untuk topik SNS, jalankan perintah di bawah Buat pelanggan SNS di bagian Informasi tambahan. Pastikan untuk mengganti `TopicARN` dan `Email address` digunakan dalam perintah CLI. Untuk menerima notifikasi email, pastikan untuk mengonfirmasi alamat email yang digunakan sebagai pelanggan.	Pengembang, administrator sistem AWS

Buat fungsi dan pemicu Lambda CodeCommit

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi dan pemicunya.	Untuk membuat fungsi Lambda dengan CodeCommit pemicu, jalankan perintah di bawah fungsi Lambda dan CodeCommit picu di bagian Informasi tambahan. Pastikan untuk mengganti parameter dengan nilai yang sesuai sebelum menjalankan perintah. Skrip membuat fungsi Lambda dan mengonfig	Pengembang, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	urasinya untuk dipanggil saat permintaan tarik baru dibuat.	

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aplikasi.	Jika Anda memeriksa informasi sensitif AWS apa pun ke CodeCommit repo, fungsi Lambda harus dimulai. Fungsi Lambda memulai tugas Fargate, yang memindai kode dan mengirimkan hasil pemindaian dalam pemberita huan email.	Pengembang, administrator sistem AWS

Sumber daya terkait

- Membuat repositori Amazon ECR
- Mendorong gambar Docker ke Amazon ECR

Informasi tambahan

Dorong gambar wadah

```
> cd 1-ecr-image-push
> ./run.sh <<ecr-repository>>
```

Buat CodeCommit repositori

```
aws codecommit create-repository --repository-name test-repo --repository-description
 "My Test repository"
```

Buat VPC

```
> cd 2-create-vpc
> ./run.sh
```

Keluaran

```
aws-batch-cdk-vpc-efs-launch-template.privatesubnet = subnet-<<id>>
aws-batch-cdk-vpc-efs-launch-template.publicsubnet = subnet-<<id>>
aws-batch-cdk-vpc-efs-launch-template.vpcid = vpc-<<id>>
```

Buat cluster dan tugas

```
> export CDK_DEFAULT_ACCOUNT = <<aws_account_id>>
> export CDK_DEFAULT_REGION = <<aws_region>>
> cd 3-create-ecs-task
> ./run.sh <<vpc-id>> <<ecr-repo-uri>>
```

Keluaran

```
aws-cdk-fargate-ecs.CLUSTERNAME = Fargate-Job-Cluster
aws-cdk-fargate-ecs.ClusterARN = <<cluster_arn>>
aws-cdk-fargate-ecs.ContainerARN = Fargate-Container
aws-cdk-fargate-ecs.TaskARN = <<task_arn>>
aws-cdk-fargate-ecs.TaskExecutionRole = <<execution_role_arn>>
aws-cdk-fargate-ecs.TaskRole = <<task_role_arn>>
```

Buat topik SNS

```
aws sns create-topic --name code-commit-topic
```

Buat pelanggan SNS

```
aws sns subscribe \
    --topic-arn <<topic_arn>> \
    --protocol email \
    --notification-endpoint <<email_address>>
```

Fungsi dan pemicu Lambda CodeCommit

```
> export CDK_DEFAULT_ACCOUNT = <<aws_account_id>>
> export CDK_DEFAULT_REGION = <<aws_region>>
> cd 5-Lambda-CodeCommit-Trigger
> ./run.sh <<taskarn>> <<snstopicarn>> subnet-<<id>> <<codecommitarn>>
```

Keluaran

```
aws-cdk-fargate-lambda-event.Cloudwatchrule = <<cloudwatchrule>>
aws-cdk-fargate-lambda-event.CodeCommitLambda = AWS-Code-Scanner-Function
aws-cdk-fargate-lambda-event.LambdaRole = <<lambdaiamrole>>
```

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Orientasi penyewa dalam arsitektur SaaS untuk model silo menggunakan C# dan AWS CDK

Tabby Ward, Susmitha Reddy Gankidi, dan Vijai Anand Ramalingam, Amazon Web Services

Ringkasan

Aplikasi perangkat lunak sebagai layanan (SaaS) dapat dibangun dengan berbagai model arsitektur yang berbeda. Model silo mengacu pada arsitektur di mana penyewa disediakan sumber daya khusus.

Aplikasi SaaS mengandalkan model tanpa gesekan untuk memperkenalkan penyewa baru ke lingkungan mereka. Ini sering membutuhkan orkestrasi sejumlah komponen untuk berhasil menyediakan dan mengkonfigurasi semua elemen yang diperlukan untuk membuat penyewa baru. Proses ini, dalam arsitektur SaaS, disebut sebagai penyewa on-boarding. On-boarding harus sepenuhnya otomatis untuk setiap lingkungan SaaS dengan memanfaatkan infrastruktur sebagai kode dalam proses on-boarding Anda.

Pola ini memandu Anda melalui contoh pembuatan penyewa dan penyediaan infrastruktur dasar untuk penyewa di Amazon Web Services (AWS). Pola ini menggunakan C# dan AWS Cloud Development Kit (AWS CDK).

Karena pola ini menciptakan alarm penagihan, sebaiknya gunakan tumpukan di US East (Virginia N.), atau us-east-1, Wilayah AWS. Untuk informasi selengkapnya, lihat dokumentasi AWS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Prinsipal AWS Identity and Access Management (IAM) dengan akses IAM yang memadai untuk membuat sumber daya AWS untuk pola ini. Untuk informasi selengkapnya, lihat peran IAM.
- Instal Amazon Command Line Interface (AWS CLI) dan konfigurasikan AWS CLI untuk melakukan penerapan AWS CDK.
- Visual Studio 2022 diunduh dan diinstal atau Visual Studio Code diunduh dan diinstal.
- AWS Toolkit untuk penyiapan AWS Toolkit for Visual Studio.
- .NET Core 3.1 atau yang lebih baru (diperlukan untuk aplikasi CDK C# AWS)
- Amazon.Lambda.Tools diinstal.

Keterbatasan

- AWS CDK menggunakan AWS CloudFormation, sehingga aplikasi AWS CDK tunduk pada kuota CloudFormation layanan. Untuk informasi selengkapnya, lihat CloudFormation kuota AWS.
- CloudFormation Tumpukan penyewa dibuat dengan peran CloudFormation `infra-cloudformation-role` layanan dengan karakter wildcard pada tindakan (`sns*` dan`sqs*`) tetapi dengan sumber daya yang dikunci ke awalan. `tenant-cluster` Untuk kasus penggunaan produksi, evaluasi pengaturan ini dan berikan hanya akses yang diperlukan ke peran layanan ini. Fungsi `InfrastructureProvision` Lambda juga menggunakan karakter wildcard (`cloudformation*`) untuk menyediakan CloudFormation tumpukan tetapi dengan sumber daya yang dikunci ke awalan. `tenant-cluster`
- Contoh docker build kode ini digunakan `--platform=linux/amd64` untuk memaksa gambar `linux/amd64` berbasis. Ini untuk memastikan bahwa artefak gambar akhir akan cocok untuk Lambda, yang secara default menggunakan arsitektur x86-64. Jika Anda perlu mengubah arsitektur Lambda target, pastikan untuk mengubah kode Dockerfiles dan AWS CDK. Untuk informasi selengkapnya, lihat posting blog Memigrasi fungsi AWS Lambda ke prosesor AWS Graviton2 berbasis ARM.
- Proses penghapusan tumpukan tidak akan membersihkan CloudWatch Log (grup log dan log) yang dihasilkan oleh tumpukan. Anda harus membersihkan log secara manual melalui konsol AWS Management CloudWatch Console Amazon atau melalui API.

Pola ini diatur sebagai contoh. Untuk penggunaan produksi, evaluasi pengaturan berikut dan buat perubahan berdasarkan kebutuhan bisnis Anda:

- Bucket AWS Simple Storage Service (Amazon S3) dalam contoh ini tidak mengaktifkan versi untuk kesederhanaan. Evaluasi dan perbarui pengaturan sesuai kebutuhan.
- Contoh ini menyiapkan titik akhir Amazon API Gateway REST API tanpa autentikasi, otorisasi, atau pembatasan untuk kesederhanaan. Untuk penggunaan produksi, kami merekomendasikan mengintegrasikan sistem dengan infrastruktur keamanan bisnis. Evaluasi pengaturan ini dan tambahkan pengaturan keamanan yang diperlukan sesuai kebutuhan.
- Untuk contoh infrastruktur penyewa ini, Amazon Simple Notification Service (Amazon SNS) dan Amazon Simple Queue Service (Amazon SQS) hanya memiliki pengaturan minimum. AWS Key Management Service (AWS KMS) untuk setiap penyewa terbuka untuk layanan Amazon dan Amazon CloudWatch SNS di akun untuk dikonsumsi berdasarkan kebijakan kunci AWS KMS. Pengaturan hanya contoh placeholder. Sesuaikan pengaturan sesuai kebutuhan berdasarkan kasus penggunaan bisnis Anda.

- Seluruh penyiapan, yang mencakup tetapi tidak terbatas pada titik akhir API dan penyediaan dan penghapusan penyewa backend dengan menggunakan AWS CloudFormation, hanya mencakup kasus happy path dasar. Evaluasi dan perbarui pengaturan dengan logika coba lagi yang diperlukan, logika penanganan kesalahan tambahan, dan logika keamanan berdasarkan kebutuhan bisnis Anda.
- Kode contoh diuji dengan up-to-date cdk-nag untuk memeriksa kebijakan pada saat penulisan ini. Kebijakan baru mungkin ditegakkan di masa depan. Kebijakan baru ini mungkin mengharuskan Anda untuk memodifikasi tumpukan secara manual berdasarkan rekomendasi sebelum tumpukan dapat diterapkan. Tinjau kode yang ada untuk memastikan bahwa itu selaras dengan kebutuhan bisnis Anda.
- Kode bergantung pada AWS CDK untuk menghasilkan akhiran acak alih-alih mengandalkan nama fisik statis yang ditetapkan untuk sebagian besar sumber daya yang dibuat. Pengaturan ini untuk memastikan bahwa sumber daya ini unik dan tidak bertentangan dengan tumpukan lain. Untuk informasi selengkapnya, lihat dokumentasi AWS CDK. Sesuaikan ini berdasarkan kebutuhan bisnis Anda.
- Contoh kode ini mengemas artefak .NET Lambda ke dalam gambar berbasis Docker dan berjalan dengan runtime image Container yang disediakan Lambda. Runtime image container memiliki keunggulan untuk mekanisme transfer dan penyimpanan standar (pendaftar kontainer) dan lingkungan pengujian lokal yang lebih akurat (melalui image container). Anda dapat mengganti proyek untuk menggunakan runtime .NET yang disediakan Lambda untuk mengurangi waktu pembuatan gambar Docker, tetapi Anda kemudian perlu mengatur mekanisme transfer dan penyimpanan dan memastikan bahwa pengaturan lokal cocok dengan pengaturan Lambda. Sesuaikan kode agar selaras dengan persyaratan bisnis pengguna.

Versi produk

- AWS CDK versi 2.45.0 atau yang lebih baru
- Studio Visual 2022

Arsitektur

Tumpukan teknologi

- Amazon API Gateway
- AWS CloudFormation
- Amazon CloudWatch

- Amazon DynamoDB
- AWS Identity and Access Management (IAM)
- AWS KMS
- AWS Lambda
- Amazon S3
- Amazon SNS
- Amazon SQS

Arsitektur

Diagram berikut menunjukkan aliran pembuatan tumpukan penyewa. Untuk informasi selengkapnya tentang tumpukan teknologi pesawat kontrol dan penyewa, lihat bagian Informasi tambahan.

Aliran pembuatan tumpukan penyewa

1. Pengguna mengirimkan permintaan POST API dengan payload penyewa baru (nama penyewa, deskripsi penyewa) di JSON ke REST API yang dihosting oleh Amazon API Gateway. API Gateway memproses permintaan dan meneruskannya ke fungsi backend Lambda Tenant On-boarding. Dalam contoh ini, tidak ada otorisasi atau otentikasi. Dalam pengaturan produksi, API ini harus diintegrasikan dengan sistem keamanan infrastruktur SaaS.
2. Fungsi Tenant On-boarding memverifikasi permintaan. Kemudian mencoba untuk menyimpan catatan penyewa, yang mencakup nama penyewa, pengidentifikasi unik universal penyewa (UUID) yang dihasilkan, dan deskripsi penyewa, ke dalam tabel On-boarding Amazon DynamoDB Tenant.
3. Setelah DynamoDB menyimpan catatan, aliran DynamoDB memulai fungsi Infrastruktur Penyewa Lambda hilir.
4. Fungsi Lambda Infrastruktur Penyewa bertindak berdasarkan aliran DynamoDB yang diterima. Jika aliran untuk acara INSERT, fungsi menggunakan NewImage bagian aliran (catatan pembaruan terbaru, bidang Nama Penyewa) untuk memanggil CloudFormation untuk membuat infrastruktur penyewa baru menggunakan templat yang disimpan di bucket S3. CloudFormation Template membutuhkan parameter Nama Penyewa.
5. AWS CloudFormation membuat infrastruktur penyewa berdasarkan CloudFormation template dan parameter input.
6. Setiap pengaturan infrastruktur penyewa memiliki CloudWatch alarm, alarm penagihan, dan acara alarm.

7. Acara alarm menjadi pesan ke topik SNS, yang dienkripsi oleh kunci AWS KMS penyewa.
8. Topik SNS meneruskan pesan alarm yang diterima ke antrean SQS, yang dienkripsi oleh AWS KMS penyewa untuk kunci enkripsi.

Sistem lain dapat diintegrasikan dengan Amazon SQS untuk melakukan tindakan berdasarkan pesan dalam antrian. Dalam contoh ini, untuk menjaga kode generik, pesan masuk tetap dalam antrian dan memerlukan penghapusan manual.

Aliran penghapusan tumpukan penyewa

1. Pengguna mengirimkan permintaan DELETE API dengan payload penyewa baru (nama penyewa, deskripsi penyewa) di JSON ke REST API yang dihosting oleh Amazon API Gateway, yang akan memproses permintaan dan meneruskan ke fungsi Tenant On-boarding. Dalam contoh ini, tidak ada otorisasi atau otentikasi. Dalam pengaturan produksi, API ini akan diintegrasikan dengan sistem keamanan infrastruktur SaaS.
2. Fungsi Tenant On-Boarding akan memverifikasi permintaan dan kemudian mencoba menghapus catatan penyewa (nama penyewa) dari meja On-Boarding Penyewa.
3. Setelah DynamoDB berhasil menghapus catatan (catatan ada di tabel dan dihapus), aliran DynamoDB memulai fungsi Infrastruktur Penyewa Lambda hilir.
4. Fungsi Lambda Infrastruktur Penyewa bertindak berdasarkan catatan aliran DynamoDB yang diterima. Jika aliran untuk acara REMOVE, fungsi menggunakan OldImage bagian rekaman (informasi rekaman dan bidang Nama Penyewa, sebelum perubahan terbaru, yang dihapus) untuk memulai penghapusan tumpukan yang ada berdasarkan informasi rekaman tersebut.
5. AWS CloudFormation menghapus tumpukan penyewa target sesuai dengan input.

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS CDK Toolkit adalah kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan aplikasi AWS Cloud Development Kit (AWS CDK).

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.
- AWS Toolkit for Visual Studio adalah plugin untuk lingkungan pengembangan terintegrasi Visual Studio (IDE). Toolkit for Visual Studio mendukung pengembangan, debugging, dan penerapan aplikasi.NET yang menggunakan layanan AWS.

Alat-alat lainnya

- Visual Studio adalah IDE yang mencakup kompiler, alat penyelesaian kode, desainer grafis, dan fitur lain yang mendukung pengembangan perangkat lunak.

Kode

Kode untuk pola ini ada di orientasi Tenant di SaaS Architecture for Silo Model APG Example repository.

Epik

Mengatur AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi instalasi Node.js.	Untuk memverifikasi bahwa Node.js diinstal pada mesin lokal Anda, jalankan perintah berikut. `node --version`	Administrator AWS, AWS DevOps
Instal AWS CDK Toolkit.	Untuk menginstal AWS CDK Toolkit di komputer lokal Anda, jalankan perintah berikut. `npm install -g aws-cdk` Jika npm tidak diinstal, Anda dapat menginstalnya dari situs Node.js.	Administrator AWS, AWS DevOps
Verifikasi versi AWS CDK Toolkit.	Untuk memverifikasi bahwa versi AWS CDK Toolkit diinstal dengan benar di mesin Anda, jalankan perintah berikut. `cdk --version`	Administrator AWS, AWS DevOps

Tinjau kode untuk pesawat kontrol orientasi penyewa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Kloning repositori, dan arahkan ke folder. `\tenant-`	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`onboarding-in-saas-architecture-for-silo-model-apg-example` Di Visual Studio 2022, buka `\src\TenantOnboardingInfra.sln` solusinya. Buka `TenantOnboardingInfraStack.cs` file dan tinjau kodenya. Sumber daya berikut dibuat sebagai bagian dari tumpukan ini: • Tabel DynamoDB • Bucket S3 (Unggah CloudFormation template ke bucket S3.) • Peran eksekusi Lambda • Fungsi Lambda • API Gateway API • Sumber acara ke fungsi Lambda	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau CloudFormation template.	Di `\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\template` folder, buka `infra.yaml` , dan tinjau CloudFormation template. Template ini akan terhidrasi dengan nama penyewa diambil dari tabel DynamoDB orientasi penyewa. Template menyediakan infrastruktur khusus penyewa. Dalam contoh ini, ia menyediakan kunci AWS KMS, Amazon SNS, Amazon SQS, dan alarm. CloudWatch	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau fungsi orientasi penyewa.	Buka `Function.cs`, dan tinjau kode untuk fungsi orientasi penyewa, yang dibuat dengan templat Visual Studio AWS Lambda Project (.NET Core- C#) dengan cetak biru .NET 6 (Gambar Kontainer). Buka `Dockerfile`, dan tinjau kodenya. `Dockerfile` Ini adalah file teks yang terdiri dari instruksi untuk membangun gambar wadah Lambda. Perhatikan bahwa NuGet paket-paket berikut ditambahkan sebagai dependensi ke proyek: `TenantOnboardingFunction` • `Amazon.Lambda.APIGatewayEvents` • `AWSSDK.DynamoDBv2` • `Newtonsoft.Json`	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau InfraProvisioning fungsi Penyewa.	Navigasi ke `\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\InfraProvisioningFunction` . Buka `Function.cs` , dan tinjau kode untuk fungsi penyediaan infrastruktur penyewa, yang dibuat dengan templat Visual Studio AWS Lambda Project (.NET Core-C#) dengan cetak biru .NET 6 (Gambar Kontainer). Buka `Dockerfile` , dan tinjau kodenya. Perhatikan bahwa NuGet paket-paket berikut ditambahkan sebagai dependensi ke proyek: `InfraProvisioningFunction` • `Amazon.Lambda.DynamoDBEvents` • `AWSSDK.DynamoDBv2` • `AWSSDK.Cloudformation`	Pengembang aplikasi, AWS DevOps

Menerapkan sumber daya AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bangun solusinya.	Untuk membangun solusi, lakukan langkah-langkah berikut: 1. Di Visual Studio 2022, buka `\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\TenantOnboardingInfra.sln` solusinya. 2. Buka menu konteks (klik kanan) untuk solusinya, dan pilih Build solution. **Note** Pastikan Anda memperbarui `Amazon.CDK.Lib` `NuGet` paket ke versi terbaru dalam `\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\TenantOnboardingInfra`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	proyek sebelum Anda membuat solusi.	

Bootstrap lingkungan AWS CDK.	Buka prompt perintah Windows dan arahkan ke folder root aplikasi AWS CDK tempat `cdk.json` file tersedia (`\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example`). Jalankan perintah berikut untuk bootstrap.
	`cdk bootstrap`
	Jika Anda telah membuat profil AWS untuk kredensialnya, gunakan perintah dengan profil Anda.
	`cdk bootstrap --profile <profile name>`
		Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat daftar tumpukan AWS CDK.	Untuk daftar semua tumpukan yang akan dibuat sebagai bagian dari proyek ini, jalankan perintah berikut.
	`cdk ls`
	`cdk ls --profile <profile name>`
	Jika Anda telah membuat profil AWS untuk kredensia lnya, gunakan perintah dengan profil Anda.
	`cdk ls --profile <profile name>`
		Administrator AWS, AWS DevOps

Tinjau sumber daya AWS mana yang akan dibuat.	Untuk meninjau semua sumber daya AWS yang akan dibuat sebagai bagian dari proyek ini, jalankan perintah berikut.
	`cdk diff`
	Jika Anda telah membuat profil AWS untuk kredensia lnya, gunakan perintah dengan profil Anda.
	`cdk diff --profile <profile name>`
		Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan semua sumber daya AWS dengan menggunakan AWS CDK.	Untuk menerapkan semua sumber daya AWS, jalankan perintah berikut.
	`cdk deploy --all --require-approval never`
	Jika Anda telah membuat profil AWS untuk kredensialnya, gunakan perintah dengan profil Anda.
	`cdk deploy --all --require-approval never --profile <profile name>`
	Setelah penerapan selesai, salin URL API dari bagian output di prompt perintah, yang ditampilkan dalam contoh berikut.
	`Outputs: TenantOnboardingInfraStack.TenantOnboardingAPIEndpoint42E526D7 = https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/`
		Administrator AWS, AWS DevOps

Verifikasi fungsionalitas

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat penyewa baru.	Untuk membuat penyewa baru, kirim permintaan curl berikut.
	`curl -X POST <TenantOnboardingAPIEndpoint* from CDK Output>tenant -d '{"Name":"Tenant123", "Description":"Stack for Tenant123"}'`
	Ubah pemegang tempat `<TenantOnboardingAPIEndpoint* from CDK Output>` ke nilai aktual dari AWS CDK, seperti yang ditunjukkan pada contoh berikut.
	`curl -X POST https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/tenant -d '{"Name": "Tenant123", "Description":"test12"}'`
	Contoh berikut menunjukkan output.
	`{"message": "A new tenant added - 5/4/2022 7:11:30 AM"}`
		Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi detail penyewa yang baru dibuat di DynamoDB.	Untuk memverifikasi detail penyewa yang baru dibuat di DynamoDB, lakukan langkah-langkah berikut. 1. Buka AWS Management Console, dan navigasikan ke layanan Amazon DynamoDB. 2. Di navigasi kiri, pilih Jelajahi item, dan pilih `TenantOnboarding` tabel. **Note** Nama penyewa akan ditambahkan dengan. `tenantcluster-` Untuk informasi selengkapnya, lihat bagian Informasi tambahan. 3. Verifikasi bahwa item baru dibuat dengan detail penyewa.	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi pembuatan tumpukan untuk penyewa baru.	Verifikasi bahwa tumpukan baru berhasil dibuat dan disediakan dengan infrastru ktur untuk penyewa yang baru dibuat sesuai dengan template. CloudFormation 1. Buka CloudFormation konsol. 2. Di navigasi kiri, pilih Tumpukan, dan verifikasi bahwa tumpukan dengan nama penyewa berhasil dibuat. 3. Pilih tumpukan penyewa yang baru dibuat, lalu pilih tab Resources. Perhatika n sumber daya alarm dan sumber daya Amazon SQS. 4. Buka terminal baru dengan kredensi AWS yang dikonfigurasi, dan arahkan ke Wilayah yang benar. Untuk menaikkan alarm pengujian, masukkan kode berikut, ganti `<alarm resource name>` dengan nama sumber daya alarm yang tercantum pada langkah 3. `aws cloudwatch set-alarm-state --alarm-name <alarm resource name> --state-value`	Pengembang aplikasi, administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`ALARM --state-reason 'Test setup'`
	
	Contoh berikut menunjukkan kode dengan nama sumber daya alarm.
	
	`aws cloudwatch set-alarm-state --alarm-name tenantcluster-tenant123-alarm --state-value ALARM --state-reason 'Test setup'`
	
	5. Buka konsol dan arahkan ke konsol Amazon SQS. Pilih nama sumber daya Amazon SQS yang diidentifikasi pada langkah 3. Ikuti petunjuk dokumentasi AWS untuk menerima dan menghapus pesan pengujian dari alarm yang dimunculkan pada langkah 4.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hapus tumpukan penyewa.	Untuk menghapus tumpukan penyewa, kirim permintaan curl berikut.
	`curl -X DELETE <TenantOnboardingAPIEndpoint* from CDK Output>tenant/<Tenant Name from previous step>`
	Ubah pemegang tempat `<TenantOnboardingAPIEndpoint* from CDK Output>` ke nilai aktual dari AWS CDK, dan ubah `<Tenant Name from previous step>` ke nilai aktual dari langkah pembuatan penyewa sebelumnya, seperti yang ditunjukkan pada contoh berikut.
	`curl -X DELETE https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/tenant/Tenant123`
	Contoh berikut menunjukkan output.
	`{"message": "Tenant destroyed - 5/4/2022 7:14:48 AM"}`
		Pengembang aplikasi, AWS DevOps, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi penghapusan tumpukan untuk penyewa yang ada.	Untuk memverifikasi bahwa tumpukan penyewa yang ada telah dihapus, lakukan langkah-langkah berikut: 1. Buka konsol dan navigasikan ke CloudFormation konsol. 2. Di navigasi kiri, verifikasi bahwa tumpukan yang ada dengan nama penyewa tidak lagi ada di konsol (jika CloudFormation konsol diatur untuk hanya menampilkan tumpukan Aktif) atau sedang dalam proses dihapus. Jika tumpukan tidak lagi ada di CloudFormation konsol, gunakan daftar tarik-turun untuk mengubah pengaturan konsol dari Aktif ke Dihapus untuk melihat tumpukan yang dihapus dan verifikasi bahwa tumpukan berhasil dihapus.	Pengembang aplikasi, administrator AWS, AWS DevOps

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hancurkan lingkungan.	Sebelum tumpukan dibersihkan, pastikan hal berikut:	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Semua catatan di DynamoDB dihapus baik melalui operasi penghapus an penyewa sebelumny a atau melalui konsol DynamoDB atau API. Setiap penghapusan catatan penyewa akan memulai pembersihan mitra AWS-nya. CloudFormation • Semua CloudFormation tumpukan AWS berbasis penyewa dibersihkan (jika logika pembersihan pemicu DynamoDB gagal) di konsol AWS. CloudFormation Setelah pengujian selesai, AWS CDK dapat digunakan untuk menghancurkan semua tumpukan dan sumber daya terkait dengan menjalankan perintah berikut. `cdk destroy --all;` Jika Anda membuat profil AWS untuk kredensialnya, gunakan profil tersebut. Konfirmasikan prompt penghapusan tumpukan untuk menghapus tumpukan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan CloudWatch Log Amazon.	Proses penghapusan tumpukan tidak akan membersihkan CloudWatch Log (grup log dan log) yang dihasilkan oleh tumpukan. Bersihkan CloudWatch sumber daya secara manual dengan menggunakan CloudWatch konsol atau API.	Pengembang aplikasi, AWS DevOps, administrator AWS

Sumber daya terkait

- Workshop AWS CDK .NET
- Bekerja dengan AWS CDK di C #
- CDK .NET Referensi

Informasi tambahan

Tumpukan teknologi bidang kontrol

Kode CDK yang ditulis dalam.NET digunakan untuk menyediakan infrastruktur bidang kontrol, yang terdiri dari sumber daya berikut:

1. API Gateway

   Berfungsi sebagai titik masuk REST API untuk tumpukan bidang kontrol.

2. Fungsi Lambda on-boarding penyewa

   Fungsi Lambda ini diprakarsai oleh API Gateway menggunakan metode m.

   Permintaan API metode POST menghasilkan (`tenant name`,`tenant description`) yang dimasukkan ke dalam tabel DynamoDB`tenant Onboarding`.

   Dalam contoh kode ini, nama penyewa juga digunakan sebagai bagian dari nama tumpukan penyewa dan nama sumber daya dalam tumpukan itu. Ini untuk membuat sumber daya ini lebih

mudah diidentifikasi. Nama penyewa ini harus unik di seluruh pengaturan untuk menghindari konflik atau kesalahan. Pengaturan validasi masukan terperinci dijelaskan dalam dokumentasi peran IAM dan bagian Batasan.

Proses persistensi ke tabel DynamoDB akan berhasil hanya jika nama penyewa tidak digunakan dalam catatan lain dalam tabel.

Nama penyewa dalam hal ini adalah kunci partisi untuk tabel ini, karena hanya kunci partisi yang dapat digunakan sebagai ekspresi `PutItem` kondisi.

Jika nama penyewa tidak pernah direkam sebelumnya, catatan akan berhasil disimpan ke dalam tabel.

Namun, jika nama penyewa sudah digunakan oleh catatan yang ada dalam tabel, operasi akan gagal dan memulai pengecualian DynamoDB. `ConditionalCheckFailedException` Pengecualian akan digunakan untuk mengembalikan pesan kegagalan (`HTTP BadRequest`) yang menunjukkan bahwa nama penyewa sudah ada.

Permintaan `DELETE` metode API akan menghapus catatan untuk nama penyewa tertentu dari tabel `Tenant Onboardin` g.

Penghapusan catatan DynamoDB dalam contoh ini akan berhasil bahkan jika catatan tidak ada.

Jika catatan target ada dan dihapus, itu akan membuat catatan aliran DynamoDB. Jika tidak, tidak ada catatan hilir yang akan dibuat.

3. Penyewa on-boarding DynamoDB, dengan Amazon DynamoDB Streams diaktifkan

   Ini mencatat informasi metadata penyewa, dan penyimpanan atau penghapusan catatan apa pun akan mengirim aliran hilir ke fungsi Lambda. `Tenant Infrastructure`

4. Infrastruktur penyewa fungsi Lambda

   Fungsi Lambda ini diprakarsai oleh catatan aliran DynamoDB dari langkah sebelumnya. Jika record adalah untuk suatu `INSERT` peristiwa, ia memanggil AWS CloudFormation untuk membuat infrastruktur penyewa baru dengan CloudFormation template yang disimpan dalam bucket S3. Jika catatan untuk`REMOVE`, itu memulai penghapusan tumpukan yang ada berdasarkan bidang rekaman aliran. `Tenant Name`

5. Ember S3

   Ini untuk menyimpan CloudFormation template.

6. Peran IAM untuk setiap fungsi Lambda dan peran layanan untuk CloudFormation

   Setiap fungsi Lambda memiliki peran IAM yang unik dengan izin hak istimewa paling sedikit untuk mencapai tugasnya. Misalnya, fungsi `Tenant On-boarding` Lambda memiliki read/write akses ke DynamoDB, dan fungsi `Tenant Infrastructure` Lambda hanya dapat membaca aliran DynamoDB.

   Peran CloudFormation layanan khusus dibuat untuk penyediaan tumpukan penyewa. Peran layanan ini berisi izin tambahan untuk penyediaan CloudFormation tumpukan (misalnya, kunci AWS KMS). Ini membagi peran antara Lambda CloudFormation dan untuk menghindari semua izin pada satu peran (peran Lambda Infrastruktur).

   Izin yang memungkinkan tindakan kuat (seperti membuat dan menghapus CloudFormation tumpukan) dikunci dan hanya diizinkan pada sumber daya yang dimulai. `tenantcluster-` Pengecualiannya adalah AWS KMS, karena konvensi penamaan sumber dayanya. Nama penyewa yang dicerna dari API akan ditambahkan `tenantcluster-` bersama dengan pemeriksaan validasi lainnya (alfanumerik dengan tanda hubung saja, dan dibatasi hingga kurang dari 30 karakter agar sesuai dengan sebagian besar penamaan sumber daya AWS). Ini memastikan bahwa nama penyewa tidak akan secara tidak sengaja mengakibatkan gangguan tumpukan infrastruktur inti atau sumber daya.

Tumpukan teknologi penyewa

CloudFormation Template disimpan dalam ember S3. Template menyediakan kunci AWS KMS khusus penyewa, CloudWatch alarm, topik SNS, antrian SQS, dan kebijakan SQS.

Kunci AWS KMS digunakan untuk enkripsi data oleh Amazon SNS dan Amazon SQS untuk pesan mereka. Praktik keamanan untuk AwsSolutions- SNS2 dan AwsSolutions - SQS2 merekomendasikan agar Anda mengatur Amazon SNS dan Amazon SQS dengan enkripsi. Namun, CloudWatch alarm tidak berfungsi dengan Amazon SNS saat menggunakan kunci yang dikelola AWS, jadi Anda harus menggunakan kunci yang dikelola pelanggan dalam kasus ini. Untuk informasi selengkapnya, lihat Pusat Pengetahuan AWS.

Kebijakan SQS digunakan pada antrean Amazon SQS untuk memungkinkan topik SNS yang dibuat mengirimkan pesan ke antrian. Tanpa kebijakan SQS, akses akan ditolak. Untuk informasi selengkapnya, lihat dokumentasi Amazon SNS.

Menguraikan monolit menjadi layanan mikro dengan menggunakan CQRS dan sumber acara

Rodolfo Jr. Cerrada, Dmitry Gulin, dan Tabby Ward, Amazon Web Services

Ringkasan

Pola ini menggabungkan dua pola, menggunakan pola pemisahan tanggung jawab permintaan perintah (CQRS) dan pola sumber peristiwa. Pola CQRS memisahkan tanggung jawab model perintah dan kueri. Pola sumber acara memanfaatkan komunikasi berbasis peristiwa asinkron untuk meningkatkan pengalaman pengguna secara keseluruhan.

Anda dapat menggunakan layanan CQRS dan Amazon Web Services (AWS) untuk memelihara dan menskalakan setiap model data secara independen sambil memfaktorkan ulang aplikasi monolit Anda ke dalam arsitektur layanan mikro. Kemudian Anda dapat menggunakan pola sumber acara untuk menyinkronkan data dari database perintah ke database kueri.

Pola ini menggunakan kode contoh yang menyertakan file solusi (*.sln) yang dapat Anda buka menggunakan versi terbaru Visual Studio. Contoh ini berisi kode Reward API untuk menampilkan cara kerja CQRS dan sumber acara di aplikasi AWS tanpa server dan tradisional atau lokal.

Untuk mempelajari lebih lanjut tentang CQRS dan sumber acara, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Amazon CloudWatch
- Tabel Amazon DynamoDB
- Amazon DynamoDB Streams
- Kunci akses AWS Identity and Access Management (IAM) dan kunci rahasia; untuk informasi selengkapnya, lihat video di bagian Sumber daya terkait
- AWS Lambda
- Keakraban dengan Visual Studio
- Keakraban dengan AWS Toolkit for Visual Studio; untuk informasi selengkapnya, lihat video demo AWS Toolkit for Visual Studio di bagian Sumber daya terkait

Versi produk

- Edisi Komunitas Visual Studio 2019.
- AWS Toolkit for Visual Studio 2019.
- .NET Inti 3.1. Komponen ini merupakan opsi dalam instalasi Visual Studio. Untuk menyertakan .NET Core selama instalasi, pilih NET Core cross-platform development.

Batasan

- Kode contoh untuk aplikasi lokal tradisional (ASP.NET Core Web API dan objek akses data) tidak disertakan dengan database. Namun, ia datang dengan objek `CustomerData` dalam memori, yang bertindak sebagai database tiruan. Kode yang diberikan cukup bagi Anda untuk menguji polanya.

Arsitektur

Tumpukan teknologi sumber

- Proyek API Web Inti ASP.NET
- Server Web IIS
- Objek akses data
- Model CRUD

Arsitektur sumber

Dalam arsitektur sumber, model CRUD berisi antarmuka perintah dan kueri dalam satu aplikasi. Misalnya kode, lihat `CustomerDA0.cs` (terlampir).

Tumpukan teknologi target

- Amazon DynamoDB
- Amazon DynamoDB Streams
- AWS Lambda
- (Opsional) Amazon API Gateway
- (Opsional) Layanan Pemberitahuan Sederhana Amazon (Amazon SNS)

Arsitektur target

Dalam arsitektur target, antarmuka perintah dan kueri dipisahkan. Arsitektur yang ditunjukkan dalam diagram berikut dapat diperluas dengan API Gateway dan Amazon SNS. Untuk informasi selengkapnya, lihat bagian Informasi tambahan.

1. Fungsi Command Lambda melakukan operasi tulis, seperti membuat, memperbarui, atau menghapus, pada database.
2. Fungsi Query Lambda melakukan operasi baca, seperti dapatkan atau pilih, pada database.
3. Fungsi Lambda ini memproses aliran DynamoDB dari database Command dan memperbarui database Query untuk perubahan.

Alat

Alat

- Amazon DynamoDB - Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi dengan skalabilitas yang mulus.
- Amazon DynamoDB Streams - DynamoDB Streams menangkap urutan urutan waktu modifikasi tingkat item dalam tabel DynamoDB apa pun. Kemudian menyimpan informasi ini dalam log hingga 24 jam. Enkripsi saat istirahat mengenkripsi data dalam DynamoDB streams.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.
- AWS Management Console — AWS Management Console adalah aplikasi web yang terdiri dari koleksi luas konsol layanan untuk mengelola layanan AWS.
- Visual Studio 2019 Community Edition — Visual Studio 2019 adalah lingkungan pengembangan terintegrasi (IDE). Edisi Komunitas gratis untuk kontributor sumber terbuka. Dalam pola ini, Anda akan menggunakan Visual Studio 2019 Community Edition untuk membuka, mengkompilasi, dan menjalankan kode contoh. Untuk melihat saja, Anda dapat menggunakan editor teks atau Kode Visual Studio.

- AWS Toolkit for Visual Studio — AWS Toolkit for Visual Studio adalah plugin untuk Visual Studio IDE. AWS Toolkit for Visual Studio memudahkan Anda mengembangkan, men-debug, dan menerapkan aplikasi.NET yang menggunakan layanan AWS.

Kode

Kode contoh terlampir. Untuk petunjuk tentang penerapan kode contoh, lihat bagian Epik.

Epik

Buka dan bangun solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka solusinya.	1. Unduh contoh kode sumber (`CQRS-ES Code.zip`) dari bagian Lampiran, dan ekstrak file. 2. Di Visual Studio IDE, pilih File, Open, Project Solution, dan navigasikan ke folder tempat Anda mengekstrak kode sumber. 3. Pilih AWS.apg.cqrses.sln , lalu pilih Buka. Seluruh solusi dimuat ke Visual Studio.	Pengembang aplikasi
Bangun solusinya.	Buka menu konteks (klik kanan) untuk solusinya, lalu pilih Build Solution. Ini akan membangun dan mengkompilasi semua proyek dalam solusi. Itu harus berhasil dikompilasi.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Visual Studio Solution Explorer harus menunjukkan struktur direktori. • `CQRS On-Premises Code Sample`berisi contoh penggunaan CQRS di tempat. • `CQRS AWS Serverless` berisi semua kode contoh CQRS dan sumber peristiwa menggunakan layanan tanpa server AWS.	

Membangun tabel DynamoDB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan kredensyal.	Jika Anda belum memiliki kunci akses, lihat video di bagian Sumber daya terkait. 1. Di Solution Explorer, perluas CQRS AWS Serverless, lalu perluas folder solusi Build. 2. Perluas proyek AWS.apg.cqrses.Build dan lihat file. `Program.cs` 3. Gulir ke atas `Program.cs` dan cari`Program()` . 4. Ganti `YOUR ACCESS KEY` dengan kunci akses akun Anda, dan ganti `YOUR`	Pengembang aplikasi, Insinyur data, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	SECRET KEY dengan kunci rahasia akun Anda. Perhatikan bahwa dalam lingkungan produksi, Anda tidak akan melakukan hardcode kunci Anda. Sebagai gantinya, Anda dapat menggunakan AWS Secrets Manager untuk menyimpan dan mengambil kredensialnya.	
Bangun proyek.	Untuk membangun proyek, buka menu konteks (klik kanan) untuk proyek AWS.apg.cqrses.Build, lalu pilih Build.	Pengembang aplikasi, Insinyur data, DBA
Bangun dan isi tabel.	Untuk membuat tabel dan mengisinya dengan data benih, buka menu konteks (klik kanan) untuk proyek AWS.apg.cqrses.Build, lalu pilih Debug, Start New Instance.	Pengembang aplikasi, Insinyur data, DBA
Verifikasi konstruksi tabel dan data.	Untuk memverifikasi, navigasikan ke AWS Explorer, dan perluas Amazon DynamoDB. Ini harus menampilkan tabel. Buka setiap tabel untuk menampilk an contoh data.	Pengembang aplikasi, Insinyur data, DBA

Jalankan tes lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membangun proyek CQRS.	1. Buka solusinya, dan arahkan ke folder Services/CQRS/Tests solusi CQRS AWS. 2. Di AWS.APG.CQRSES.CQRSLambda.Tests proyek, buka BaseFunctionTest.cs, dan ganti AccessKeydan SecretKey dengan kunci IAM yang Anda buat. 3. Simpan perubahan. 4. Untuk mengkompilasi dan membangun proyek pengujian, buka menu konteks (klik kanan) untuk proyek, lalu pilih Build.	Pengembang aplikasi, Insinyur uji
Bangun proyek sumber acara.	1. Arahkan ke folder solusi Services/Event Sumber/Pengujian AWS CQRS. 2. Di AWS.APG.CQRSES.EventSourceLambda.Tests proyek, buka BaseFunctionTest.cs, dan ganti AccessKeydan SecretKey dengan kunci IAM yang Anda buat. 3. Simpan perubahan. 4. Untuk mengkompilasi dan membangun proyek pengujian, buka menu	Pengembang aplikasi, Insinyur uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	konteks (klik kanan) untuk proyek, lalu pilih Build.	
Jalankan tes.	Untuk menjalankan semua pengujian, pilih View, Test Explorer, lalu pilih Run All Tests In View. Semua tes harus lulus, yang ditunjukkan oleh ikon tanda centang hijau.	Pengembang aplikasi, Insinyur uji

Publikasikan fungsi Lambda CQRS ke AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Publikasikan fungsi Lambda pertama.	1. Di Solution Explorer, buka menu konteks (klik kanan) untuk AWS.APG.CQRSES.CommandCreateLambda project, lalu pilih Publish to AWS Lambda. 2. Pilih profil yang ingin Anda gunakan dan Wilayah AWS tempat Anda ingin menerapkan fungsi Lambda, dan nama fungsi. 3. Untuk bidang yang tersisa, pertahankan nilai default, dan pilih Berikutnya. 4. Dalam daftar dropdown Nama Peran, pilih. AWSLambdaFullAccess 5. Untuk memberikan kunci akun Anda, pilih Tambah, dan masukkan `AcessKey`	Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sebagai variabel dan kunci akses Anda sebagai nilai. Kemudian pilih Tambah lagi, masukkan `SecretKey` sebagai variabel dan kunci rahasia Anda sebagai nilainya. 6. Untuk bidang yang tersisa, pertahankan nilai default, dan pilih Unggah. Setelah fungsi uji Lambda diunggah, itu muncul di Visual Studio secara otomatis. 7. Ulangi langkah 1-6 untuk proyek-proyek berikut: • AWS.APG.CQRSES.CommandDeleteLambda • AWS.APG.CQRSES.CommandUpdateLambda • AWS.APG.CQRSES.CommandAddRewardLambda • AWS.APG.CQRSES.CommandRedeemRewardLambda • AWS.APG.CQRSES.QueryCustomerListLambda • AWS.APG.CQRSES.QueryRewqardLambda	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi fungsi upload.	(Opsional) Anda dapat memverifikasi bahwa fungsi berhasil dimuat dengan menavigasi ke AWS Explorer dan memperluas AWS Lambda. Untuk membuka jendela pengujian, pilih fungsi Lambda (klik dua kali).	Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji fungsi Lambda.	1. Masukkan data permintaan, atau salin contoh data permintaan dari Data uji di bagian Informasi tambahan. Pastikan Anda memilih data yang untuk fungsi yang Anda uji. 2. Untuk menjalankan pengujian, pilih Invoke. Respons dan kesalahan apa pun ditampilkan di kotak teks Respons, dan log ditampilkan di kotak teks Log atau di CloudWatch Log. 3. Untuk memverifikasi data, di AWS Explorer, pilih tabel DynamoDB (klik dua kali). Semua proyek Lambda CQRS ditemukan di bawah folder dan solusi. `CQRS AWS Serverless\CQRS\Command Microservice CQRS AWS Serverless\CQRS\Command Microservice` Untuk direktori solusi dan proyek, lihat Direktori kode sumber di bagian Informasi tambahan.	Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Publikasikan fungsi yang tersisa.	Ulangi langkah sebelumnya untuk proyek-proyek berikut: • AWS.APG.CQRSES. CommandDeleteLambda • AWS.APG.CQRSES. CommandUpdateLambda • AWS.APG.CQRSES. CommandAddRewardLa mbda • AWS.APG.CQRSES. CommandRedeemRewar dLambda • AWS.APG.CQRSES. QueryCustomerListLambda • AWS.APG.CQRSES. QueryRewqardLambda	Pengembang aplikasi, DevOps insinyur

Siapkan fungsi Lambda sebagai pendengar acara

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Publikasikan penangan acara Pelanggan dan Hadiah Lambda.	Untuk mempublikasikan setiap event handler, ikuti langkah-l angkah dalam epik sebelumny a. Proyek-proyek berada di bawah folder `CQRS AWS Serverless\Event Source\Reward Event` solusi `CQRS AWS Serverles s\Event Source\Cu stomer Event` dan. Untuk	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi selengkapnya, lihat Direktori kode sumber di bagian Informasi tambahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan pendengar acara Lambda sumber acara.	1. Masuk ke AWS Management Console menggunakan akun yang sama yang Anda gunakan saat memublikasikan project Lambda. 2. Untuk Wilayah, pilih US East 1 atau Wilayah tempat Anda menerapkan fungsi Lambda di epik sebelumnya. 3. Arahkan ke layanan Lambda. 4. Pilih fungsi `EventSourceCustomer` Lambda. 5. Pilih Tambahkan Pemicu. 6. Dalam daftar dropdown konfigurasi Trigger, pilih DynamoDB. 7. Dalam daftar dropdown tabel DynamoDB, pilih. cqrses-customer-cmd 8. Dalam daftar dropdown posisi awal, pilih Trim horizon dari. Trim horizon berarti bahwa pemicu DynamoDB akan mulai membaca pada catatan aliran terakhir (tidak dipangkas), yang merupakan catatan tertua di pecahan.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	9. Pilih kotak centang Aktifkan pemicu. 10. Untuk bidang yang tersisa, pertahankan nilai default, dan pilih Tambah. Setelah pendengar berhasil dilampirkan ke tabel DynamoDB, itu akan ditampilkan di halaman desainer Lambda.	
Publikasikan dan lampirkan fungsi EventSourceReward Lambda.	Untuk mempublikasikan dan melampirkan fungsi **`EventSourceReward`** Lambda, ulangi langkah-langkah dalam dua cerita sebelumnya, pilih cqrses-reward-cmddari daftar dropdown tabel DynamoDB.	Pengembang aplikasi

Menguji dan memvalidasi aliran DynamoDB dan pemicu Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aliran dan pemicu Lambda.	1. Di Visual Studio, navigasikan ke AWS Explorer. 2. Perluas AWS Lambda, dan pilih CommandReedeemRewardfungsinya (klik dua kali). Di jendela fungsi yang terbuka, Anda dapat menguji fungsinya.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di kotak teks Permintaan, masukkan data permintaan dalam format JavaScript Object Notation (JSON). Untuk permintaan contoh, lihat Menguji data di bagian Informasi tambahan. 4. Pilih Panggil.	
Validasi, menggunakan tabel query reward DynamoddB.	1. Buka cqrses-reward-querymeja. 2. Periksa poin pelanggan yang menukarkan hadiah. Poin yang ditukarkan harus dikurangi dari total poin agregat pelanggan.	Pengembang aplikasi
Validasi, menggunakan CloudWatch Log.	1. Arahkan ke CloudWatch dan pilih Grup log. 2. Grup/aws/lambda/EventSourceRewardlog berisi log untuk `EventSourceReward` pemicu. Semua panggilan Lambda dicatat, termasuk pesan yang Anda tempatkan `context.Logger.LogLine` dan `Console.Writeline` dalam kode Lambda.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi EventSourceCustomer pemicunya.	Untuk memvalidasi `EventSourceCustomer` pemicu, ulangi langkah-langkah dalam epik ini, menggunakan tabel dan CloudWatch log pelanggan masing-masing `EventSourceCustomer` pemicu.	Pengembang aplikasi

Sumber daya terkait

Referensi

- Unduhan Edisi Komunitas Visual Studio 2019
- Unduhan AWS Toolkit for Visual Studio
- Panduan Pengguna AWS Toolkit for Visual Studio
- Tanpa server di AWS
- Kasus Penggunaan DynamoDB dan Pola Desain
- Martin Fowler CQRS
- Sumber Acara Martin Fowler

Video

- AWS Toolkit untuk demo AWS Toolkit for Visual Studio
- Bagaimana cara membuat ID kunci akses untuk pengguna IAM baru?

Informasi tambahan

CQRS dan sumber acara

CQRS

Pola CQRS memisahkan model operasi konseptual tunggal, seperti objek akses data tunggal CRUD (create, read, update, delete) model, ke dalam model operasi perintah dan query. Model perintah

mengacu pada operasi apa pun, seperti membuat, memperbarui, atau menghapus, yang mengubah status. Model query mengacu pada setiap operasi yang mengembalikan nilai.

1. Model CRUD Pelanggan mencakup antarmuka berikut:
   - `Create Customer()`
   - `UpdateCustomer()`
   - `DeleteCustomer()`
   - `AddPoints()`
   - `RedeemPoints()`
   - `GetVIPCustomers()`
   - `GetCustomerList()`
   - `GetCustomerPoints()`

Karena kebutuhan Anda menjadi lebih kompleks, Anda dapat beralih dari pendekatan model tunggal ini. CQRS menggunakan model perintah dan model kueri untuk memisahkan tanggung jawab untuk menulis dan membaca data. Dengan begitu, data dapat dikelola dan dikelola secara independen. Dengan pemisahan tanggung jawab yang jelas, peningkatan untuk setiap model tidak memengaruhi yang lain. Pemisahan ini meningkatkan pemeliharaan dan kinerja, dan mengurangi kompleksitas aplikasi saat tumbuh.

1. Antarmuka dalam model Customer Command:
   - `Create Customer()`
   - `UpdateCustomer()`
   - `DeleteCustomer()`
   - `AddPoints()`
   - `RedeemPoints()`
2. Antarmuka dalam model Customer Query:
   - `GetVIPCustomers()`
   - `GetCustomerList()`
   - `GetCustomerPoints()`
   - `GetMonthlyStatement()`

Misalnya kode, lihat Direktori kode sumber.

Pola CQRS kemudian memisahkan database. Decoupling ini mengarah pada independensi total setiap layanan, yang merupakan bahan utama arsitektur microservice.

Dengan menggunakan CQRS di AWS Cloud, Anda dapat mengoptimalkan setiap layanan lebih lanjut. Misalnya, Anda dapat mengatur pengaturan komputasi yang berbeda atau memilih antara layanan mikro tanpa server atau berbasis kontainer. Anda dapat mengganti caching lokal dengan Amazon. ElastiCache Jika memiliki publish/subscribe pesan lokal, Anda dapat menggantinya dengan Amazon Simple Notification Service (Amazon SNS). Selain itu, Anda dapat memanfaatkan pay-as-you-go harga dan beragam layanan AWS yang Anda bayar hanya untuk apa yang Anda gunakan.

CQRS mencakup manfaat berikut:

- Penskalaan independen — Setiap model dapat memiliki strategi penskalaan yang disesuaikan untuk memenuhi persyaratan dan permintaan layanan. Mirip dengan aplikasi berkinerja tinggi, memisahkan baca dan tulis memungkinkan model untuk menskalakan secara independen untuk memenuhi setiap permintaan. Anda juga dapat menambah atau mengurangi sumber daya komputasi untuk mengatasi permintaan skalabilitas dari satu model tanpa mempengaruhi yang lain.
- Pemeliharaan independen - Pemisahan model kueri dan perintah meningkatkan pemeliharaan model. Anda dapat membuat perubahan kode dan penyempurnaan ke satu model tanpa mempengaruhi yang lain.
- Keamanan — Lebih mudah menerapkan izin dan kebijakan untuk memisahkan model untuk dibaca dan ditulis.
- Bacaan yang dioptimalkan - Anda dapat menentukan skema yang dioptimalkan untuk kueri. Misalnya, Anda dapat menentukan skema untuk data agregat dan skema terpisah untuk tabel fakta.
- Integrasi — CQRS cocok dengan model pemrograman berbasis acara.
- Kompleksitas terkelola - Pemisahan menjadi model kueri dan perintah cocok untuk domain yang kompleks.

Saat menggunakan CQRS, ingatlah peringatan berikut:

- Pola CQRS hanya berlaku untuk bagian tertentu dari aplikasi dan bukan seluruh aplikasi. Jika diterapkan pada domain yang tidak sesuai dengan pola, dapat mengurangi produktivitas, meningkatkan risiko, dan menimbulkan kompleksitas.

- Pola ini bekerja paling baik untuk model yang sering digunakan yang memiliki operasi baca dan tulis ketidakseimbangan.
- Untuk aplikasi read-heavy, seperti laporan besar yang membutuhkan waktu untuk diproses, CQRS memberi Anda opsi untuk memilih database yang tepat dan membuat skema untuk menyimpan data agregat Anda. Ini meningkatkan waktu respons membaca dan melihat laporan dengan memproses data laporan hanya satu kali dan membuangnya dalam tabel agregat.
- Untuk aplikasi berat tulis, Anda dapat mengonfigurasi database untuk operasi penulisan dan memungkinkan layanan mikro perintah untuk menskalakan secara independen ketika permintaan untuk menulis meningkat. Sebagai contoh, lihat `AWS.APG.CQRSES.CommandRedeemRewardLambda` dan `AWS.APG.CQRSES.CommandAddRewardLambda` layanan mikro.

Sumber acara

Langkah selanjutnya adalah menggunakan sumber acara untuk menyinkronkan database kueri saat perintah dijalankan. Misalnya, pertimbangkan peristiwa-peristiwa berikut:

- Poin hadiah pelanggan ditambahkan yang mengharuskan total atau poin hadiah gabungan pelanggan dalam database kueri diperbarui.
- Nama belakang pelanggan diperbarui dalam database perintah, yang mengharuskan informasi pelanggan pengganti dalam database kueri diperbarui.

Dalam model CRUD tradisional, Anda memastikan konsistensi data dengan mengunci data hingga menyelesaikan transaksi. Dalam sumber acara, data disinkronkan melalui penerbitan serangkaian acara yang akan dikonsumsi oleh pelanggan untuk memperbarui data masing-masing.

Pola sumber peristiwa memastikan dan mencatat serangkaian tindakan lengkap yang diambil pada data dan menerbitkannya melalui serangkaian peristiwa. Peristiwa ini mewakili serangkaian perubahan pada data yang harus diproses oleh pelanggan acara tersebut agar catatan mereka diperbarui. Peristiwa ini dikonsumsi oleh pelanggan, menyinkronkan data pada database pelanggan. Dalam hal ini, itulah database query.

Diagram berikut menunjukkan sumber peristiwa yang digunakan dengan CQRS di AWS.

1. Fungsi Command Lambda melakukan operasi tulis, seperti membuat, memperbarui, atau menghapus, pada database.

2. Fungsi Query Lambda melakukan operasi baca, seperti dapatkan atau pilih, pada database.
3. Fungsi Lambda ini memproses aliran DynamoDB dari database Command dan memperbarui database Query untuk perubahan. Anda juga dapat menggunakan fungsi ini juga untuk mempublikasikan pesan ke Amazon SNS sehingga pelanggannya dapat memproses data.
4. (Opsional) Pelanggan acara Lambda memproses pesan yang diterbitkan oleh Amazon SNS dan memperbarui database Kueri.
5. (Opsional) Amazon SNS mengirimkan pemberitahuan email tentang operasi tulis.

Di AWS, database kueri dapat disinkronkan oleh DynamoDB Streams. DynamoDB menangkap urutan modifikasi tingkat item yang diurutkan waktu dalam tabel DynamoBDB dalam waktu nyaris nyata dan tahan lama menyimpan informasi dalam waktu 24 jam.

Mengaktifkan DynamoDB Streams memungkinkan database untuk mempublikasikan urutan peristiwa yang memungkinkan pola sumber peristiwa. Pola sumber acara menambahkan pelanggan acara. Aplikasi pelanggan acara mengkonsumsi acara dan memprosesnya tergantung pada tanggung jawab pelanggan. Pada diagram sebelumnya, pelanggan acara mendorong perubahan ke database Query DynamoDB untuk menjaga data tetap disinkronkan. Penggunaan Amazon SNS, broker pesan, dan aplikasi pelanggan acara membuat arsitektur dipisahkan.

Sumber acara mencakup manfaat berikut:

- Konsistensi untuk data transaksional
- Jejak audit yang andal dan riwayat tindakan, yang dapat digunakan untuk memantau tindakan yang diambil dalam data
- Memungkinkan aplikasi terdistribusi seperti microservices untuk menyinkronkan data mereka di seluruh lingkungan
- Publikasi acara yang andal setiap kali negara berubah
- Rekonstruksi atau pemutaran ulang keadaan masa lalu
- Entitas yang digabungkan secara longgar yang bertukar peristiwa untuk migrasi dari aplikasi monolitik ke layanan mikro
- Pengurangan konflik yang disebabkan oleh pembaruan bersamaan; sumber acara menghindari persyaratan untuk memperbarui objek secara langsung di penyimpanan data
- Fleksibilitas dan ekstensibilitas dari memisahkan tugas dan acara
- Pembaruan sistem eksternal
- Manajemen beberapa tugas dalam satu acara

Saat menggunakan sumber acara, ingatlah peringatan berikut:

- Karena ada beberapa keterlambatan dalam memperbarui data antara database pelanggan sumber, satu-satunya cara untuk membatalkan perubahan adalah dengan menambahkan acara kompensasi ke toko acara.
- Menerapkan sumber acara memiliki kurva pembelajaran karena gaya pemrogramannya yang berbeda.

Data uji

Gunakan data pengujian berikut untuk menguji fungsi Lambda setelah penerapan berhasil.

CommandCreate Pelanggan

```
{  "Id":1501,  "Firstname":"John",  "Lastname":"Done",  "CompanyName":"AnyCompany",
 "Address": "USA",  "VIP":true }
```

CommandUpdate Pelanggan

```
{  "Id":1501,  "Firstname":"John",  "Lastname":"Doe",  "CompanyName":"Example Corp.",
 "Address": "Seattle, USA",  "VIP":true }
```

CommandDelete Pelanggan

Masukkan ID pelanggan sebagai data permintaan. Misalnya, jika ID pelanggan adalah 151, masukkan 151 sebagai data permintaan.

```
151
```

QueryCustomerList

Ini kosong. Ketika dipanggil, itu akan mengembalikan semua pelanggan.

CommandAddReward

Ini akan menambah 40 poin untuk pelanggan dengan ID 1 (Richard).

```
{
  "Id":10101,
```

```
  "CustomerId":1,
  "Points":40
}
```

CommandRedeemReward

Ini akan mengurangi 15 poin untuk pelanggan dengan ID 1 (Richard).

```
{
  "Id":10110,
  "CustomerId":1,
  "Points":15
}
```

QueryReward

Masukkan ID pelanggan. Misalnya, masukkan 1 untuk Richard, 2 untuk Arnav, dan 3 untuk Shirley.

```
2
```

Direktori kode sumber

Gunakan tabel berikut sebagai panduan untuk struktur direktori solusi Visual Studio.

Direktori solusi Contoh Kode Lokal CQRS

Model CRUD pelanggan

Contoh Kode Lokal CQRS\ Model CRUD\ Proyek AWS.APG.CQRSES.DAL

Versi CQRS dari model CRUD Pelanggan

- Perintah pelanggan: `CQRS On-Premises Code Sample\CQRS Model\Command Microservice\AWS.APG.CQRSES.Command` proyek
- Permintaan pelanggan: `CQRS On-Premises Code Sample\CQRS Model\Query Microservice\AWS.APG.CQRSES.Query` proyek

Layanan mikro Command dan Query

Layanan mikro Command berada di bawah folder `CQRS On-Premises Code Sample\CQRS Model\Command Microservice` solusi:

- `AWS.APG.CQRSES.CommandMicroservice`Proyek API ASP.NET Core bertindak sebagai titik masuk di mana konsumen berinteraksi dengan layanan.
- `AWS.APG.CQRSES.Command`.NET Core adalah objek yang menampung objek dan antarmuka yang berhubungan dengan perintah.

Layanan mikro kueri ada di bawah folder `CQRS On-Premises Code Sample\CQRS Model \Query Microservice` solusi:

- `AWS.APG.CQRSES.QueryMicroservice`Proyek API ASP.NET Core bertindak sebagai titik masuk di mana konsumen berinteraksi dengan layanan.
- `AWS.APG.CQRSES.Query`.NET Core adalah objek yang menampung objek dan antarmuka terkait kueri.

Direktori solusi kode tanpa server CQRS AWS

Kode ini adalah versi AWS dari kode lokal yang menggunakan layanan tanpa server AWS.

Dalam C# .NET Core, setiap fungsi Lambda diwakili oleh satu proyek .NET Core. Dalam contoh kode pola ini, ada proyek terpisah untuk setiap antarmuka dalam model perintah dan kueri.

CQRS menggunakan layanan AWS

Anda dapat menemukan direktori solusi root untuk CQRS menggunakan layanan tanpa server AWS ada di folder. `CQRS AWS Serverless\CQRS` Contohnya mencakup dua model: Pelanggan dan Hadiah.

Perintah Lambda berfungsi untuk Pelanggan dan Hadiah berada di bawah `CQRS\Command Microservice\Customer` dan `CQRS\Command Microservice\Reward` folder. Mereka berisi proyek Lambda berikut:

- Perintah pelanggan:`CommandCreateLambda`,`CommandDeleteLambda`, dan `CommandUpdateLambda`
- Perintah hadiah: `CommandAddRewardLambda` dan `CommandRedeemRewardLambda`

Fungsi Lambda kueri untuk Pelanggan dan Hadiah ditemukan di bawah folder `CQRS\Query Microservice\Customer` dan`CQRS\QueryMicroservice\Reward`. Mereka berisi proyek `QueryCustomerListLambda` dan `QueryRewardLambda` Lambda.

Proyek uji CQRS

Proyek pengujian ada di bawah `CQRS\Tests` folder. Proyek ini berisi skrip pengujian untuk mengotomatiskan pengujian fungsi Lambda CQRS.

Sumber acara menggunakan layanan AWS

Penangan peristiwa Lambda berikut diprakarsai oleh aliran DynamoDB Pelanggan dan Reward untuk memproses dan menyinkronkan data dalam tabel kueri.

- Fungsi `EventSourceCustomer` Lambda dipetakan ke aliran `cqrses-customer-cmd` DynamoDB tabel Pelanggan ().
- Fungsi `EventSourceReward` Lambda dipetakan ke aliran `cqrses-reward-cmd` DynamoDB tabel Reward ().

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Lebih banyak pola

- Akses aplikasi kontainer secara pribadi di Amazon EKS menggunakan AWS PrivateLink dan Network Load Balancer
- Otomatis menambahkan atau memperbarui entri registri Windows menggunakan AWS Systems Manager
- Otomatiskan failover dan failback lintas wilayah dengan menggunakan DR Orchestrator Framework
- Secara otomatis membangun dan menyebarkan aplikasi Java ke Amazon EKS menggunakan pipeline CI/CD
- Membuat CI/CD pipeline dan kluster Amazon ECS secara otomatis untuk layanan mikro menggunakan AWS CDK
- Cadangkan dan arsipkan data mainframe ke Amazon S3 menggunakan BMC AMI Cloud Data
- Bangun PAC Server Perusahaan Fokus Mikro dengan Amazon EC2 Auto Scaling dan Systems Manager
- Membangun mesh data perusahaan dengan Amazon DataZone, AWS CDK, dan AWS CloudFormation
- Kontainerisasi beban kerja mainframe yang telah dimodernisasi oleh Blu Age
- Mengonversi dan membongkar data EBCDIC ke ASCII di AWS dengan menggunakan Python
- Mengkonversi file data mainframe dengan layout rekaman yang kompleks menggunakan Micro Focus
- Buat portal untuk frontend mikro dengan menggunakan AWS Amplify, Angular, dan Federasi Modul
- Menyebarkan wadah dengan menggunakan Elastic Beanstalk
- Meniru Oracle DR dengan menggunakan database global Aurora yang kompatibel dengan PostgreSQL
- Hasilkan wawasan data dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight
- Hasilkan wawasan z/OS data Db2 dengan menggunakan dan AWS Mainframe Modernization Amazon Q di Quick Sight
- Identifikasi gambar kontainer duplikat secara otomatis saat bermigrasi ke repositori Amazon ECR
- Menerapkan diagnostik Kubernetes yang didukung AI dan pemecahan masalah dengan integrasi K8SGPT dan Amazon Bedrock
- Menerapkan otentikasi berbasis Microsoft Entra ID dalam aplikasi mainframe modern AWS Blu Age

- Menerapkan versi API berbasis jalur dengan menggunakan domain khusus di Amazon API Gateway
- Bermigrasi secara bertahap dari Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL menggunakan Oracle SQL Developer dan AWS SCT
- Integrasikan Stonebranch Universal Controller dengan AWS Mainframe Modernisasi
- Kelola produk AWS Service Catalog di beberapa akun AWS dan Wilayah AWS
- Migrasi akun AWS anggota dari ke AWS Organizations AWS Control Tower
- Memigrasi dan mereplikasi file VSAM ke Amazon RDS atau Amazon MSK menggunakan Connect from Excently
- Bermigrasi dari SAP ASE ke Amazon RDS for SQL Server menggunakan AWS DMS
- Migrasikan tabel eksternal Oracle ke Amazon Aurora PostgreSQL yang kompatibel
- Modernisasi aplikasi CardDemo mainframe dengan menggunakan AWS Transform
- Memodernisasi dan menyebarkan aplikasi mainframe menggunakan dan Terraform AWS Transform
- Modernisasi beban kerja pencetakan batch mainframe dengan menggunakan Rocket Enterprise Server dan LRS AWS VPSX/LFI
- Modernisasi beban kerja pencetakan online mainframe di AWS dengan menggunakan Micro Focus Enterprise Server dan LRS VPSX/MFI
- Modernisasi manajemen output mainframe AWS dengan menggunakan Rocket Enterprise Server dan LRS X PageCenter
- Pindahkan file mainframe langsung ke Amazon S3 menggunakan Transfer Family
- Optimalkan penerapan tanpa server multi-akun dengan menggunakan alur kerja dan Tindakan AWS CDK GitHub
- Optimalkan kinerja aplikasi modern AWS Blu Age Anda
- Mengotomatiskan blue/green penyebaran database global Amazon Aurora dengan menggunakan prinsip IAc
- Replikasi database mainframe ke AWS dengan menggunakan Exactly Connect
- Jalankan tugas Amazon ECS di Amazon WorkSpaces dengan Amazon ECS Anywhere
- Kirim data telemetri dari AWS Lambda ke OpenSearch untuk analitik dan visualisasi waktu nyata
- Mengatur deteksi CloudFormation drift di organisasi multi-wilayah dan multi-akun
- Struktur proyek Python dalam arsitektur heksagonal menggunakan AWS Lambda
- Uji AWS infrastruktur dengan menggunakan LocalStack dan Tes Terraform

- Tingkatkan cluster SAP Pacemaker dari ke ENSA1 ENSA2
- Gunakan Amazon Q Developer sebagai asisten pengkodean untuk meningkatkan produktivitas Anda
- Validasi Account Factory untuk kode Terraform (AFT) secara lokal

Mainframe

Topik

- Akses Layanan AWS dari IBM z/OS dengan menginstal AWS CLI
- Cadangkan dan arsipkan data mainframe ke Amazon S3 menggunakan BMC AMI Cloud Data
- Membangun program COBOL Db2 dengan menggunakan dan AWS Mainframe Modernization AWS CodeBuild
- Bangun PAC Server Perusahaan Fokus Mikro dengan Amazon EC2 Auto Scaling dan Systems Manager
- Membangun penampil file mainframe tingkat lanjut di AWS Cloud
- Kontainerisasi beban kerja mainframe yang telah dimodernisasi oleh Blu Age
- Mengonversi dan membongkar data EBCDIC ke ASCII di AWS dengan menggunakan Python
- Konversi file mainframe dari format EBCDIC ke format ASCII yang dibatasi karakter di Amazon S3 menggunakan AWS Lambda
- Mengkonversi file data mainframe dengan layout rekaman yang kompleks menggunakan Micro Focus
- Menerapkan lingkungan untuk aplikasi Blu Age kontainer dengan menggunakan Terraform
- Hasilkan wawasan z/OS data Db2 dengan menggunakan dan AWS Mainframe Modernization Amazon Q di Quick Sight
- Hasilkan wawasan data dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight
- Menerapkan otentikasi berbasis Microsoft Entra ID dalam aplikasi mainframe modern AWS Blu Age
- Integrasikan Stonebranch Universal Controller dengan AWS Mainframe Modernisasi
- Memigrasi dan mereplikasi file VSAM ke Amazon RDS atau Amazon MSK menggunakan Connect from Excently
- Modernisasi aplikasi CardDemo mainframe dengan menggunakan AWS Transform
- Memodernisasi dan menyebarkan aplikasi mainframe menggunakan dan Terraform AWS Transform
- Modernisasi manajemen output mainframe AWS dengan menggunakan Rocket Enterprise Server dan LRS X PageCenter
- Modernisasi beban kerja pencetakan batch mainframe dengan menggunakan Rocket Enterprise Server dan LRS AWS VPSX/LFI

- Modernisasi mainframe: aktif AWS dengan Rocket DevOps Software Enterprise Suite
- Modernisasi beban kerja pencetakan online mainframe di AWS dengan menggunakan Micro Focus Enterprise Server dan LRS VPSX/MFI
- Pindahkan file mainframe langsung ke Amazon S3 menggunakan Transfer Family
- Optimalkan kinerja aplikasi modern AWS Blu Age Anda
- Mengamankan dan merampingkan akses pengguna dalam database federasi Db2 di AWS dengan menggunakan konteks tepercaya
- Transfer z/OS data Db2 skala besar ke Amazon S3 dalam file CSV
- Lebih banyak pola

Akses Layanan AWS dari IBM z/OS dengan menginstal AWS CLI

Souma Ghosh, Paulo Vitor Pereira, dan Phil de Valence, Amazon Web Services

Ringkasan

The AWS Command Line Interface (AWS CLI) adalah alat open source untuk mengelola beberapa AWS layanan dengan menggunakan perintah dalam shell baris perintah. Dengan konfigurasi minimal, Anda dapat menjalankan perintah dari sesi baris perintah seperti command prompt, terminal, dan bash shell untuk mengimplementasikan fungsionalitas yang setara dengan yang disediakan oleh browser Konsol Manajemen AWS berbasis.

Semua AWS infrastruktur sebagai layanan (IaaS) administrasi, manajemen, dan fungsi akses dalam Konsol Manajemen AWS tersedia di AWS API dan. AWS CLI Anda dapat menginstal AWS CLI pada z/OS mainframe IBM untuk langsung mengakses, mengelola, dan berinteraksi dengan Layanan AWS dari z/OS. AWS CLI Ini memungkinkan pengguna dan aplikasi untuk melakukan berbagai tugas, seperti:

- Mentransfer file atau kumpulan data antara penyimpanan objek z/OS Amazon Simple Storage Service (Amazon S3) dan melihat konten bucket
- Memulai dan menghentikan AWS sumber daya yang berbeda; misalnya, memulai pekerjaan batch di AWS Mainframe Modernization lingkungan
- Memanggil AWS Lambda fungsi untuk menerapkan logika bisnis umum
- Mengintegrasikan dengan kecerdasan buatan dan pembelajaran mesin (AI/ML) dan layanan analitik

Pola ini menjelaskan cara menginstal, mengkonfigurasi, dan menggunakan AWS CLI pada z/OS. You can install it globally, so it's available to all z/OS pengguna, atau pada tingkat pengguna. Pola ini juga merinci cara menggunakan AWS CLI dalam sesi baris perintah interaktif dari z/OS Unix System Services (USS) atau sebagai pekerjaan batch.

Prasyarat dan batasan

Prasyarat

- Komunikasi jaringan dari z/OS ke AWS

  Secara default, AWS CLI mengirim permintaan ke Layanan AWS dengan menggunakan HTTPS pada port TCP 443. AWS CLI Agar berhasil menggunakan, Anda harus dapat membuat koneksi

keluar pada port TCP 443. Anda dapat menggunakan salah satu perintah z/OS USS berikut (beberapa di antaranya mungkin tidak diinstal di lingkungan Anda) untuk menguji konektivitas jaringan dari z/OS ke AWS:

```
ping amazonaws.com
dig amazonaws.com
traceroute amazonaws.com
curl -k https://docs.aws.amazon.com/cli/v1/userguide/cli-chap-welcome.html
```

- AWS credentials

  Untuk berkomunikasi dengan AWS Cloud layanan dari z/OS, Anda harus mengonfigurasi beberapa kredensil dengan hak istimewa untuk mengakses target. AWS CLI Akun AWS Untuk perintah terprogram ke AWS, Anda dapat menggunakan kunci akses, yang terdiri dari ID kunci akses dan kunci akses rahasia. Jika Anda tidak memiliki access key, Anda dapat membuatnya dari Konsol Manajemen AWS. Sebagai praktik terbaik, jangan gunakan kunci akses untuk pengguna Akun AWS root untuk tugas apa pun kecuali pengguna root diperlukan. Sebagai gantinya, buat pengguna IAM administrator baru dan persiapkan izin hak istimewa paling sedikit untuk mengatur pengguna dengan kunci akses. Setelah Anda membuat pengguna, Anda dapat membuat ID kunci akses dan kunci akses rahasia untuk pengguna ini.

  > **Warning**
  >
  > AWS Identity and Access Management (IAM) pengguna memiliki kredensi jangka panjang yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

- IBM Python untuk z/OS

  AWS CLI Memerlukan Python 3.8 atau yang lebih baru. IBM telah mengaktifkan Python untuk berjalan z/OS dengan IBM Open Enterprise Python untuk z/OS. IBM Open Enterprise Python tersedia tanpa biaya melalui Shopz SMP/E, atau Anda dapat mengunduh file PAX dari situs web IBM. Untuk petunjuk, lihat dokumentasi instalasi dan konfigurasi untuk IBM Open Enterprise Python untuk z/OS.

Batasan

- Petunjuk penginstalan yang disediakan dalam pola ini hanya berlaku untuk AWS CLI versi 1. Versi terbaru dari versi AWS CLI ini adalah 2. Namun, pola ini menggunakan versi yang lebih lama karena metode instalasi berbeda untuk versi 2, dan executable biner yang tersedia untuk versi 2 tidak kompatibel dengan sistem. z/OS

Versi produk

- AWS CLI versi 1
- Python 3.8 atau yang lebih baru

Arsitektur

Tumpukan teknologi

- Mainframe berjalan z/OS
- Layanan Sistem z/OS UNIX Mainframe (USS)
- Mainframe Open MVS (OMVS) - Antarmuka perintah lingkungan z/OS shell UNIX
- Disk mainframe, seperti perangkat penyimpanan akses langsung (DASD)
- AWS CLI

Arsitektur target

Diagram berikut menunjukkan AWS CLI penyebaran pada IBM z/OS. Anda dapat memanggil AWS CLI dari sesi pengguna interaktif, seperti SSH, dan sesi telnet. Anda juga dapat memanggilnya dari pekerjaan batch dengan menggunakan bahasa kontrol pekerjaan (JCL), atau dari program apa pun yang dapat memanggil perintah shell z/OS Unix.

AWS CLI Berkomunikasi dengan Layanan AWS titik akhir melalui jaringan. TCP/IP Koneksi jaringan ini dapat terjadi melalui internet atau melalui AWS Direct Connect koneksi pribadi dari pusat data pelanggan ke pusat AWS Cloud data. Komunikasi diautentikasi dengan AWS kredensil dan dienkripsi.

Otomatisasi dan skala

Anda dapat menjelajahi kemampuan Layanan AWS dengan AWS CLI dan mengembangkan skrip shell USS untuk mengelola AWS sumber daya Anda dari z/OS. Anda juga dapat menjalankan AWS CLI perintah dan skrip shell dari lingkungan z/OS batch, dan Anda dapat mengotomatiskan pekerjaan batch untuk berjalan pada jadwal tertentu dengan mengintegrasikan dengan penjadwal mainframe. AWS CLI perintah atau skrip dapat dikodekan di dalam parameter (PARMs) dan prosedur (PROCs), dan dapat diskalakan dengan mengikuti pendekatan standar memanggil PARM atau PROC dari pekerjaan batch yang berbeda dengan parameter yang berbeda.

Alat

- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

Praktik terbaik

- Untuk alasan keamanan, batasi izin akses ke direktori USS tempat detail kunci AWS akses disimpan. Izinkan akses hanya ke pengguna atau program yang menggunakan AWS CLI.
- Jangan gunakan kunci akses pengguna Akun AWS root untuk tugas apa pun. Sebagai gantinya, buat pengguna IAM administrator baru untuk Anda sendiri dan atur dengan kunci akses.

> **Warning**
>
> Pengguna IAM memiliki kredensi jangka panjang yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Epik

Instal AWS CLI versi 1 di z/OS USS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Python 3.8 atau yang lebih baru.	1. Masuk ke antarmuka prompt perintah z/OS USS	Administrator mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan menggunakan salah satu metode berikut: • Gunakan perintah Time Sharing Option (TSO) OMVS dari panel Interactive System Productivity Facility (ISPF), atau • Gunakan SSH atau telnet untuk terhubung ke IP partisi logika mainframe (LPAR). Pola ini mengasumsikan bahwa `cliuser` adalah userid yang digunakan untuk masuk ke lingkungan USS dan `/u/cliuser/` merupakan direktori home untuk pengguna. Anda dapat mengatur direktori home pengguna secara berbeda di z/OS lingkungan Anda tergantung pada persyaratan instalasi Anda. 2. Ikuti panduan instalasi untuk IBM Open Enterprise Python untuk z/OS untuk menginstal Python 3.8 atau yang lebih baru jika belum diinstal.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tetapkan variabel lingkungan USS.	Tambahkan variabel lingkungan ke profil. Anda dapat menambahkan ini ke `/u/cliuser/.profile` file untuk pengguna individu (`cliuser`) atau ke `/etc/profile` file untuk semua pengguna.
	
	Note
	Pola ini mengasumsikan bahwa Python telah diinstal di `/u/awscli/python` direktori. Jika direktori instalasi Anda berbeda, perbarui kode yang sesuai.
	
	`# Python configuration`
	`export BPXKAUTOC VT='ON'`
	`export CEERUNOPT S='FILETAG(AUTOCVT ,AUTOTAG) POSIX(ON)'`
	`export TAGREDIR_ERR=txt`
	`export TAGREDIR_IN=txt`
	`export TAGREDIR_ OUT=txt`
	
	`# AWS CLI configuration`
	`export PATH=/u/cliuser/python/bin: $PATH`
		Administrator mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`export PYTHONPATH=/u/cliuser/python:$PYTHONPATH`	
Uji instalasi Python.	Jalankan perintah python: `python --version` Output harus mengkonfirmasi bahwa Anda memiliki Python 3.8 atau yang lebih baru diinstal dengan benar.	Administrator mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi atau instal pip.	1. Perintah pip biasanya diinstal secara otomatis ketika Anda menginstal Python dari situs web IBM. Untuk memverifikasi, jalankan perintah:
	`pip —-version`
	Jika pip diinstal, perintah ini harus menampilkan versi yang diinstal.
	2. Jika perintah pip tidak ditemukan, instal pip dengan menjalankan perintah berikut:
	`python -m ensurepip --upgrade`
	Untuk opsi penginstalan lainnya, lihat dokumentasi pip.
		Administrator mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal AWS CLI versi 1.	1. Untuk menginstal AWS CLI, jalankan perintah:
	`python -m pip install awscli`
	Output harus serupa dengan yang berikut ini.
	`Successfully installed PyYAML-6.0.1 awscli-1.32.23 botocore-1.34.23 colorama-0.4.4 docutils-0.16 jmespath-1.0.1 pyasn1-0.5.1 python-dateutil-2.8.2 rsa-4.7.2 s3transfer-0.10.0 urllib3-2.0.7`
	2. Ubah izin aws yang dapat dieksekusi dengan menjalankan perintah berikut. Pastikan untuk memperbarui direktori placeholder `<python_installation_dir>` dengan jalur instalasi Python Anda.
	`chmod 744 <python_installation_dir>/bin/aws`
		Administrator mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Jalankan perintah berikut untuk menguji AWS CLI instalasi:
	`aws --version`
	Output harus menunjukkan versi AWS CLI, Python, dan botocore, mirip dengan yang berikut ini.
	`aws-cli/1.32.3 Python/3.9.5 OS/390/27.00 botocore/1.34.3`	

Konfigurasikan AWS CLI akses dari z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan tombol AWS akses, Wilayah default, dan output.	AWS CLI Dokumentasi menjelaskan berbagai opsi untuk mengatur AWS akses. Anda dapat memilih konfigurasi sesuai dengan standar organisasi Anda. Contoh ini menggunakan konfigurasi kredensi jangka pendek. 1. Konfigurasikan AWS CLI dengan perintah berikut: `aws configure`	Administrator AWS, administrator Mainframe, z/OS pengembang z/OS Mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Berikan detail untuk item berikut saat diminta. ID kunci akses dan nilai kunci akses rahasia berasal dari kunci yang Anda peroleh saat Anda menetapkan AWS kredensional dalam langkah-langkah Prasyarat.
	
	`AWS Access Key ID [None]: ASIAIOSFODNN7EXAMPLE`
	`AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`
	`Default region name [None]: us-east-1`
	`Default output format [None]:`
	`aws configure set aws_session_token IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE`
	
	Konfigurasi ini, termasuk kunci akses, disimpan di `/u/cliuser/.aws` folder. Untuk alasan keamanan, batasi folder ini untuk mengizinkan akses hanya ke pengguna atau program yang menggunakan file. AWS CLI	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji AWS CLI.	1. Jalankan perintah berikut di command prompt untuk menguji AWS CLI dengan perintah sederhana: `aws s3 ls` Output harus mencantum kan semua bucket S3 untuk yang dikonfigurasi Akun AWS tanpa kesalahan. 2. Ikuti instruksi dalam dua epos berikutnya untuk mentransfer data dari USS ke Amazon S3. Anda dapat memilih salah satu dari dua opsi ini: • Opsi 1 (epik berikutny a): Secara interaktif mentransfer file nilai dipisahkan koma (CSV) EBCDIC ke Amazon S3 dan menanyakan file dari Amazon Athena. • Opsi 2: Transfer kumpulan data panjang tetap EBCDIC ke Amazon S3 sebagai pekerjaan batch.	z/OS Administrator mainframe, pengembang z/OS Mainframe

Opsi 1 - Transfer data dari USS ke Amazon S3 secara interaktif dari sesi USS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh dan transfer file CSV sampel.	1. Unduh `sales-records.csv` dari bagian Lampiran. File ini menyediakan contoh file CSV dengan catatan penjualan. 2. Transfer file ke z/OS USS. 3. Verifikasi bahwa `/u/cliuser/sales-records.csv` file tersebut dapat dibaca dalam format EBCDIC di USS dengan menggunakan editor teks pilihan Anda.	Pengembang aplikasi, pengembang Mainframe z/OS

Buat bucket S3 dan unggah file CSV.	1. Buat bucket S3 untuk menyimpan file CSV.
	`aws s3 mb s3://<s3_bucket_name>`
	di `<s3_bucket_name>` mana nama unik untuk ember; misalnya:
	`aws s3 mb s3://DOC-EXAMPLE-BUCKET1`
	2. Unggah file CSV dari z/OS USS ke bucket S3:
		Pengembang aplikasi, pengembang Mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`aws s3 cp <csv_file _path> s3://<s3_ bucket_name>`
	
	Contoh:
	
	`aws s3 cp /u/cliuser/ sales-records.csv s3://DOC-EXAMPLE-B UCKET1`
	
	3. Buat daftar isi bucket S3 dan konfirmasikan bahwa mereka menyertakan file yang diunggah:
	
	`aws s3 ls s3://<s3_ bucket_name>`
	
	Contoh:
	
	`aws s3 ls s3://DOC-EXAMPLE-BUCKET1`	

Lihat bucket S3 dan file yang diunggah.	1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3. 2. Arahkan untuk melihat bucket S3 baru dan objek yang diunggah. Untuk informasi selengkapnya tentang mengunggah objek, lihat Memulai Amazon S3 di dokumentasi Amazon S3.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan kueri SQL pada tabel Amazon Athena.	1. Buka konsol Amazon Athena.
	2. Buat tabel baru (misalnya ,`DOC-EXAMPLE-BUCKET`) dengan menggunakan data CSV dari Amazon S3. Untuk informasi selengkapnya, lihat Menanyakan Inventaris Amazon S3 dengan Amazon Athena di dokumentasi Amazon S3.
	3. Jalankan `SELECT` kueri terhadap tabel untuk melihat data.
	`SELECT * FROM <table_name>;`
	Contoh:
	`SELECT * FROM DOC-EXAMPLE-BUCKET;`
	Output dari query SQL akan menampilkan isi file CSV Anda.
		AWS Umum, Pengembang aplikasi

Opsi 2 - Transfer data dari USS ke Amazon S3 dengan menggunakan batch JCL

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file sampel.	1. Unduh `sales-records-fixed.txt`	Pengembang mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dari bagian Lampiran. Ini adalah contoh file dengan catatan penjualan. Ganti nama file teks; misalnya, menjadi`USER.DATA .FIXED` . 2. Transfer file ke z/OS sebagai dataset fixed-blocked (FB), 256 record length (LRECL), physical sequential (PS). 3. Gunakan utilitas daftar kumpulan data untuk memverifikasi bahwa kumpulan `USER.DATA .FIXED` data dapat dibaca dalam format EBCDIC di bawah opsi ISPF 3.4. Lihat bagian Informasi tambahan misalnya keluaran.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat batch JCL.	Kodekan JCL batch sebagai berikut untuk membuat bucket S3 tujuan, mengunggah kumpulan data, dan mencantumkan konten bucket. Pastikan untuk mengganti nama direktori, nama file, dan nama bucket ke nilai Anda sendiri.
	
	`//AWSCLICP JOB`
	` ACTINFO1,'IBMUSER'`
	`,CLASS=A,MSGCLASS=`
	`H,MSGLEVEL=(1,1),`
	`// NOTIFY=&SYSUID,TIM`
	`E=1440`
	`//*--------------`
	`------------------`
	`------------------`
	`-------`
	`//* Sample job for AWS`
	` CLI`
	`//*--------------`
	`------------------`
	`------------------`
	`-------`
	`//USSCMD EXEC PGM=BPXBA`
	`TCH`
	`//STDERR  DD SYSOUT=*`
	`//STDOUT  DD SYSOUT=*`
	`//STDENV  DD *`
	` export PATH=/u/c`
	`liuser/python/bin:`
	`$PATH`
	`//STDPARM DD *`
	`SH`
	` export _BPXK_AUT`
	`OCVT=ON;`
		Pengembang mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>aws s3 mb s3://DOC-EXAMPLE-BUCKET2;
	cp "//'USER.DATA.FIXED'" /tmp/tmpfile;
	aws s3 cp /tmp/tmpfile s3://DOC-EXAMPLE-BUCKET2/USER.DATA.FIXED;
	rm /tmp/tmpfile;
	aws s3 ls s3://DOC-EXAMPLE-BUCKET2;
	/*</pre>	

Kirimkan pekerjaan batch JCL.	1. Kirimkan job JCL yang Anda kodekan pada langkah sebelumnya. 2. Periksa status pekerjaan di System Display and Search Facility (SDSF). Jika berhasil, pekerjaan harus diakhiri dengan kode pengembalian 0. 3. Output standar (STDOUT) dari log pekerjaan menunjukkan keberhasilan pembuatan bucket, upload dataset, dan daftar konten bucket. Untuk contoh ilustrasi layar, lihat bagian Informasi tambahan.	Pengembang mainframe z/OS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat kumpulan data yang diunggah ke bucket S3.	1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3. 2. Arahkan untuk melihat file yang diunggah di bucket pengujian. 3. Anda dapat memproses `USER.DATA.FIXED` file lebih lanjut atau menganalisisnya dengan menggunakan layanan analitik seperti Amazon Redshift.	AWS Umum

Sumber daya terkait

- AWS CLI Dokumentasi versi 1
- AWS Mainframe Modernization Referensi Perintah CLI
- AWS Mainframe Modernization

Informasi tambahan

USER.DATA.FIXED dalam opsi ISPF 3.4 (utilitas daftar dataset)

SYSOUT dari pekerjaan batch yang dikirimkan

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Cadangkan dan arsipkan data mainframe ke Amazon S3 menggunakan BMC AMI Cloud Data

Santosh Kumar Singh, Gilberto Biondo, dan Maggie Li, Amazon Web Services

Mikhael Liberman, Perangkat Lunak Mainframe Model9

Ringkasan

Pola ini menunjukkan cara mencadangkan dan mengarsipkan data mainframe langsung ke Amazon Simple Storage Service (Amazon S3), lalu mengingat dan mengembalikan data tersebut ke mainframe dengan menggunakan BMC AMI Cloud Data (sebelumnya dikenal sebagai Model9 Manager). Jika Anda mencari cara untuk memodernisasi solusi cadangan dan arsip Anda sebagai bagian dari proyek modernisasi mainframe atau untuk memenuhi persyaratan kepatuhan, pola ini dapat membantu memenuhi tujuan tersebut.

Biasanya, organisasi yang menjalankan aplikasi bisnis inti pada mainframe menggunakan pustaka pita virtual (VTL) untuk mencadangkan penyimpanan data seperti file dan log. Metode ini bisa mahal karena mengkonsumsi MIPS yang dapat ditagih, dan data yang disimpan pada kaset di luar mainframe tidak dapat diakses. Untuk menghindari masalah ini, Anda dapat menggunakan BMC AMI Cloud Data untuk mentransfer data mainframe operasional dan historis secara cepat dan hemat biaya langsung ke Amazon S3. Anda dapat menggunakan BMC AMI Cloud Data untuk mencadangkan dan mengarsipkan data AWS sambil TCP/IP memanfaatkan mesin IBM z Integrated Information Processor (ZiIP) untuk mengurangi biaya, paralelisme, dan waktu transfer.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- BMC AMI Cloud Data dengan kunci lisensi yang valid
- Konektivitas TCP/IP antara mainframe dan AWS
- Peran AWS Identity and Access Management (IAM) untuk read/write akses ke bucket S3
- Akses produk keamanan mainframe (RACF) untuk menjalankan proses BMC AMI Cloud
- z/OS Agen BMC AMI Cloud (Java versi 8 64-bit SR5 FP16 atau lebih baru) yang memiliki port jaringan yang tersedia, aturan firewall yang mengizinkan akses ke bucket S3, dan sistem file khusus z/FS
- Persyaratan terpenuhi untuk server manajemen BMC AMI Cloud

Batasan

- BMC AMI Cloud Data menyimpan data operasionalnya dalam database PostgreSQL yang berjalan sebagai wadah Docker pada instance Amazon Elastic Compute Cloud (Amazon) yang sama dengan server manajemen. EC2 Amazon Relational Database Service (Amazon RDS) saat ini tidak didukung sebagai backend untuk BMC AMI Cloud Data. Untuk informasi selengkapnya tentang pembaruan produk terbaru, lihat Apa yang Baru? dalam dokumentasi BMC.
- Pola ini mencadangkan dan mengarsipkan data z/OS mainframe saja. BMC AMI Cloud Data mencadangkan dan mengarsipkan hanya file mainframe.
- Pola ini tidak mengubah data menjadi format terbuka standar seperti JSON atau CSV. Gunakan layanan transformasi tambahan seperti BMC AMI Cloud Analytics (sebelumnya dikenal sebagai Model9 Gravity) untuk mengonversi data menjadi format terbuka standar. Aplikasi cloud-native dan alat analisis data dapat mengakses data setelah ditulis ke cloud.

Versi produk

- BMC AMI Cloud Data versi 2.x

Arsitektur

Tumpukan teknologi sumber

- Mainframe berjalan z/OS
- File mainframe seperti kumpulan data dan file z/OS UNIX System Services (USS)
- Disk mainframe, seperti perangkat penyimpanan akses langsung (DASD)
- Mainframe tape (pustaka rekaman virtual atau fisik)

Tumpukan teknologi target

- Amazon S3
- EC2 Instans Amazon di cloud pribadi virtual (VPC)
- AWS Direct Connect
- Amazon Elastic File System (Amazon EFS)

Arsitektur target

Diagram berikut menunjukkan arsitektur referensi di mana agen perangkat lunak BMC AMI Cloud Data pada mainframe mendorong proses pencadangan dan arsip data lama yang menyimpan data di Amazon S3.

Diagram menunjukkan alur kerja berikut:

1. Agen perangkat lunak BMC AMI Cloud Data berjalan pada partisi logis mainframe (). LPARs Agen perangkat lunak membaca dan menulis data mainframe dari DASD atau merekam langsung ke Amazon S3 melalui TCP/IP.
2. AWS Direct Connect mengatur koneksi fisik dan terisolasi antara jaringan lokal dan AWS. Untuk keamanan yang ditingkatkan, jalankan site-to-site VPN di atas Direct Connect untuk mengenkripsi data dalam perjalanan.
3. Bucket S3 menyimpan file mainframe sebagai data penyimpanan objek, dan agen BMC AMI Cloud Data langsung berkomunikasi dengan bucket S3. Sertifikat digunakan untuk enkripsi HTTPS dari semua komunikasi antara agen dan Amazon S3. Enkripsi data Amazon S3 digunakan untuk mengenkripsi dan melindungi data saat istirahat.
4. Server manajemen BMC AMI Cloud Data berjalan sebagai kontainer Docker pada EC2 instance. Instans berkomunikasi dengan agen yang berjalan di bucket mainframe LPARs dan S3.
5. Amazon EFS dipasang pada EC2 instans aktif dan pasif untuk berbagi penyimpanan Network File System (NFS). Ini untuk memastikan bahwa metadata yang terkait dengan kebijakan yang dibuat di server manajemen tidak hilang jika terjadi failover. Jika terjadi failover oleh server aktif, server pasif dapat diakses tanpa kehilangan data. Jika server pasif gagal, server aktif dapat diakses tanpa kehilangan data.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di file AWS Cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil hampir semua jumlah data.

- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS
- AWS Direct Connectmenghubungkan jaringan internal Anda ke AWS Direct Connect lokasi melalui kabel serat optik Ethernet standar. Dengan koneksi ini, Anda dapat membuat antarmuka virtual langsung ke AWS layanan publik sambil melewati penyedia layanan internet di jalur jaringan Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Alat BMC

- BMC AMI Cloud management server adalah aplikasi GUI yang berjalan sebagai wadah Docker pada Amazon Linux Amazon Machine Image (AMI) untuk Amazon. EC2 Server manajemen menyediakan fungsionalitas untuk mengelola aktivitas BMC AMI Cloud seperti melaporkan, membuat dan mengelola kebijakan, menjalankan arsip, dan melakukan pencadangan, penarikan, dan pemulihan.
- Agen BMC AMI Cloud berjalan pada LPAR mainframe lokal yang membaca dan menulis file langsung ke penyimpanan objek dengan menggunakan TCP/IP. Tugas yang dimulai berjalan pada LPAR mainframe dan bertanggung jawab untuk membaca dan menulis cadangan dan mengarsipkan data ke dan dari Amazon S3.
- BMC AMI Cloud Mainframe Command Line Interface (M9CLI) memberi Anda serangkaian perintah untuk melakukan tindakan BMC AMI Cloud langsung dari TSO/E atau dalam operasi batch, tanpa ketergantungan pada server manajemen.

Epik

Buat bucket S3 dan kebijakan IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket S3 untuk menyimpan file dan volume yang ingin Anda cadangkan	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan arsipkan dari lingkungan mainframe Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM.	Semua server dan agen manajemen BMC AMI Cloud memerlukan akses ke bucket S3 yang Anda buat pada langkah sebelumnya. Untuk memberikan akses yang diperlukan, buat kebijakan IAM berikut: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "Listfolder", "Action": [ "s3:ListBucket", "s3:GetBucketLocation", "s3:ListBucketVersions" ], "Effect": "Allow", "Resource": [ "arn:aws:s3:::<BucketName>" ] }, { "Sid": "Objectaccess",`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObjectAcl", "s3:GetObject", "s3:DeleteObjectVersion", "s3:DeleteObject", "s3:PutObjectAcl", "s3:GetObjectVersion"], "Resource": ["arn:aws:s3:::<BucketName>/*"] }] }</pre>	

Dapatkan lisensi perangkat lunak BMC AMI Cloud dan unduh perangkat lunaknya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan lisensi perangkat lunak BMC AMI Cloud.	Untuk mendapatkan kunci lisensi perangkat lunak, hubungi tim BMC AMI Cloud. Output dari z/OS `D M=CPU`	Membangun memimpin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	perintah diperlukan untuk menghasilkan lisensi.	
Unduh perangkat lunak BMC AMI Cloud dan kunci lisensi.	Dapatkan file instalasi dan kunci lisensi dengan mengikuti instruksi dalam dokumentasi BMC.	Administrator infrastruktur mainframe

Instal agen perangkat lunak BMC AMI Cloud di mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal agen perangkat lunak BMC AMI Cloud.	1. Sebelum Anda memulai proses instalasi, verifikasi bahwa persyaratan perangkat lunak dan perangkat keras minimum untuk agen telah terpenuhi. 2. Untuk menginstal agen, ikuti instruksi dalam dokumentasi BMC. 3. Setelah agen mulai berjalan pada mainframe LPAR, periksa `ZM91000I MODEL9 BACKUP AGENT INITIALIZED` pesan di spool. Verifikasi bahwa konektivitas berhasil dibuat antara agen dan bucket S3 dengan mencari `Object store connectivity has been established successfully` pesan di STDOUT agen.	Administrator infrastruktur mainframe

Siapkan server manajemen BMC AMI Cloud pada sebuah instans EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instans Amazon EC2 Linux 2.	Luncurkan dua instans Amazon EC2 Linux 2 di Availability Zone yang berbeda dengan mengikuti petunjuk dari Langkah 1: Luncurkan instance di EC2 dokumentasi Amazon. Instans harus memenuhi persyaratan perangkat keras dan perangkat lunak yang direkomendasikan berikut: • CPU - Minimal 4 core • RAM - Minimal 8 GB • Drive - 40 GB • EC2 Contoh yang disarankan — C5.xLarge • OS — Linux • Perangkat lunak - Docker, unzip, VI/VIM • Bandwidth jaringan - Minimum 1 GB Untuk informasi lebih lanjut, lihat dokumentasi BMC.	Arsitek cloud, Administrator Cloud
Buat sistem file Amazon EFS.	Buat sistem file Amazon EFS dengan mengikuti petunjuk dari Langkah 1: Buat sistem file Amazon EFS Anda di dokumentasi Amazon EFS.	Administrator cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Saat membuat sistem file, lakukan hal berikut: • Pilih kelas penyimpanan Standar. • Pilih VPC yang sama dengan yang Anda gunakan untuk meluncurkan instans Anda EC2 .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal Docker dan konfigura sikan server manajemen.	Connect ke EC2 instans Anda:
	
	Connect ke EC2 instans Anda dengan mengikuti petunjuk dari Connect ke instans Linux Anda dalam EC2 dokumentasi Amazon.
	
	Konfigurasikan EC2 instans Anda:
	
	Untuk setiap EC2 contoh, lakukan hal berikut:
	
	1. Untuk menginstal Docker, jalankan perintah:
	
	`sudo yum install docker`
	
	2. Untuk memulai Docker, jalankan perintah:
	
	`sudo service docker start`
	
	3. Untuk memvalidasi status Docker, jalankan perintah:
	
	`sudo service docker status`
	
	4. Di `/etc/selinux` folder, ubah `config` file menjadi`SELINUX=permissive` .
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	5. Unggah `VerificationScripts.zip` file `model9-v2.x.y_build_build-id-server.zip` dan (yang Anda unduh sebelumnya) ke folder sementara di EC2 salah satu contoh (misalnya, ke `/var/tmp` folder dalam instance Anda).
	6. Untuk pergi ke `tmp` folder, jalankan perintah:
	`cd/var/tmp`
	7. Untuk membuka zip skrip verifikasi, jalankan perintah:
	`unzip VerificationScripts.zip`
	8. Untuk mengubah direktori, jalankan perintah:
	`cd /var/tmp/sysutils/PrereqsScripts`
	9. Untuk menjalankan skrip verifikasi, jalankan perintah:
	`./M9VerifyPrereqs.sh`
	10. Setelah skrip verifikasi meminta input, masukkan URL Amazon S3 dan nomor	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	port. Kemudian, masukkan z/OS IP/DNS dan nomor port. **Note** Skrip menjalankan pemeriksaan untuk mengonfirmasi bahwa EC2 instance dapat terhubung dengan bucket S3 dan agen yang berjalan di mainframe. Jika koneksi dibuat, pesan sukses ditampilkan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal perangkat lunak server manajemen.	1. Buat folder dan subfolder di direktori root (misalnya ,`/data/model9`) dalam EC2 contoh yang Anda rencanakan untuk membuat server aktif.
	2. Untuk menginstal `amazon-efs-utils` paket dan memasang sistem file Amazon EFS yang dibuat sebelumnya, jalankan perintah berikut:
	`sudo yum install -y amazon-efs-utils`
	`sudo mount -t efs -o tls <File System ID>:/ /data/model9`
	3. Untuk memperbarui `/etc/fstab` file EC2 instans dengan entri untuk sistem file Amazon EFS (sehingga Amazon EFS secara otomatis di-restart saat Amazon EC2 reboot), jalankan perintah:
	`<Amazon-EFS-file-system-id>:/ /data/model9 efs defaults,_netdev 0 0`
	4. Untuk menentukan jalur ke file instalasi BMC AMI Cloud dan lokasi instalasi
		Arsitek cloud, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	target, jalankan perintah berikut untuk mengekspor variabel:
	`export MODEL9_HOME=/data/model9`
	`export M9INSTALL=/var/tmp`
	Note
	Kami menyarankan Anda menambahkan perintah EKSPOR ini ke `.bashrc` skrip Anda.
	5. Untuk mengubah direktori, jalankan `cd $MODEL9_HOME` perintah, dan kemudian buat subdirektori lain dengan menjalankan `mkdir diag` perintah.
	6. Untuk membuka zip file instalasi, jalankan perintah:
	`unzip $M9INSTALL/model9-<v2.x.y>_build_<build-id>-server.zip`
	Note
	Ganti `x.y` (versi) dan `build-id`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dengan nilai-nilai Anda.
	7. Untuk menyebarkan aplikasi, jalankan perintah berikut:
	`docker load -i $MODEL9_HOME/model9-<v2.x.y>_build_<build-id>.docker`
	`docker load -i $MODEL9_HOME/postgres-12.10-x86.docker.gz`
	Note
	Ganti `v2.x.y` (versi) dan `build-id` dengan nilai-nilai Anda.
	8. Di `$MODEL9_HOME/conf` folder, perbarui `model9-local.yml` file.
	Note
	Beberapa parameter memiliki nilai default dan lainnya dapat diperbarui seperlunya. Untuk informasi selengkapnya, lihat petunjuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dalam `model9-local.yml` file.
	9. Buat file bernama **`$MODEL9_HOME/conf`**, lalu tambahkan parameter berikut ke file:
	`TZ=America/New_York`
	`EXTRA_JVM_ARGS=-Xmx2048m`
	10. Untuk membuat jembatan jaringan Docker, jalankan perintah:
	`docker network create -d bridge model9network`
	11. Untuk memulai wadah database PostgreSQL untuk BMC AMI Cloud, jalankan perintah berikut:
	`docker run -p 127.0.0.1:5432:5432 \`
	`-v $MODEL9_HOME/db/data:/var/lib/postgresql/data:z \`
	`--name model9db --restart unless-stopped \`
	`--network model9network \`
	`-e POSTGRES_PASSWORD=model9 -e POSTGRES_`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`DB=model9 -d postgres:12.10`	
	12. Setelah wadah PostgreSQL mulai berjalan, jalankan perintah berikut untuk memulai server aplikasi: `docker run -d -p 0.0.0.0:443:443 -p 0.0.0.0:80:80 \ --sysctl net.ipv4.tcp_keepalive_time=600 \ --sysctl net.ipv4.tcp_keepalive_intvl=30 \ --sysctl net.ipv4.tcp_keepalive_probes=10 \ -v $MODEL9_HOME:/model9:z -h $(hostname) --restart unless-stopped \ --env-file $MODEL9_HOME/conf/model9.env \ --network model9network \ --name model9-v2.x.y model9:<v2.x.y>.<build-id>` **Note** Ganti `v2.x.y` (versi) dan `build-id` dengan nilai-nilai Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	13. Untuk memeriksa status kesehatan kedua kontainer, jalankan perintah: `docker ps -a` 14. Untuk menginstal server manajemen pada EC2 instance pasif, ulangi langkah 1-4, 7, dan 10—13. **Note** Untuk memecahkan masalah, buka log yang disimpan di folder. `/data/model9/logs/` Untuk informasi lebih lanjut, lihat dokumentasi BMC.	

Tambahkan agen dan tentukan kebijakan cadangan atau arsip di server manajemen BMC AMI Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan agen baru.	Sebelum Anda menambahkan agen baru, konfirmasikan hal berikut: • Agen BMC AMI Cloud berjalan di mainframe LPAR dan telah diinisialisasi sepenuhnya.	Administrator atau pengembang penyimpanan mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Identifikasi agen dengan mencari pesan `ZM91000I MODEL9 BACKUP AGENT INITIALIZED` inisialisasi di spool. • Sebuah wadah Docker untuk server manajemen sepenuhnya diinisialisasi dan berjalan. Anda harus membuat agen di server manajemen sebelum menentukan kebijakan cadangan dan arsip apa pun. Untuk membuat agen, lakukan hal berikut: 1. Gunakan browser web untuk mengakses server manajemen yang digunakan di EC2 mesin Amazon Anda, lalu masuk dengan kredensyal mainframe Anda. 2. Pilih tab AGENTS, lalu pilih ADD NEW AGENT. 3. Untuk Nama, masukkan nama agen. 4. Untuk nama host/Alamat IP, masukkan nama host atau alamat IP mainframe Anda. 5. Untuk Port, masukkan nomor port Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Pilih TEST CONNECTION. Anda dapat melihat pesan sukses jika konektivitas berhasil dibuat. 7. Pilih CREATE. Setelah agen dibuat, Anda akan melihat status terhubung terhadap penyimpanan objek dan agen mainframe di jendela baru yang muncul di tabel.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan cadangan atau arsip.	1. Pilih KEBIJAKAN. 2. Pilih BUAT KEBIJAKAN. 3. Pada halaman BUAT KEBIJAKAN BARU, masukkan spesifikasi kebijakan Anda. **Note** Untuk informasi selengkapnya tentang spesifikasi yang tersedia, lihat Membuat kebijakan baru di dokumentasi BMC. 4. Pilih Selesai. 5. Kebijakan baru sekarang terdaftar sebagai tabel. Untuk melihat tabel ini, pilih tab KEBIJAKAN.	Administrator atau pengembang penyimpanan mainframe

Jalankan kebijakan cadangan atau arsip dari server manajemen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kebijakan cadangan atau arsip.	Jalankan kebijakan pencadangan atau arsip data yang Anda buat sebelumnya dari server manajemen baik secara manual maupun otomatis (berdasarkan jadwal).	Administrator atau pengembang penyimpanan mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk menjalankan kebijakan secara manual: 1. Pilih tab KEBIJAKAN dari menu navigasi. 2. Di sisi kanan tabel untuk kebijakan yang ingin Anda jalankan, pilih menu tiga titik. 3. Pilih Jalankan Sekarang. 4. Di jendela konfirmasi pop-up, pilih YA, JALANKAN KEBIJAKAN SEKARANG. 5. Setelah kebijakan berjalan, verifikasi status jalankan di bagian aktivitas kebijakan. 6. Untuk kebijakan yang dijalankan, pilih menu tiga titik, lalu pilih Lihat Jalankan Log untuk melihat log. 7. Untuk memverifikasi bahwa cadangan telah dibuat, periksa bucket S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan kebijakan cadangan atau arsip.	1. Pada menu navigasi, pilih tab KEBIJAKAN. 2. Pilih kebijakan untuk menjalankan proses pemulihan. Ini akan mencantumkan semua aktivitas pencadangan atau arsip yang berjalan di masa lalu untuk kebijakan spesifik tersebut. 3. Untuk memilih cadangan yang ingin Anda pulihkan, pilih kolom Tanggal-waktu. Nama `file/Volume/Storage` grup menunjukkan rincian proses kebijakan. 4. Di sisi kanan tabel, pilih menu tiga titik, lalu pilih RESTORE. 5. Di jendela pop-up, masukkan nama target, volume, dan grup penyimpanan Anda, lalu pilih RESTORE. 6. Masukkan kredenal mainframe Anda, lalu pilih RESTORE lagi. 7. Untuk memverifikasi bahwa pemulihan berhasil, periksa log atau mainframe.	Administrator atau pengembang penyimpanan mainframe

Jalankan kebijakan cadangan atau arsip dari mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kebijakan cadangan atau arsip dengan menggunakan M9CLI.	Gunakan M9CLI untuk melakukan proses pencadangan dan pemulihan dari TSO/E, REXX, atau melalui JCLs tanpa mengatur aturan di server manajemen BMC AMI Cloud. Menggunakan TSO/E: Jika Anda menggunakan TSO/E, pastikan `M9CLI REXX` itu digabungkan ke. `TSO` Untuk mencadangkan dataset melalui TSO/E, gunakan perintah. `TSO M9CLI BACKDSN <DSNAME>` **Note** Untuk informasi selengkapnya tentang perintah M9CLI, lihat referensi CLI dalam dokumentasi BMC. Menggunakan JCLs: Untuk menjalankan kebijakan pencadangan dan arsip dengan menggunakan JCLs, jalankan `M9CLI` perintah.	Administrator atau pengembang penyimpanan mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Menggunakan operasi batch:
	
	Contoh berikut menunjukkan cara mengarsipkan dataset dengan menjalankan `M9CLI` perintah dalam batch:
	
	`//JOBNAME JOB …`
	`//M9CLI EXEC PGM=IKJEF`
	`T01`
	`//STEPLIB DD DISP=SHR,`
	`DSN=<MODEL9 LOADLIB>`
	`//SYSEXEC DD DISP=SHR,`
	`DSN=<MODEL9 EXEC LIB>`
	`//SYSTSPRT DD SYSOUT=*`
	`//SYSPRINT DD SYSOUT=*`
	`//SYSTSIN DD TSO M9CLI`
	` ARCHIVE`
	` M9CLI ARCHIVE <DSNNAME`
	` OR DSN PATTERN>`
	`/`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan kebijakan pencadangan atau arsip dalam batch JCL.	BMC AMI Cloud menyediakan sampel rutin JCL yang disebut M9SAPIJ. Anda dapat menyesuaikan M9SAPIJ untuk menjalankan kebijakan tertentu yang dibuat di server manajemen dengan JCL. Pekerjaan ini juga dapat menjadi bagian dari penjadwal batch untuk menjalankan proses pencadangan dan pemulihan secara otomatis. Pekerjaan batch mengharapkan nilai wajib berikut: • address/host Nama IP server manajemen • Nomor port • ID kebijakan atau nama kebijakan (yang dibuat di server manajemen) **Note** Anda juga dapat mengubah nilai lain dengan mengikuti instruksi pada pekerjaan sampel.	Administrator atau pengembang penyimpanan mainframe

Sumber daya terkait

- Modernisasi Mainframe dengan AWS (dokumentasi AWS)
- Bagaimana Cloud Backup untuk Mainframe Memotong Biaya dengan Model9 dan AWS (AWS Partner Network Blog)
- Cara Mengaktifkan Analisis Data Mainframe di AWS Menggunakan Model9 (AWS Partner Network Blog)
- Rekomendasi Ketahanan AWS Direct Connect (dokumentasi AWS)
- Dokumentasi BMC AMI Cloud (situs BMC)

Membangun program COBOL Db2 dengan menggunakan dan AWS Mainframe Modernization AWS CodeBuild

Luis Gustavo Dantas dan Eduardo Zimelewicz, Amazon Web Services

Ringkasan

> **Note**
>
> AWS Mainframe Modernization Layanan (Managed Runtime Environment experience) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Pola ini menjelaskan cara membuat AWS CodeBuild proyek sederhana untuk mengkompilasi dan mengikat program COBOL Db2 dengan menggunakan alat Replatform. AWS Mainframe Modernization Ini memungkinkan penerapan dan eksekusi program-program ini di lingkungan runtime AWS Mainframe Modernization Replatform.

COBOL, bahasa pemrograman berorientasi bisnis, mendukung banyak aplikasi penting karena keandalan dan keterbacaannya. IBM Db2, sistem manajemen database relasional, mengelola volume data yang besar secara efisien dan terintegrasi dengan program COBOL melalui SQL. Bersama-sama, COBOL dan Db2 membentuk tulang punggung operasi misi kritis di industri seperti keuangan dan pemerintah, meskipun munculnya teknologi yang lebih baru.

Migrasi komponen COBOL dan Db2 dari lingkungan mainframe ke platform lain mengarah pada tantangan seperti kompatibilitas platform, kompleksitas integrasi, migrasi data, dan pengoptimalan kinerja. Memindahkan komponen penting ini membutuhkan perencanaan yang cermat, keahlian teknis, dan sumber daya untuk memastikan migrasi yang lancar sambil mempertahankan keandalan dan fungsionalitas.

AWS Mainframe Modernization Layanan ini menyediakan alat dan sumber daya untuk memplatform ulang aplikasi mainframe dan database untuk berjalan pada AWS infrastruktur, seperti instans Amazon Elastic Compute Cloud (Amazon). EC2 Ini melibatkan pemindahan beban kerja mainframe ke cloud tanpa perubahan kode besar.

Proses precompile dan bind Db2 sangat penting untuk mengoptimalkan kinerja dan keandalan aplikasi database. Prakompilasi mengubah pernyataan SQL yang disematkan menjadi kode yang dapat dieksekusi, yang mengurangi overhead runtime dan meningkatkan efisiensi. Proses mengikat menghubungkan kode yang telah dikompilasi sebelumnya dengan struktur database, memfasilitasi jalur akses dan optimasi kueri. Proses ini memastikan integritas data, meningkatkan daya tanggap aplikasi, dan menjaga terhadap kerentanan keamanan. Aplikasi yang dikompilasi dan terikat dengan benar meminimalkan konsumsi sumber daya, meningkatkan skalabilitas, dan mengurangi risiko serangan injeksi SQL.

Prasyarat dan batasan

Prasyarat

- Akses konsol tingkat administratif Akun AWS dan administratif.
- Sistem database IBM Db2, seperti IBM Db2 untuk z/OS atau Db2 untuk Linux, Unix, dan Windows (LUW).
- Perangkat lunak IBM Data Server Client, yang tersedia untuk diunduh dari situs web IBM. Untuk informasi selengkapnya, lihat tipe IBM Data Server Client dan Data Server Driver.
- Program COBOL Db2 yang akan dikompilasi dan diikat. Atau, pola ini menyediakan program sampel dasar yang dapat Anda gunakan.
- Virtual Private Cloud (VPC) aktif AWS dengan jaringan pribadi. Untuk informasi tentang membuat VPC, lihat dokumentasi Amazon Virtual Private Cloud (Amazon VPC).
- Sebuah repositori kontrol sumber seperti GitHub atau. GitLab

Keterbatasan

- Untuk AWS CodeBuild kuota, lihat Kuota untuk. AWS CodeBuild
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Tumpukan teknologi sumber

Tumpukan sumber meliputi:

- Program COBOL yang menggunakan database Db2 untuk menyimpan data
- Kompiler IBM COBOL dan Db2 untuk precompiler z/OS
- Bagian lain dari pengaturan mainframe, seperti sistem file, manajer transaksi, dan spool

Tumpukan teknologi target

Pendekatan pola ini berfungsi untuk dua opsi: memindahkan data dari Db2 ke Db2 z/OS untuk LUW, atau tetap menggunakan Db2 untuk z/OS. Arsitektur target meliputi:

- Program COBOL yang menggunakan database Db2 untuk menyimpan data
- AWS Mainframe Modernization Alat kompilasi replatform
- AWS CodeBuild sebagai infrastruktur untuk membangun aplikasi
- AWS Cloud Sumber daya lain seperti Amazon Linux

Arsitektur target

Diagram ini menggambarkan hal sebagai berikut:

1. Pengguna mengunggah kode mereka ke repositori kontrol sumber seperti atau. GitHub GitLab
2. AWS CodePipeline memperhatikan perubahan dan mendapatkan kode dari repositori.
3. CodePipeline Memulai AWS CodeBuild dan mengirimkan kode
4. CodeBuild mengikuti instruksi dalam `buildspec.yml` template (disediakan di bagian Informasi tambahan) untuk:
   a. Dapatkan Klien Server Data IBM dari bucket Amazon Simple Storage Service (Amazon S3).
   b. Instal dan atur IBM Data Server Client.
   c. Ambil kredensi Db2 dari. AWS Secrets Manager
   d. Hubungi server Db2.
   e. Prakompilasi, kompilasi, dan ikat program COBOL.
   f. Simpan produk jadi dalam ember S3 AWS CodeDeploy untuk digunakan.
5. CodePipeline dimulai CodeDeploy.
6. CodeDeploy mengkoordinasikan agennya, yang sudah diinstal di lingkungan runtime. Agen mengambil aplikasi dari Amazon S3 dan menginstalnya berdasarkan instruksi di. `appspec.yml`

Untuk menjaga hal-hal sederhana dan fokus pada build, instruksi dalam pola ini mencakup langkah 1 hingga 4 tetapi tidak termasuk penerapan program COBOL Db2.

Otomatisasi dan skala

Untuk mempermudah, pola ini menjelaskan cara menyediakan sumber daya secara manual. Namun, ada banyak opsi otomatisasi yang tersedia, seperti,, dan HashiCorp Terraform CloudFormation AWS Cloud Development Kit (AWS CDK), yang mengotomatiskan tugas-tugas ini. Untuk informasi lebih lanjut, lihat CloudFormationdan AWS CDKdokumentasi.

Alat

Layanan AWS

- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeDeploymengotomatiskan penerapan ke Amazon EC2 atau instans, fungsi AWS Lambda , atau layanan Amazon Elastic Container Service (Amazon ECS) lokal.
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS Mainframe Modernizationmenyediakan alat dan sumber daya untuk membantu Anda merencanakan dan mengimplementasikan migrasi dan modernisasi dari mainframe ke lingkungan runtime AWS terkelola.

Alat lainnya

- Gambar Amazon ECR untuk alat AWS Mainframe Modernization Replatform. Untuk mengkompilasi aplikasi COBOL, Anda harus memulai CodeBuild dengan menggunakan image Amazon Elastic Container Registry (Amazon ECR) Registry ECR) yang berisi alat Replatform: AWS Mainframe Modernization

  ```
  673918848628.dkr.ecr.<your-region>.amazonaws.com/m2-enterprise-build-tools:9.0.7.R1
  ```

  Untuk informasi lebih lanjut tentang gambar ECR yang tersedia, lihat tutorial di Panduan AWS Mainframe Modernization Pengguna.

- Perangkat lunak IBM Data Server Client sangat penting untuk pra-kompilasi dan pengikatan program COBOL Db2 di. CodeBuild Ini bertindak sebagai jembatan antara kompiler COBOL dan Db2.

Praktik terbaik

- Tidak setiap program COBOL bergantung pada Db2 sebagai lapisan persistensi datanya. Pastikan bahwa arahan kompilasi untuk mengakses Db2 diterapkan hanya untuk program COBOL yang dirancang khusus untuk berinteraksi dengan Db2. Menerapkan logika untuk membedakan antara program COBOL Db2 dan program COBOL yang tidak menggunakan Db2.
- Kami menyarankan Anda menghindari kompilasi program yang belum dimodifikasi. Menerapkan proses untuk mengidentifikasi program mana yang memerlukan kompilasi.

Epik

Buat infrastruktur cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 untuk meng-host IBM Data Server Client dan artefak pipeline.	Anda perlu menyiapkan bucket S3 untuk (a) mengunggah Klien Server Data IBM, (b) menyimpan kode Anda dari repositori, dan (c) menyimpan hasil proses pembuatan. 1. Masuk ke Konsol Manajemen AWS, dan buka konsol Amazon S3. 2. Pilih bucket S3 yang sudah ada atau buat bucket baru. Perhatikan Amazon Resource Name (ARN) bucket untuk penggunaan di masa mendatang.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk cara membuat bucket S3, lihat dokumentasi Amazon S3.	
Unggah Klien Server Data IBM ke bucket S3.	1. Di konsol Amazon S3, pilih ember untuk membukanya. 2. Pilih Buat folder, tentukan namanya sebagai klien, lalu pilih Buat folder. 3. Buka folder klien, pilih Unggah, Tambahkan file. 4. Pilih file Klien Server Data IBM yang sebelumnya Anda unduh dari situs web IBM ke sistem file lokal Anda. Nama file harus mirip dengan `v11.5.8_linuxx64_client.tar.gz` atau `v11.5.9_linuxx64_client.tar.gz` . 5. Pilih Buka, Unggah dan tunggu hingga unggahan selesai. 6. Pada tab File dan folder, pilih Data Server Client dan catat URI S3-nya.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AWS Secrets Manager rahasia untuk kredensi Db2 Anda.	Untuk membuat rahasia untuk menyimpan DB2 kredensil Anda dengan aman: 1. Di konsol Secrets Manager, pilih Simpan rahasia baru. 2. Di panel Pilih tipe rahasia, pilih Jenis rahasia lain dan Plaintext. 3. Dalam kotak Plaintext, ketik kredenal Db2 Anda dengan menggunakan struktur JSON berikut. `{ "username": "<your-db2-user-name>", "password": "<your-db2-password>", "db2node": "db2dev", "db2host": "<your-db2-hostname-or-IP>", "db2port": <your-db2-port>, "db2name": "<your-db2-location>", "qualifier": "<your-db2-qualifier>" }`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pilih Berikutnya dan beri nama rahasia sepertidev-db2-cred . 5. Pilih Berikutnya, Berikutnya, dan Simpan. Untuk informasi selengkap nya tentang membuat rahasia, lihat dokumentasi Secrets Manager.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa Db2 dapat diakses dari subnet VPC.	AWS CodeBuild membutuhk an koneksi ke server Db2 sehingga Klien Server Data dapat melakukan operasi prakompilasi dan mengikat. Pastikan itu CodeBuild dapat mencapai server Db2 melalui koneksi yang aman. 1. Buka konsol Amazon VPC. 2. Pada panel navigasi, pilih Subnet dan tuliskan dan subnet pribadi di mana CodeBuild akan berfungsi. IDs IPv4 CIDRs 3. Perbarui pengaturan kontrol akses jaringan saat ini untuk sistem Db2 Anda dengan memperkenalkan aturan masuk. Aturan ini harus mengaktifkan akses TCP khusus ke port Db2 secara eksklusif dari subnet CIDRs yang terkait dengan proyek Anda. CodeBuild	Administrator jaringan, AWS Umum

Buat artefak aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aset COBOL Db2.	1. Jika Anda ingin menggunak an contoh COBOL Db2 sederhana, simpan kode sumber berikut sebagai.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`CDB2SMP.cbl` Atau, Anda dapat mengganti contoh ini dengan program yang sudah Anda miliki. `IDENTIFICATION DIVISION. PROGRAM-ID. CDB2SMP. DATA DIVISION. WORKING-STORAGE SECTION. 01 WS-NAME PIC X(100). PROCEDURE DIVISION. EXEC SQL SELECT NAME INTO :WS-NAME FROM SYSIBM.SYSTABLES END-EXEC GOBACK.` 2. Komit perubahan dan dorong file ke repositori Anda.	
Buat file `buildspec.yml` .	1. Buat `buildspec.yml` file berdasarkan contoh yang disediakan di bagian Informasi tambahan. 2. Komit perubahan dan dorong file ke repositori Anda.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect repositori Anda ke. CodePipeline	1. Buka konsol Alat AWS Pengembang. 2. Di panel navigasi, pilih Pengaturan, Koneksi. 3. Ikuti petunjuk dalam dokumentasi konsol Alat Pengembang untuk penyedia sumber pilihan Anda. Anda akan memerlukan Nama Sumber Daya Amazon (ARN) untuk koneksi saat Anda membuat kebijakan AWS Identity and Access Management (IAM) untuk langkah CodePipeline selanjutnya.	AWS DevOps

Mengatur Konfigurasi Izin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM untuk CodeBuild.	CodeBuild Proyek ini memerlukan akses ke beberapa sumber daya, termasuk Secrets Manager dan Amazon S3. Untuk mengatur izin yang diperlukan: 1. Buka konsol IAM.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di panel navigasi, pilih Kebijakan, Buat Kebijakan , lalu pilih CodeBuild layanan. 3. Alihkan pemformatan dari Visual ke JSON, dan salin CodeBuild kebijakan yang disediakan di bagian Informasi tambahan ke bidang Editor kebijakan. 4. Beri nama dan simpan kebijakan ini untuk referensi future di langkah berikutny a. Untuk informasi selengkapnya tentang membuat kebijakan IAM, lihat dokumentasi IAM.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk CodeBuild.	Agar kebijakan keamanan tersedia CodeBuild, Anda perlu mengonfigurasi peran IAM. Untuk membuat peran ini: 1. Di konsol IAM, di panel navigasi, pilih Peran, Buat Peran. 3. Untuk jenis entitas Tepercaya, pertahankan Layanan AWSpengaturan default. 4. Untuk kasus penggunaan, pilih CodeBuild layanan, dan kemudian pilih Berikutnya. 4. Dalam daftar kebijakan IAM yang tersedia, cari kebijakan yang Anda buat CodeBuild , lalu pilih Berikutnya untuk melampirkannya ke peran. 5. Tentukan nama untuk peran, dan pilih Buat peran untuk menyimpannya untuk referensi di masa mendatang CodeBuild. Untuk informasi selengkapnya tentang membuat peran IAM untuk sebuah Layanan AWS, lihat dokumentasi IAM.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM untuk CodePipeline.	AWS CodePipeline Pipeline memerlukan akses ke beberapa sumber daya, termasuk repositori kode Anda dan Amazon S3. Ulangi langkah-langkah yang diberikan sebelumnya CodeBuild untuk membuat kebijakan IAM CodePipel ine (pada langkah 2, pilih CodePipelinesebagai ganti CodeBuild).	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat peran IAM untuk CodePipeline.	Agar kebijakan keamanan tersedia CodePipeline, Anda perlu mengonfigurasi peran IAM.
	
	Untuk membuat peran ini:
	
	1. Di konsol IAM, pilih Peran, Buat Peran.
	2. Untuk jenis entitas Tepercaya, pilih Kebijakan kepercayaan khusus.
	
	Kebijakan dengan `Principal` elemen kosong akan ditampilkan.
	3. Di `Principal` telepon, di antara kawat gigi, tambahkan:
	
	`"Service": "codepipeline.amazonaws.com"`
	
	Kebijakan kepercayaan akan terlihat seperti ini:
	
	`{ "Version": "2012-10-17", "Statement": [ { "Sid": "Statement1", "Effect": "Allow", "Principal": {`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"Service": "codepipeline.amazonaws.com" }, "Action": "sts:AssumeRole" }] }</pre> 4. Pilih Berikutnya. 5. Dalam daftar kebijakan IAM yang tersedia, cari kebijakan yang Anda buat CodePipeline, lalu pilih Berikutnya untuk melampirkannya ke peran. 6. Tentukan nama untuk peran, dan pilih Buat peran untuk menyimpannya untuk referensi di masa mendatang CodePipeline.	

Kompilasi dan ikat program COBOL Db2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodePipeline pipeline dan CodeBuild proyek.	Untuk membuat CodePipeline pipeline dan CodeBuild proyek yang mengkompilasi dan mengikat program COBOL Db2: 1. Buka CodePipeline konsol, dan pilih Create Pipeline, Build custom pipeline.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Tentukan nama untuk pipeline. 3. Untuk peran Layanan, pilih Peran layanan yang ada dan pilih tentukan ARN untuk peran IAM yang Anda buat. CodePipeline 4. Perluas Pengaturan lanjutan, pilih Lokasi khusus, pilih bucket S3 yang sebelumnya Anda buat, lalu pilih Berikutnya. 5. Untuk penyedia Sumber, pilih penyedia sumber pihak ketiga Anda, dan berikan informasi yang relevan untuk penyedia: a. Untuk Koneksi, pilih koneksi yang dibuat untuk penyedia sumber. b. Untuk nama Repositori, pilih repositori Anda. c. Untuk cabang Default, pilih cabang yang menyimpan program COBOL Anda dan`buildspec.yml` . d. Pilih Berikutnya. 6. Untuk penyedia Build, pilih Penyedia build lainnya, AWS CodeBuild. 7. Untuk nama Project, pilih Create project.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Konsol menampilkan CodeBuild jendela tempat Anda dapat membuat proyek build. Di jendela ini: a. Masukkan nama untuk proyek tersebut. b. Untuk gambar Lingkungan, pilih Gambar kustom. c. Untuk jenis Lingkungan, pilih Linux Container. d. Untuk akun ECR, pilih Akun ECR lainnya. e. Untuk URI repositori Amazon ECR, masukkan:. `673918848628.dkr.ecr.<your-region>.amazonaws.com/m2-enterprise-build-tools:8.0.9.R1` f. Untuk peran Layanan, pilih Peran layanan yang ada dan pilih peran yang Anda buat CodeBuild. g. Perluas bagian Konfigurasi tambahan, lalu pilih VPC, subnet pribadi, dan grup keamanan untuk proyek ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	h. Di bagian Buildspec, pilih Gunakan file buildspec. i. Di akhir jendela, pilih Lanjutkan keCodePipeline. CodeBuild Jendela akan ditutup sehingga Anda dapat kembali ke CodePipeline konsol. 8. Kembali ke CodePipeline konsol, pilih Berikutnya. 9. Di panel Tambahkan tahap penerapan, pilih Lewati tahap penerapan, dan konfirmasikan. 10. Tinjau parameter pipeline lalu pilih Create pipeline.	
Tinjau output.	Verifikasi keberhasilan build dengan meninjau log CodePipeline build.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa hasil di Db2.	Verifikasi versi paket pada tabel SYSPLAN.
	`select CAST(NAME AS VARCHAR(10)) as name, VALIDATE, LAST_BIND_TIME, LASTUSED, CAST(PKGVERSION AS VARCHAR(10)) as PKGVERSION from SYSIBM.SYSPLAN where NAME = 'CDB2SMP' order by LAST_BIND_TIME desc`
	Versi harus cocok dengan ID CodeBuild build, yang ada `CDB2SMP` dalam contoh kita:
	`NAME VALIDATE LAST_BIND_TIME LASTUSED PKGVERSION ---------- -------- ---------- ---------- -------- ---------- CDB2SMP B 2024-05-18-11.53.11.503738 01/01/0001 19`	

Pemecahan masalah

Isu	Solusi
Terkadang, AWS konsol beralih Wilayah saat Anda berpindah antar layanan.	Pastikan untuk memverifikasi yang dipilih Wilayah AWS setiap kali Anda beralih antar layanan.

Isu	Solusi
	Wilayah AWS Pemilih berada di sudut kanan atas jendela konsol.
Mungkin sulit untuk mengidentifikasi masalah konektivitas Db2 dari. CodeBuild	Untuk memecahkan masalah konektivitas, tambahkan perintah DB2 connect berikut ke file. `buildspec.yml` Penambahan ini membantu Anda men-debug dan menyelesa ikan masalah konektivitas. `db2 connect to $DB_NAME user $DB2USER using $DB2PASS`
Terkadang, panel peran di konsol IAM tidak langsung menampilkan kebijakan IAM yang Anda buat.	Jika Anda mengalami penundaan, segarkan layar untuk menampilkan informasi terbaru.

Sumber daya terkait

Dokumentasi IBM

- IBM Data Server Client dan tipe driver
- Unduh IBM Data Server Client dan jenis driver

AWS dokumentasi

- Panduan Pengguna Amazon S3
- AWS CodeBuild Panduan Pengguna
- AWS Mainframe Modernization Panduan Pengguna
- AWS Secrets Manager Panduan Pengguna
- AWS CodePipeline Panduan Pengguna
- AWS CodeDeploy Panduan Pengguna

Informasi tambahan

CodeBuild kebijakan

Ganti placeholder<RegionID>,,<AccountID>, <SubnetARN><BucketARN>, dan <DB2CredSecretARN> dengan nilai-nilai Anda.

```
{"Version": "2012-10-17",
    "Statement": [
        {"Action": "ecr:GetAuthorizationToken", "Effect": "Allow", "Resource": "*" },
        {"Action": ["ecr:GetDownloadUrlForLayer", "ecr:BatchGetImage",
                    "ecr:BatchCheckLayerAvailability"],
         "Effect": "Allow",
         "Resource": "arn:aws:ecr:*:673918848628:repository/m2-enterprise-build-
tools"},
        {"Action": "s3:PutObject", "Effect": "Allow", "Resource": "arn:aws:s3:::aws-m2-
repo-*/*"},
        {"Action": ["logs:PutLogEvents", "logs:CreateLogStream",
 "logs:CreateLogGroup"],
         "Effect": "Allow", "Resource": "arn:aws:logs:<RegionId>:<AccountId>:*"},
        {"Action": ["ec2:DescribeVpcs", "ec2:DescribeSubnets",
                    "ec2:DescribeSecurityGroups", "ec2:DescribeNetworkInterfaces",
                    "ec2:DescribeDhcpOptions", "ec2:DeleteNetworkInterface",
                    "ec2:CreateNetworkInterface"],
         "Effect": "Allow", "Resource": "*"},
        {"Action": "ec2:CreateNetworkInterfacePermission",
         "Effect": "Allow", "Resource": ["<SubnetARN>"]},
        {"Action": "s3:*", "Effect": "Allow", "Resource": ["<BucketARN>/
*","<BucketARN>"]},
        {"Action": "secretsmanager:GetSecretValue",
         "Effect": "Allow", "Resource": "<DB2CredSecretARN>"}
    ]
}
```

CodePipeline kebijakan

Ganti placeholder <BucketARN> dan <ConnectionARN> dengan nilai-nilai Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {"Action": ["s3:List*", "s3:GetObjectVersion", "s3:GetObject",
 "s3:GetBucketVersioning" ],
```

```
        "Effect": "Allow",
        "Resource": ["<BucketARN>/*", "<BucketARN>"]},
        {"Action": ["codebuild:StartBuild", "codebuild:BatchGetBuilds"],
         "Effect": "Allow", "Resource": "*"},
        {"Action": ["codestar-connections:UseConnection"],
         "Effect": "Allow", "Resource": "<ConnectionARN>"}
        ]
}
```

`buildspec.yml`

Ganti <your-bucket-name> placeholder dengan nama bucket S3 Anda yang sebenarnya.

```
version: 0.2
phases:
  pre_build:
    commands:
      - /var/microfocuslicensing/bin/mfcesd -no > /var/microfocuslicensing/logs/
mfcesd_startup.log 2>&1 &
      - |
        mkdir $CODEBUILD_SRC_DIR/db2client
        aws s3 cp s3://<your-bucket-name>/v11.5.8_linuxx64_client.tar.gz
 $CODEBUILD_SRC_DIR/db2client/ >> /dev/null 2>&1
        tar -xf $CODEBUILD_SRC_DIR/db2client/v11.5.8_linuxx64_client.tar.gz -C
 $CODEBUILD_SRC_DIR/db2client/
        cd $CODEBUILD_SRC_DIR/db2client/
        ./client/db2_install -f sysreq -y -b /opt/ibm/db2/V11.5 >> /dev/null 2>&1

        useradd db2cli
        /opt/ibm/db2/V11.5/instance/db2icrt -s client -u db2cli db2cli
        DB2CRED=$(aws secretsmanager get-secret-value --secret-id dev-db2-cred | jq -r
 '.SecretString | fromjson')
        read -r DB2USER DB2PASS DB_NODE DB_HOST DB_PORT DB_NAME DB_QUAL <<<$(echo
 $DB2CRED | jq -r
 '.username, .password, .db2node, .db2host, .db2port, .db2name, .qualifier')
        . /home/db2cli/sqllib/db2profile
        db2 catalog tcpip node $DB_NODE remote $DB_HOST server $DB_PORT
        db2 catalog db $DB_NAME as $DB_NAME at node $DB_NODE authentication server
  build:
    commands:
      - |
        revision=$CODEBUILD_SRC_DIR/loadlib
        mkdir -p $revision; cd $revision
        . /opt/microfocus/EnterpriseDeveloper/bin/cobsetenv
```

```
        cob -zU $CODEBUILD_SRC_DIR/CDB2SMP.cbl -C "DB2(DB==${DB_NAME} PASS==${DB2USER}.
${DB2PASS} VERSION==${CODEBUILD_BUILD_NUMBER} COLLECTION==DB2AWSDB"
artifacts:
  files:
    - "**/*"
  base-directory: $revision
```

Bangun PAC Server Perusahaan Fokus Mikro dengan Amazon EC2 Auto Scaling dan Systems Manager

Kevin Yung dan Krithika Palani Selvam, Amazon Web Services

Peter Woods, Tidak Ada

Abraham Rondon, Fokus Mikro

Ringkasan

Pola ini memperkenalkan arsitektur skalabel untuk aplikasi mainframe menggunakan Micro Focus Enterprise Server di Scale-Out Performance and Availability Cluster (PAC) dan grup EC2 Auto Scaling Amazon Elastic Compute Cloud (Amazon) di Amazon Web Services ().AWS Solusinya sepenuhnya otomatis dengan AWS Systems Manager dan kait siklus hidup Amazon EC2 Auto Scaling. Dengan menggunakan pola ini, Anda dapat mengatur aplikasi online dan batch mainframe Anda untuk mencapai ketahanan tinggi dengan secara otomatis menskalakan masuk dan keluar berdasarkan permintaan kapasitas Anda.

> **Note**
>
> Pola ini diuji dengan Micro Focus Enterprise Server versi 6.0. Untuk versi 8, lihat Mengatur Waktu Proses Fokus Mikro (di Amazon EC2).

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Perangkat lunak dan lisensi Micro Focus Enterprise Server. Untuk detailnya, hubungi penjualan Micro Focus.
- Pemahaman tentang konsep membangun kembali dan memberikan aplikasi mainframe untuk dijalankan di Micro Focus Enterprise Server. Untuk ikhtisar tingkat tinggi, lihat Lembar Data Server Perusahaan Fokus Mikro.
- Pemahaman tentang konsep-konsep dalam Kinerja dan Ketersediaan Cluster Kinerja dan Ketersediaan Server Micro Focus Enterprise. Untuk informasi selengkapnya, lihat dokumentasi Micro Focus Enterprise Server.

- Pemahaman tentang konsep keseluruhan aplikasi mainframe DevOps dengan integrasi berkelanjutan (CI). Untuk pola AWS Prescriptive Guidance yang dikembangkan oleh AWS dan Micro Focus, lihat Modernisasi mainframe: DevOps dengan Micro Focus. AWS

> **Note**
>
> Pola ini diuji dengan Micro Focus Enterprise Server versi 6. Untuk versi 8, lihat Mengatur Waktu Proses Fokus Mikro (di Amazon EC2).

Batasan

- Untuk daftar platform yang didukung oleh Micro Focus Enterprise Server, lihat Lembar Data Server Perusahaan Fokus Mikro.
- Skrip dan tes yang digunakan dalam pola ini didasarkan pada Amazon EC2 Windows Server 2019; versi Windows Server dan sistem operasi lainnya tidak diuji untuk pola ini.
- Pola ini didasarkan pada Micro Focus Enterprise Server 6.0 untuk Windows; rilis sebelumnya atau yang lebih baru tidak diuji dalam pengembangan pola ini.

Versi produk

- Server Perusahaan Fokus Mikro 6.0
- Windows Server 2019

Arsitektur

Dalam lingkungan mainframe konvensional, Anda harus menyediakan perangkat keras untuk meng-host aplikasi dan data perusahaan Anda. Untuk memenuhi dan memenuhi lonjakan permintaan musiman, bulanan, triwulanan, atau bahkan belum pernah terjadi sebelumnya atau tidak terduga, pengguna mainframe harus meningkatkan skala dengan membeli penyimpanan tambahan dan kapasitas komputasi. Meningkatkan jumlah penyimpanan dan sumber daya kapasitas komputasi meningkatkan kinerja secara keseluruhan, tetapi penskalaannya tidak linier.

Ini tidak terjadi ketika Anda mulai mengadopsi model konsumsi sesuai permintaan di AWS dengan menggunakan Amazon Auto EC2 Scaling dan Micro Focus Enterprise Server. Bagian berikut menjelaskan secara rinci cara membuat arsitektur aplikasi mainframe yang sepenuhnya otomatis

dan dapat diskalakan menggunakan Micro Focus Enterprise Server Scale-Out Performance and Availability Cluster (PAC) dengan grup Amazon Auto Scaling. EC2

Arsitektur penskalaan otomatis Micro Focus Enterprise Server

Pertama, penting untuk memahami konsep dasar Micro Focus Enterprise Server. Lingkungan ini menyediakan lingkungan penyebaran x86 yang kompatibel dengan mainframe untuk aplikasi yang secara tradisional berjalan di mainframe IBM. Ini memberikan proses online dan batch dan lingkungan transaksi yang mendukung hal-hal berikut:

- IBM COBOL
- IBM PL/I
- Lowongan kerja batch IBM JCL
- Transaksi IBM CICS dan IMS TM
- Layanan web
- Utilitas batch umum, termasuk SORT

Micro Focus Enterprise Server memungkinkan aplikasi mainframe berjalan dengan sedikit perubahan. Beban kerja mainframe yang ada dapat dipindahkan ke platform x86 dan dimodernisasi untuk memanfaatkan ekstensi asli AWS Cloud untuk ekspansi cepat ke pasar atau geografi baru.

Modernisasi Mainframe pola AWS Prescriptive Guidance: di DevOps AWS dengan Micro Focus memperkenalkan arsitektur untuk mempercepat pengembangan dan pengujian aplikasi mainframe di AWS menggunakan Micro Focus Enterprise Developer dan Enterprise Test Server dengan AWS dan AWS. CodePipeline CodeBuild Pola ini berfokus pada penerapan aplikasi mainframe ke lingkungan produksi AWS untuk mencapai ketersediaan dan ketahanan yang tinggi.

Dalam lingkungan produksi mainframe, Anda mungkin telah menyiapkan IBM Parallel Sysplex di mainframe untuk mencapai kinerja tinggi dan ketersediaan tinggi. Untuk membuat arsitektur scale-out yang mirip dengan Sysplex, Micro Focus memperkenalkan Performance and Availability Cluster (PAC) ke Enterprise Server. PACs mendukung penerapan aplikasi mainframe ke beberapa wilayah Server Perusahaan yang dikelola sebagai satu gambar dan diskalakan dalam instance Amazon. EC2 PACs juga mendukung kinerja aplikasi yang dapat diprediksi dan throughput sistem sesuai permintaan.

Dalam PAC, beberapa instance Enterprise Server bekerja sama sebagai entitas logis tunggal. Kegagalan satu instance Server Perusahaan, oleh karena itu, tidak akan mengganggu kelangsungan bisnis karena kapasitas dibagi dengan wilayah lain sementara instance baru secara otomatis

dimulai menggunakan fungsionalitas standar industri seperti grup Amazon Auto Scaling EC2 . Ini menghilangkan satu titik kegagalan, meningkatkan ketahanan terhadap masalah perangkat keras, jaringan, dan aplikasi. Instans Enterprise Server yang diskalakan dapat dioperasikan dan dikelola dengan menggunakan Enterprise Server Common Web Administration (ESCWA) APIs, menyederhanakan pemeliharaan operasional dan kemudahan servis Server Perusahaan.

> **Note**
>
> Micro Focus merekomendasikan bahwa Performance and Availability Cluster (PAC) harus terdiri dari setidaknya tiga wilayah Server Perusahaan sehingga ketersediaan tidak terganggu jika wilayah Server Perusahaan gagal atau memerlukan pemeliharaan.

Konfigurasi PAC memerlukan layanan manajemen basis data relasional yang didukung (RDBMS) untuk mengelola database wilayah, database lintas wilayah, dan database penyimpanan data opsional. Database penyimpanan data harus digunakan untuk mengelola file Metode Akses Penyimpanan Virtual (VSAM) menggunakan dukungan Micro Focus Database File Handler untuk meningkatkan ketersediaan dan skalabilitas. Didukung RDBMSs termasuk yang berikut:

- Microsoft SQL Server 2009 R2 dan yang lebih baru
- PostgreSQL 10.x, termasuk Amazon Aurora PostgreSQL Edisi yang kompatibel
- DB2 10.4 dan kemudian

Untuk detail persyaratan RDBMS dan PAC yang didukung, lihat Server Perusahaan Fokus Mikro - Prasyarat dan Server Perusahaan Fokus Mikro - Konfigurasi PAC yang Disarankan.

Diagram berikut menunjukkan pengaturan arsitektur AWS khas untuk Micro Focus PAC.

	Komponen	Deskripsi
1	Server Perusahaan melakukan grup penskalaan otomatis	Siapkan grup penskalaan otomatis yang digunakan dengan instance Enterprise Server di PAC. Jumlah instans dapat ditingkatkan atau

		diprakarsai oleh CloudWatch alarm Amazon menggunakan metrik. CloudWatch
2	Enterprise Server ESCWA instance grup penskalaan otomatis	Siapkan grup penskalaan otomatis yang digunakan dengan Enterprise Server Common Web Administr ation (ESCWA). ESCWA menyediakan manajemen cluster. APIs Server ESCWA bertindak sebagai bidang kontrol untuk menambah atau menghapus Server Perusahaan dan memulai atau menghentikan wilayah Server Perusahaan di PAC selama peristiwa penskalaan otomatis instance Server Perusahaan. Karena instans ESCWA hanya digunakan untuk manajemen PAC, pola lalu lintasnya dapat diprediksi, dan persyaratan kapasitas penskalaan otomatis yang diinginkan dapat diatur ke 1.
3	Instans Amazon Aurora dalam pengaturan Multi-AZ	Siapkan sistem manajemen basis data relasional (RDBMS) untuk meng-host file data pengguna dan sistem untuk dibagikan di seluruh instance Server Perusahaan.

4	Amazon ElastiCache (Redis OSS) contoh dan replika	Siapkan instance utama ElastiCache (Redis OSS) dan setidaknya satu replika untuk meng-host data pengguna dan bertindak sebagai repositor i scale-out (SOR) untuk instance Server Perusahaan. Anda dapat mengonfigurasi satu atau beberapa repositor i scale-out untuk menyimpan jenis data pengguna tertentu. Enterprise Server menggunak an database Redis NoSQL sebagai SOR, persyaratan untuk menjaga integritas PAC.
5	Penyeimbang Beban Jaringan	Siapkan penyeimbang beban, berikan nama host untuk aplikasi untuk terhubung ke layanan yang disediakan oleh instance Enterprise Server (misalnya, mengakses aplikasi melalui emulator 3270).

Komponen-komponen ini membentuk persyaratan minimum untuk cluster PAC Server Micro Focus Enterprise. Bagian selanjutnya mencakup otomatisasi manajemen cluster.

Menggunakan AWS Systems Manager Automation untuk penskalaan

Setelah klaster PAC digunakan di AWS, PAC dikelola melalui Enterprise Server Common Web Administration (ESCWA). APIs

Untuk mengotomatiskan tugas manajemen klaster selama peristiwa penskalaan otomatis, Anda dapat menggunakan runbook Otomasi Systems Manager dan Amazon Auto EC2 Scaling dengan Amazon. EventBridge Arsitektur otomatisasi ini ditunjukkan pada diagram berikut.

	Komponen	Deskripsi
1	Kait siklus hidup penskalaan otomatis	Siapkan kait siklus hidup penskalaan otomatis dan kirim notifikasi ke Amazon EventBridge saat instans baru diluncurkan dan instans yang ada dihentikan di grup penskalaan otomatis.
2	Amazon EventBridge	Siapkan EventBridge aturan Amazon untuk merutekan peristiwa penskalaan otomatis ke target runbook Automation Systems Manager.
3	Runbook otomatisasi	Siapkan runbook Systems Manager Automation untuk menjalankan PowerShell skrip Windows dan memanggil ESCWA untuk mengelola PAC APIs . Sebagai contoh, lihat bagian Informasi tambahan.
4	Instans Enterprise Server ESCWA dalam grup penskalaan otomatis	Siapkan instance Enterprise Server ESCWA dalam grup penskalaan otomatis. Instans ESCWA menyediakan APIs untuk mengelola PAC.

Alat

- Micro Focus Enterprise Server — Micro Focus Enterprise Server menyediakan lingkungan run untuk aplikasi yang dibuat dengan varian lingkungan pengembangan terintegrasi (IDE) dari Enterprise Developer.

- EC2 Auto Scaling Amazon — Amazon EC2 Auto Scaling membantu Anda memastikan bahwa Anda memiliki jumlah instans EC2 Amazon yang benar yang tersedia untuk menangani pemuatan aplikasi Anda. Anda membuat koleksi EC2 instance, yang disebut grup Auto Scaling, dan menentukan jumlah instans minimum dan maksimum.
- Amazon ElastiCache (Redis OSS) — Amazon ElastiCache adalah layanan web untuk menyiapkan, mengelola, dan menskalakan penyimpanan data dalam memori terdistribusi atau lingkungan cache di cloud. Layanan ini menyediakan solusi caching beperforma tinggi, dapat diskalakan, dan hemat biaya.
- Amazon RDS - Amazon Relational Database Service (Amazon RDS) adalah layanan web yang memudahkan pengaturan, pengoperasian, dan skala database relasional di AWS Cloud. Ini menyediakan efisiensi biaya, kapasitas resizable untuk database relasional dan mengelola tugas-tugas administrasi database umum.
- AWS Systems Manager — AWS Systems Manager adalah layanan AWS yang dapat Anda gunakan untuk melihat dan mengontrol infrastruktur di AWS. Menggunakan konsol Systems Manager, Anda dapat melihat data operasional dari beberapa layanan AWS dan mengotomatiskan tugas operasional di seluruh sumber daya AWS Anda. Systems Manager membantu Anda menjaga keamanan dan kepatuhan dengan memindai instans terkelola Anda dan melaporkan (atau mengambil tindakan korektif) setiap pelanggaran kebijakan yang terdeteksi.

Epik

Buat instans Amazon Aurora

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation template AWS untuk instans Amazon Aurora.	Gunakan cuplikan kode contoh AWS untuk membuat CloudFormation templat yang akan membuat instance Edisi yang kompatibel dengan Amazon Aurora PostgreSQL.	Arsitek awan
Terapkan CloudFormation tumpukan untuk membuat instance Amazon Aurora.	Gunakan CloudFormation template untuk membuat instance yang kompatibel dengan Aurora PostgreSQL	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang mengaktifkan replikasi Multi-AZ untuk beban kerja produksi.	
Konfigurasikan pengaturan koneksi database untuk Enterprise Server.	Ikuti petunjuk dalam dokumentasi Micro Focus untuk menyiapkan string koneksi dan konfigurasi database untuk Micro Focus Enterprise Server.	Insinyur data, DevOps insinyur

Buat ElastiCache klaster Amazon untuk instans Redis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation template untuk ElastiCache cluster Amazon untuk instance Redis.	Gunakan cuplikan kode contoh AWS untuk membuat CloudFormation template yang akan membuat ElastiCache klaster Amazon untuk instance Redis.	Arsitek awan
Terapkan CloudFormation tumpukan untuk membuat ElastiCache klaster Amazon untuk instance Redis.	Buat ElastiCache klaster Amazon untuk instans Redis yang mengaktifkan replikasi Multi-AZ untuk beban kerja produksi.	Arsitek awan
Konfigurasikan pengaturan koneksi PSOR Server Perusahaan.	Ikuti petunjuk dalam dokumentasi Micro Focus untuk menyiapkan konfigurasi koneksi PAC Scale-Out Repository (PSOR) untuk Micro Focus Enterprise Server PAC.	DevOps insinyur

Buat grup penskalaan otomatis Micro Focus Enterprise Server ESCWA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AMI Server Perusahaan Fokus Mikro.	Buat instance Amazon EC2 Windows Server dan instal biner Micro Focus Enterprise Server dalam EC2 instance. Buat Amazon Machine Image (AMI) dari EC2 instance. Untuk informasi selengkapnya, lihat dokumentasi instalasi Enterprise Server.	Arsitek awan
Buat CloudFormation template untuk Enterprise Server ESCWA.	Gunakan cuplikan kode contoh AWS untuk membuat templat untuk membuat tumpukan khusus Enterprise Server ESCWA dalam grup penskalaan otomatis.	Arsitek awan
Menerapkan CloudFormation tumpukan untuk membuat grup EC2 penskalaan Amazon untuk Enterprise Server ESCWA.	Gunakan CloudFormation template untuk menyebark an grup penskalaan otomatis dengan Micro Focus Enterpris e Server ESCWA AMI yang dibuat di cerita sebelumnya.	Arsitek awan

Membuat runbook AWS Systems Manager Automation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation template untuk runbook Systems Manager Automation.	Gunakan contoh cuplikan kode di bagian Informasi tambahan untuk membuat CloudFormation template yang akan membuat runbook	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Systems Manager Automatio n untuk mengotomatisasi pembuatan PAC, skala Server Perusahaan, dan skala Server Perusahaan.	
Menerapkan CloudFormation tumpukan yang berisi runbook Systems Manager Automation.	Gunakan CloudFormation template untuk menyebark an tumpukan yang berisi runbook Otomasi untuk pembuatan PAC, skala Server Perusahaan, dan skala Server Perusahaan.	Arsitek awan

Buat grup penskalaan otomatis untuk Micro Focus Enterprise Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation template untuk menyiapkan grup penskalaan otomatis untuk Micro Focus Enterprise Server.	Gunakan cuplikan kode contoh AWS untuk membuat CloudFormation template yang akan membuat grup penskalaan otomatis. Template ini akan menggunak an kembali AMI yang sama yang dibuat untuk instance Micro Focus Enterprise Server ESCWA. Kemudian gunakan cuplikan kode contoh AWS untuk membuat peristiwa siklus hidup penskalaan otomatis dan siapkan Amazon EventBridge untuk memfilter	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	peristiwa penskalaan dan penskalaan dalam templat yang sama. CloudFormation	
Terapkan CloudFormation tumpukan untuk grup penskalaan otomatis untuk Server Perusahaan Fokus Mikro.	Terapkan CloudFormation tumpukan yang berisi grup penskalaan otomatis untuk Server Perusahaan Fokus Mikro.	Arsitek awan

Sumber daya terkait

- Kluster Kinerja dan Ketersediaan Server Perusahaan Fokus Mikro (PAC)
- Kait EC2 siklus hidup Auto Scaling Amazon
- Menjalankan otomatisasi dengan pemicu menggunakan EventBridge

Informasi tambahan

Skenario berikut harus otomatis untuk penskalaan atau penskalaan cluster PAC.

Otomatisasi untuk memulai atau membuat ulang PAC

Pada awal cluster PAC, Enterprise Server memerlukan ESCWA untuk memanggil APIs untuk membuat konfigurasi PAC. Ini memulai dan menambahkan wilayah Server Perusahaan ke dalam PAC. Untuk membuat atau membuat ulang PAC, gunakan langkah-langkah berikut:

1. Konfigurasikan PAC Scale-Out Repository (PSOR) di ESCWA dengan nama yang diberikan.

```
POST /server/v1/config/groups/sors
```

2. Buat PAC dengan nama tertentu dan lampirkan PSOR ke sana.

```
POST /server/v1/config/groups/pacs
```

3. Konfigurasikan database wilayah dan basis data lintas wilayah jika ini adalah pertama kalinya Anda menyiapkan PAC.

> **Note**
>
> Langkah ini menggunakan kueri SQL dan alat dbhfhadmin baris perintah Micro Focus Enterprise Suite untuk membuat database dan mengimpor data awal.

4. Instal definisi PAC ke wilayah Server Perusahaan.

```
POST /server/v1/config/mfds
POST /native/v1/config/groups/pacs/${pac_uid}/install
```

5. Mulai wilayah Server Perusahaan di PAC.

```
POST /native/v1/regions/${host_ip}/${port}/${region_name}/start
```

Langkah-langkah sebelumnya dapat diimplementasikan dengan menggunakan PowerShell skrip Windows.

Langkah-langkah berikut menjelaskan cara membangun otomatisasi untuk membuat PAC dengan menggunakan kembali skrip Windows PowerShell .

1. Buat template EC2 peluncuran Amazon yang mengunduh atau membuat PowerShell skrip Windows sebagai bagian dari proses bootstrap. Misalnya, Anda dapat menggunakan data EC2 pengguna untuk mengunduh skrip dari bucket Amazon Simple Storage Service (Amazon S3).
2. Buat runbook AWS Systems Manager Automation untuk menjalankan skrip Windows PowerShell .
3. Kaitkan runbook ke instance ESCWA dengan menggunakan tag instance.
4. Buat grup penskalaan otomatis ESCWA dengan menggunakan template peluncuran.

Anda dapat menggunakan contoh CloudFormation cuplikan AWS berikut untuk membuat runbook Otomasi.

Contoh CloudFormation cuplikan untuk runbook Systems Manager Automation yang digunakan untuk pembuatan PAC

```
  PACInitDocument:
     Type: AWS::SSM::Document
     Properties:
       DocumentType: Command
```

```
      Content:
        schemaVersion: '2.2'
        description: Operation Runbook to create Enterprise Server PAC
        mainSteps:
        - action: aws:runPowerShellScript
          name: CreatePAC
          inputs:
            onFailure: Abort
            timeoutSeconds: "1200"
            runCommand:
            - |
              C:\Scripts\PAC-Init.ps1
  PacInitAutomation:
     Type: AWS::SSM::Document
     Properties:
       DocumentType: Automation
       Content:
         description: Prepare Micro Focus PAC Cluster via ESCWA Server
         schemaVersion: '0.3'
         assumeRole: !GetAtt SsmAssumeRole.Arn
         mainSteps:
           - name: RunPACInitDocument
             action: aws:runCommand
             timeoutSeconds: 300
             onFailure: Abort
             inputs:
               DocumentName: !Ref PACInitDocument
               Targets:
                 - Key: tag:Enterprise Server - ESCWA
                   Values:
                     - "true"
   PacInitDocumentAssociation:
     Type: AWS::SSM::Association
     Properties:
       DocumentVersion: "$LATEST"
       Name: !Ref PACInitDocument
       Targets:
         - Key: tag:Enterprise Server - ESCWA
           Values:
             - "true"
```

Untuk informasi selengkapnya, lihat Server Perusahaan Fokus Mikro - Mengkonfigurasi PAC.

Otomatisasi untuk penskalaan dengan instance Server Perusahaan baru

Ketika instance Enterprise Server diskalakan, wilayah Enterprise Server harus ditambahkan ke PAC. Langkah-langkah berikut menjelaskan cara memanggil ESCWA APIs dan menambahkan wilayah Server Perusahaan ke dalam PAC.

1. Instal definisi PAC ke wilayah Server Perusahaan.

```
POST '/server/v1/config/mfds'
POST /native/v1/config/groups/pacs/${pac_uid}/install
```

2. Hangat Mulai wilayah di PAC.

```
POST /native/v1/regions/${host_ip}/${port}/${region_name}/start
```

3. Tambahkan instance Enterprise Server ke penyeimbang beban dengan mengaitkan grup penskalaan otomatis ke penyeimbang beban.

Langkah-langkah sebelumnya dapat diimplementasikan dengan menggunakan PowerShell skrip Windows. Untuk informasi selengkapnya, lihat Server Perusahaan Fokus Mikro - Mengkonfigurasi PAC.

Langkah-langkah berikut dapat digunakan untuk membangun otomatisasi berbasis peristiwa untuk menambahkan instance Enterprise Server yang baru diluncurkan ke PAC dengan menggunakan kembali skrip Windows PowerShell .

1. Buat template EC2 peluncuran Amazon untuk instance Enterprise Server yang menyediakan Wilayah Server Perusahaan selama bootstrap. Misalnya, Anda dapat menggunakan perintah Micro Focus Enterprise Server mfds untuk mengimpor konfigurasi wilayah. Untuk detail lebih lanjut dan opsi yang tersedia untuk perintah ini, lihat Referensi Server Perusahaan.
2. Buat grup penskalaan otomatis Enterprise Server yang menggunakan template peluncuran yang dibuat pada langkah sebelumnya.
3. Buat runbook Systems Manager Automation untuk menjalankan skrip Windows PowerShell .
4. Kaitkan runbook ke instance ESCWA dengan menggunakan tag instance.
5. Buat EventBridge aturan Amazon untuk memfilter peristiwa EC2 Instance Launch Successful untuk grup penskalaan otomatis Enterprise Server, dan buat target untuk menggunakan runbook Otomasi.

Anda dapat menggunakan contoh CloudFormation cuplikan berikut untuk membuat runbook Otomasi dan aturannya. EventBridge

Contoh CloudFormation cuplikan untuk Systems Manager yang digunakan untuk menskalakan instance Enterprise Server

```
  ScaleOutDocument:
     Type: AWS::SSM::Document
     Properties:
       DocumentType: Command
       Content:
         schemaVersion: '2.2'
         description: Operation Runbook to Adding MFDS Server into an existing PAC
         parameters:
           MfdsPort:
             type: String
           InstanceIpAddress:
             type: String
             default: "Not-Available"
           InstanceId:
             type: String
             default: "Not-Available"
         mainSteps:
         - action: aws:runPowerShellScript
           name: Add_MFDS
           inputs:
             onFailure: Abort
             timeoutSeconds: "300"
             runCommand:
             - |
               $ip = "{{InstanceIpAddress}}"
               if ( ${ip} -eq "Not-Available" ) {
                 $ip = aws ec2 describe-instances --instance-id {{InstanceId}} --output
text --query "Reservations[0].Instances[0].PrivateIpAddress"
               }
               C:\Scripts\Scale-Out.ps1 -host_ip ${ip} -port {{MfdsPort}}

   PacScaleOutAutomation:
     Type: AWS::SSM::Document
     Properties:
       DocumentType: Automation
       Content:
         parameters:
```

```
          MfdsPort:
            type: String
          InstanceIpAddress:
            type: String
            default: "Not-Available"
          InstanceId:
            type: String
            default: "Not-Available"
        description: Scale Out 1 New Server in Micro Focus PAC Cluster via ESCWA
Server
        schemaVersion: '0.3'
        assumeRole: !GetAtt SsmAssumeRole.Arn
        mainSteps:
          - name: RunScaleOutCommand
            action: aws:runCommand
            timeoutSeconds: 300
            onFailure: Abort
            inputs:
              DocumentName: !Ref ScaleOutDocument
              Parameters:
                InstanceIpAddress: "{{InstanceIpAddress}}"
                InstanceId: "{{InstanceId}}"
                MfdsPort: "{{MfdsPort}}"
              Targets:
                - Key: tag:Enterprise Server - ESCWA
                  Values:
                    - "true"
```

Otomatisasi untuk penskalaan dalam instance Server Perusahaan

Mirip dengan scaling out, ketika instance Enterprise Server diskalakan, event EC2 Instance-terminate Lifecycle Action dimulai, dan proses serta panggilan API berikut diperlukan untuk menghapus instance Micro Focus Enterprise Server dari PAC.

1. Hentikan wilayah dalam instance Enterprise Server yang mengakhiri.

```
POST "/native/v1/regions/${host_ip}/${port}/${region_name}/stop"
```

2. Hapus Enterprise Server Instance dari PAC.

```
DELETE "/server/v1/config/mfds/${uid}"
```

3. Kirim sinyal untuk melanjutkan penghentian instance Enterprise Server.

Langkah-langkah sebelumnya dapat diimplementasikan dalam PowerShell skrip Windows. Untuk detail tambahan dari proses ini, lihat dokumen Micro Focus Enterprise Server - Mengelola PAC.

Langkah-langkah berikut menjelaskan cara membangun otomatisasi berbasis peristiwa untuk menghentikan instance Enterprise Server dari PAC dengan menggunakan kembali skrip Windows. PowerShell

1. Buat runbook Systems Manager Automation untuk menjalankan skrip Windows PowerShell .
2. Kaitkan runbook ke instance ESCWA dengan menggunakan tag instance.
3. Buat hook siklus hidup grup penskalaan otomatis untuk EC2 penghentian misalnya.
4. Buat EventBridge aturan Amazon untuk memfilter peristiwa Tindakan Siklus Hidup EC2 Penghentian Instans untuk grup penskalaan otomatis Server Perusahaan, dan buat target untuk menggunakan runbook Otomasi.

Anda dapat menggunakan contoh CloudFormation template berikut untuk membuat runbook, lifecycle hook, dan aturan Systems Manager Automation. EventBridge

Contoh CloudFormation cuplikan untuk runbook Systems Manager Automation yang digunakan untuk penskalaan dalam instance Server Perusahaan

```
ScaleInDocument:
   Type: AWS::SSM::Document
   Properties:
     DocumentType: Command
     Content:
       schemaVersion: '2.2'
       description: Operation Runbook to Remove MFDS Server from PAC
       parameters:
         MfdsPort:
           type: String
         InstanceIpAddress:
           type: String
           default: "Not-Available"
         InstanceId:
           type: String
           default: "Not-Available"
       mainSteps:
       - action: aws:runPowerShellScript
         name: Remove_MFDS
         inputs:
```

```
            onFailure: Abort
            runCommand:
            - |
              $ip = "{{InstanceIpAddress}}"
              if ( ${ip} -eq "Not-Available" ) {
                $ip = aws ec2 describe-instances --instance-id {{InstanceId}} --output
text --query "Reservations[0].Instances[0].PrivateIpAddress"
              }
              C:\Scripts\Scale-In.ps1 -host_ip ${ip} -port {{MfdsPort}}

  PacScaleInAutomation:
    Type: AWS::SSM::Document
    Properties:
      DocumentType: Automation
      Content:
        parameters:
          MfdsPort:
            type: String
          InstanceIpAddress:
            type: String
            default: "Not-Available"
          InstanceId:
            type: String
            default: "Not-Available"
        description: Scale In 1 New Server in Micro Focus PAC Cluster via ESCWA Server
        schemaVersion: '0.3'
        assumeRole: !GetAtt SsmAssumeRole.Arn
        mainSteps:
          - name: RunScaleInCommand
            action: aws:runCommand
            timeoutSeconds: "600"
            onFailure: Abort
            inputs:
              DocumentName: !Ref ScaleInDocument
              Parameters:
                InstanceIpAddress: "{{InstanceIpAddress}}"
                MfdsPort: "{{MfdsPort}}"
                InstanceId: "{{InstanceId}}"
              Targets:
                - Key: tag:Enterprise Server - ESCWA
                  Values:
                    - "true"
          - name: TerminateTheInstance
            action: aws:executeAwsApi
```

```
            inputs:
              Service: autoscaling
              Api: CompleteLifecycleAction
              AutoScalingGroupName: !Ref AutoScalingGroup
              InstanceId: "{{ InstanceId }}"
              LifecycleActionResult: CONTINUE
              LifecycleHookName: !Ref ScaleInLifeCycleHook
```

Otomatisasi untuk pemicu penskalaan EC2 otomatis Amazon

Proses menyiapkan kebijakan penskalaan untuk instance Enterprise Server memerlukan pemahaman tentang perilaku aplikasi. Dalam kebanyakan kasus, Anda dapat mengatur kebijakan penskalaan pelacakan target. Misalnya, Anda dapat menggunakan pemanfaatan CPU rata-rata sebagai CloudWatch metrik Amazon untuk mengatur kebijakan penskalaan otomatis. Untuk informasi selengkapnya, lihat Kebijakan penskalaan pelacakan target untuk EC2 Auto Scaling Amazon. Untuk aplikasi yang memiliki pola lalu lintas reguler, pertimbangkan untuk menggunakan kebijakan penskalaan prediktif. Untuk informasi selengkapnya, lihat Penskalaan prediktif untuk Auto EC2 Scaling Amazon.

Membangun penampil file mainframe tingkat lanjut di AWS Cloud

Boopathy GOPALSAMY dan Jeremiah O'Connor, Amazon Web Services

Ringkasan

Pola ini menyediakan contoh kode dan langkah-langkah untuk membantu Anda membangun alat canggih untuk menjelajahi dan meninjau file format tetap mainframe Anda dengan menggunakan layanan tanpa server AWS. Pola ini memberikan contoh cara mengonversi file input mainframe ke dokumen OpenSearch Layanan Amazon untuk menjelajah dan mencari. Alat penampil file dapat membantu Anda mencapai hal berikut:

- Pertahankan struktur dan tata letak file mainframe yang sama untuk konsistensi di lingkungan migrasi target AWS Anda (misalnya, Anda dapat mempertahankan tata letak yang sama untuk file dalam aplikasi batch yang mentransmisikan file ke pihak eksternal)
- Mempercepat pengembangan dan pengujian selama migrasi mainframe
- Mendukung kegiatan pemeliharaan setelah migrasi

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Virtual Private Cloud (VPC) dengan subnet yang dapat dijangkau oleh platform lama Anda
-
> **Note**
>
> File input dan copybook bahasa berorientasi bisnis umum (COBOL) yang sesuai (: Untuk contoh file input dan copybook COBOL, lihat di repositori. gfs-mainframe-solutions GitHub Untuk informasi selengkapnya tentang copybook COBOL, lihat Panduan Pemrograman COBOL Enterprise untuk z/OS 6.3 di situs web IBM.)

Batasan

- Penguraian copybook dibatasi tidak lebih dari dua level bersarang (TERJADI)

Arsitektur

Tumpukan teknologi sumber

- Masukan file dalam format FB (Tetap Diblokir)
- Tata letak copybook COBOL

Tumpukan teknologi target

- Amazon Athena
- OpenSearch Layanan Amazon
- Amazon Simple Storage Service (Amazon S3)
- AWS Lambda
- AWS Step Functions

Arsitektur target

Diagram berikut menunjukkan proses parsing dan konversi file input mainframe ke dokumen OpenSearch Service untuk browsing dan pencarian.

Diagram menunjukkan alur kerja berikut:

1. Pengguna admin atau aplikasi mendorong file input ke satu bucket S3 dan copybook COBOL ke bucket S3 lainnya.
2.
> **Note**
>
> Bucket S3 dengan file input memanggil fungsi Lambda yang memulai alur kerja Step Functions tanpa server. : Penggunaan pemicu peristiwa S3 dan fungsi Lambda untuk menggerakkan alur kerja Step Functions dalam pola ini adalah opsional. Contoh GitHub kode dalam pola ini tidak termasuk penggunaan layanan ini, tetapi Anda dapat menggunakan layanan ini berdasarkan kebutuhan Anda.

3. Alur kerja Step Functions mengkoordinasikan semua proses batch dari fungsi Lambda berikut:
   - `s3copybookparser.py`Fungsi mem-parsing tata letak copybook dan mengekstrak atribut bidang, tipe data, dan offset (diperlukan untuk pemrosesan data input).

- `s3toathena.py`Fungsi ini menciptakan tata letak tabel Athena. Athena mem-parsing data input yang diproses oleh `s3toathena.py` fungsi dan mengonversi data ke file CSV.
- `s3toelasticsearch.py`Fungsi ini menyerap file hasil dari bucket S3 dan mendorong file ke Service. OpenSearch

4. Pengguna mengakses OpenSearch Dasbor dengan OpenSearch Layanan untuk mengambil data dalam berbagai format tabel dan kolom dan kemudian menjalankan kueri terhadap data yang diindeks.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon Simple Storage Service (Amazon S3) menggunakan SQL standar.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan. Dalam pola ini, Anda menggunakan Lambda untuk mengimplementasikan logika inti, seperti mengurai file, mengonversi data, dan memuat data ke OpenSearch Layanan untuk akses file interaktif.
- Amazon OpenSearch Service adalah layanan terkelola yang membantu Anda menerapkan, mengoperasikan, dan menskalakan kluster OpenSearch Layanan di AWS Cloud. Dalam pola ini, Anda menggunakan OpenSearch Layanan untuk mengindeks file yang dikonversi dan menyediakan kemampuan pencarian interaktif bagi pengguna.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan fungsi Lambda dan layanan AWS lainnya untuk membangun aplikasi yang penting bagi bisnis. Dalam pola ini, Anda menggunakan Step Functions untuk mengatur fungsi Lambda.

Alat-alat lainnya

- GitHubadalah layanan hosting kode yang menyediakan alat kolaborasi dan kontrol versi.
- Python adalah bahasa pemrograman tingkat tinggi.

Kode

Kode untuk pola ini tersedia di GitHub gfs-mainframe-patternsrepositori.

Epik

Siapkan lingkungan target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket S3 untuk menyimpan copybook, file input, dan file output. Kami merekomendasikan struktur folder berikut untuk bucket S3 Anda: • `copybook/` • `input/` • `output/` • `query/` • `results/`	AWS Umum
Buat fungsi s3copyboo kparser.	1. Buat fungsi Lambda yang dipanggil `s3copyboo kparser` dan unggah kode sumber (`s3copybookparser.p y` dan`copybook.py`) dari repositori. GitHub	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Lampirkan kebijakan IAM `S3ReadOnly` ke fungsi Lambda.	
Buat fungsi s3toathena.	1. Buat fungsi Lambda yang dipanggil `s3toathena` dan unggah kode sumber (`s3toathena.py`) dari repositori. GitHub Konfigurasikan batas waktu Lambda menjadi> 60 detik. 2. Untuk menyediakan akses ke sumber daya yang diperlukan, lampirkan kebijakan IAM `AmazonAthenaFullAccess` dan `S3FullAccess` ke fungsi Lambda.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi s3toelasticsearch.	1. **Important** Tambahkan dependensi Python ke lingkungan Lambda Anda. : Untuk menggunakan `s3toelasticsearch` fungsi ini, Anda harus menambahkan dependensi Python karena fungsi Lambda menggunakan dependensi klien Elasticsearch Python (dan). `Elasticsearch==7.9.0 requests_aws4auth` 2. Buat fungsi Lambda yang dipanggil `s3toelasticsearch` dan unggah kode sumber (`s3toelasticsearch.py`) dari repositori. GitHub 3. Impor dependensi Python sebagai layer Lambda. 4. Lampirkan kebijakan IAM `S3ReadOnly` dan `AmazonOpenSearchSe`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`rviceReadOnlyAcces s` ke fungsi Lambda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kluster OpenSearch Layanan.	Buat cluster 1. Buat kluster OpenSearch Layanan. Saat Anda membuat cluster, lakukan hal berikut: • **Note** Buat pengguna utama dan kata sandi untuk cluster yang dapat Anda gunakan untuk masuk ke OpenSearch Dasbor. : Langkah ini tidak diperlukan jika Anda menggunakan otentikasi melalui Amazon Cognito. • Pilih kontrol akses berbutir halus. Ini memberi Anda cara tambahan untuk mengontrol akses ke data Anda di OpenSearch Layanan. 2. Salin URL domain dan teruskan sebagai variabel lingkungan	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	'HOST' ke fungsi Lambda. `s3toelasticsearch` Berikan akses ke peran IAM Untuk memberikan akses berbutir halus ke peran IAM fungsi Lambda (`arn:aws:iam::**:role/service-role/s3toelasticsearch-role-**`, lakukan hal berikut: 1. Masuk ke OpenSearch Dasbor sebagai pengguna utama. 2. Pilih tab Keamanan, lalu pilih Peran, all_access, Map user, Backend role. 3. Tambahkan Nama Sumber Daya Amazon (ARN) dari peran IAM fungsi Lambda, lalu pilih Simpan. Untuk informasi selengkapnya, lihat Memetakan peran ke pengguna dalam dokumentasi OpenSearch Layanan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Step Functions untuk orkestrasi.	1. Buat mesin status Step Functions dengan aliran standar. Definisi ini termasuk dalam GitHub repositori. 2. Dalam skrip JSON, ganti fungsi Lambda dengan ARNs fungsi ARNs from Lambda di lingkungan Anda.	AWS Umum

Menyebarkan dan menjalankan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah file input dan copybook ke bucket S3.	Unduh file sampel dari folder sampel GitHub repositori dan unggah file ke bucket S3 yang Anda buat sebelumnya. 1. Unggah `Mockedcopy.cpy` dan `acctix.cpy` ke `<S3_Bucket>/copybook` folder. 2. Unggah `Modedupdate.txt` dan `acctindex.cpy` sampel file input ke `<S3_Bucket>/input` folder.	AWS Umum
Memanggil Step Functions.	1. Masuk ke AWS Management Console dan buka konsol Step Functions .	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Di panel navigasi, pilih mesin Negara.
	3. Pilih mesin status Anda, lalu pilih Mulai eksekusi.
	4. Di kotak Input, masukkan copybook/file jalur berikut sebagai variabel JSON ke bucket S3, lalu pilih Mulai eksekusi.
	
	`{ "s3_copybook_bucket_name": "<BUCKET NAME>", "s3_copybook_bucket_key": "<COPYBOOK PATH>", "s3_source_bucket_name": "<BUCKET NAME", "s3_source_bucket_key": "INPUT FILE PATH" }`
	
	Contoh:
	
	`{ "s3_copybook_bucket_name": "fileaidtest", "s3_copybook_bucket_key": "copybook/acctix.cpy", "s3_source_bucket_name": "fileaidtest", "s3_source_bucket_key": "input/acctindex"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`}`	
Validasi eksekusi alur kerja di Step Functions.	Di konsol Step Functions, tinjau eksekusi alur kerja di inspektur Grafik. Status eksekusi dijalankan diberi kode warna untuk mewakili status eksekusi. Misalnya, biru menunjukkan Dalam Kemajuan, hijau menunjukkan Berhasil, dan merah menunjukkan Gagal. Anda juga dapat meninjau tabel di bagian Riwayat peristiwa eksekusi untuk informasi lebih rinci tentang peristiwa eksekusi. Untuk contoh eksekusi alur kerja grafis, lihat grafik Step Functions di bagian Informasi tambahan dari pola ini.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi log pengiriman di Amazon CloudWatch.	1. Masuk ke AWS Management Console dan buka CloudWatch konsol. 2. Di panel navigasi, perluas Log, lalu pilih Grup log. 3. Di kotak pencarian, cari grup log `s3toelasticsearch` fungsi. Untuk contoh log pengiriman yang berhasil, lihat log CloudWatch pengiriman di bagian Informasi tambahan dari pola ini.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi file yang diformat di OpenSearch Dasbor dan lakukan operasi file.	1. Masuk ke AWS Management Console. Di bawah Analytics, pilih OpenSearch Layanan Amazon. 2. Di panel navigasi, pilih Domain. 3. Di kotak pencarian, masukkan URL untuk domain Anda di OpenSearch Dasbor. 4. Pilih dasbor Anda, lalu masuk sebagai pengguna utama. 5. Jelajahi data yang diindeks dalam format tabel. 6. Bandingkan file input dengan file keluaran yang diformat (dokumen yang diindeks) di Dasbor. OpenSearch Tampilan dasbor menunjukkan header kolom yang ditambahkan untuk file yang diformat. Konfirmasikan bahwa data sumber dari file input yang belum diformat cocok dengan data target di tampilan dasbor. 7. Lakukan tindakan seperti pencarian (misalnya, dengan menggunakan nama bidang, nilai, atau	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ekspresi), filter, dan operasi DQL (Dashboard Query Language) terhadap file yang diindeks.	

Sumber daya terkait

Referensi

- Contoh Copybook COBOL (dokumentasi IBM)
- BMC Compuware File-aid (dokumentasi BMC)

Tutorial

- Tutorial: Menggunakan pemicu Amazon S3 untuk menjalankan fungsi Lambda (dokumentasi AWS Lambda)
- Bagaimana cara membuat alur kerja tanpa server dengan AWS Step Functions dan AWS Lambda (dokumentasi AWS)
- Menggunakan OpenSearch Dasbor dengan OpenSearch Layanan Amazon (dokumentasi AWS)

Informasi tambahan

Grafik Step Functions

Contoh berikut menunjukkan grafik Step Functions. Grafik menunjukkan status eksekusi run untuk fungsi Lambda yang digunakan dalam pola ini.

CloudWatch log pengiriman

Contoh berikut menunjukkan log pengiriman yang berhasil untuk eksekusi `s3toelasticsearch` eksekusi.

2022-08-10T 15:53:33. 033-05:00	Jumlah dokumen pemrosesan: 100

2022-08-10T 15:53:33. 171-05:00	[INFO] 2022-08-10T 20:53:33. 171 Z a1b2c3d4-5678-90ab-cdef- Post:443/_Bulk [status: 200 permintaan: 0.100s] EXAMPLE11111 https://search-essearch-3h4uqclifeqaj2vg4mphe7ffle.us-east-2.es.amazonaws.com
2022-08-10T 15:53:33. 172-05:00	Penulisan massal berhasil: 100 dokumen

Kontainerisasi beban kerja mainframe yang telah dimodernisasi oleh Blu Age

Richard Milner-Watts, Amazon Web Services

Ringkasan

Pola ini menyediakan lingkungan wadah sampel untuk menjalankan beban kerja mainframe yang telah dimodernisasi dengan menggunakan alat Blu Age. Blu Age mengubah beban kerja mainframe lama menjadi kode Java modern. Pola ini menyediakan pembungkus di sekitar aplikasi Java sehingga Anda dapat menjalankannya dengan menggunakan layanan orkestrasi kontainer seperti Amazon Elastic Container Service (Amazon ECS) Service Elastic Container (Amazon ECS) atau Amazon Elastic Kubernetes Service (Amazon EKS).

Untuk informasi selengkapnya tentang memodernisasi beban kerja Anda dengan menggunakan layanan Blu Age dan AWS, lihat publikasi AWS Prescriptive Guidance ini:

- Menjalankan beban kerja mainframe Blu Age yang dimodernisasi pada infrastruktur AWS tanpa server
- Menerapkan lingkungan untuk aplikasi Blu Age dalam kontainer dengan menggunakan Terraform

Untuk bantuan menggunakan Blu Age untuk memodernisasi beban kerja mainframe Anda, hubungi tim Blu Age dengan memilih Hubungi pakar kami di situs web Blu Age. Untuk bantuan dalam memigrasikan beban kerja modern Anda ke AWS, mengintegrasikannya dengan layanan AWS, dan memindahkannya ke produksi, hubungi manajer akun AWS Anda atau isi formulir AWS Professional Services.

Prasyarat dan batasan

Prasyarat

- Aplikasi Java modern yang dibuat oleh Blu Age. Untuk tujuan pengujian, pola ini menyediakan contoh aplikasi Java yang dapat Anda gunakan sebagai bukti konsep.
- Lingkungan Docker yang dapat Anda gunakan untuk membangun wadah.

Batasan

Bergantung pada platform orkestrasi kontainer yang Anda gunakan, sumber daya yang dapat disediakan untuk wadah (seperti CPU, RAM, dan penyimpanan) mungkin terbatas. Misalnya, jika Anda menggunakan Amazon ECS dengan AWS Fargate, lihat dokumentasi Amazon ECS untuk batasan dan pertimbangan.

Arsitektur

Tumpukan teknologi sumber

- Usia Blu
- Java

Tumpukan teknologi target

- Docker

Arsitektur target

Diagram berikut menunjukkan arsitektur aplikasi Blu Age dalam wadah Docker.

1. Titik masuk untuk wadah adalah skrip pembungkus. Skrip bash ini bertanggung jawab untuk mempersiapkan lingkungan runtime untuk aplikasi Blu Age dan memproses output.
2. Variabel lingkungan dalam wadah digunakan untuk mengonfigurasi variabel dalam skrip pembungkus, seperti nama bucket Amazon Simple Storage Service (Amazon S3) dan kredenal database Simple Storage Service (Amazon S3). Variabel lingkungan disediakan oleh AWS Secrets Manager atau Parameter Store, kemampuan AWS Systems Manager. Jika Anda menggunakan Amazon ECS sebagai layanan orkestrasi kontainer, Anda juga dapat melakukan hardcode variabel lingkungan dalam definisi tugas Amazon ECS.
3. Skrip pembungkus bertanggung jawab untuk menarik file input apa pun dari bucket S3 ke dalam wadah sebelum Anda menjalankan aplikasi Blu Age. AWS Command Line Interface (AWS CLI) diinstal di dalam container. Ini menyediakan mekanisme untuk mengakses objek yang disimpan di Amazon S3 melalui titik akhir gateway virtual private cloud (VPC).
4. File Java Archive (JAR) untuk aplikasi Blu Age mungkin perlu berkomunikasi dengan sumber data lain, seperti Amazon Aurora.
5. Setelah selesai, skrip pembungkus mengirimkan file keluaran yang dihasilkan ke dalam bucket S3 untuk diproses lebih lanjut (misalnya, oleh layanan CloudWatch logging Amazon). Pola ini

juga mendukung pengiriman file log zip ke Amazon S3, jika Anda menggunakan alternatif untuk CloudWatch logging standar.

Alat

Layanan AWS

- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.

Alat

- Docker adalah platform perangkat lunak untuk membangun, menguji, dan menyebarkan aplikasi. Docker mengemas perangkat lunak ke dalam unit standar yang disebut kontainer, yang memiliki semua yang dibutuhkan perangkat lunak untuk dijalankan, termasuk perpustakaan, alat sistem, kode, dan runtime. Anda dapat menggunakan Docker untuk menyebarkan dan menskalakan aplikasi ke lingkungan apa pun.
- Bash adalah antarmuka bahasa perintah (shell) untuk sistem operasi GNU.
- Java adalah bahasa pemrograman dan lingkungan pengembangan yang digunakan dalam pola ini.
- Blu Age adalah alat modernisasi mainframe AWS yang mengubah beban kerja mainframe lama, termasuk kode aplikasi, dependensi, dan infrastruktur, menjadi beban kerja modern untuk cloud.

Repositori kode

Kode untuk pola ini tersedia di repositori wadah sampel GitHub Blu Age.

Praktik terbaik

- Eksternalisasi variabel untuk mengubah perilaku aplikasi Anda dengan menggunakan variabel lingkungan. Variabel-variabel ini memungkinkan solusi orkestrasi kontainer untuk mengubah lingkungan runtime tanpa membangun kembali wadah. Pola ini mencakup contoh variabel lingkungan yang dapat berguna untuk aplikasi Blu Age.

- Validasi dependensi aplikasi apa pun sebelum Anda menjalankan aplikasi Blu Age Anda. Misalnya, verifikasi bahwa database tersedia dan kredensialnya valid. Tulis tes dalam skrip pembungkus untuk memverifikasi dependensi, dan gagal lebih awal jika tidak terpenuhi.
- Gunakan logging verbose dalam skrip pembungkus. Berinteraksi langsung dengan wadah yang sedang berjalan dapat menjadi tantangan, tergantung pada platform orkestrasi dan berapa lama pekerjaan itu berlangsung. Pastikan bahwa output yang berguna ditulis `STDOUT` untuk membantu mendiagnosis masalah apa pun. Misalnya, output mungkin termasuk isi direktori kerja aplikasi baik sebelum dan sesudah Anda menjalankan aplikasi.

Epik

Dapatkan file JAR aplikasi Blu Age

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 1 - Bekerja dengan Blu Age untuk mendapatkan file JAR aplikasi Anda.	Wadah dalam pola ini membutuhkan aplikasi Blu Age. Atau, Anda dapat menggunakan contoh aplikasi Java yang disediakan dengan pola ini untuk prototipe. Bekerja dengan tim Blu Age untuk mendapatkan file JAR untuk aplikasi Anda yang dapat dipanggang ke dalam wadah. Jika file JAR tidak tersedia, lihat tugas berikutny a untuk menggunakan aplikasi sampel sebagai gantinya.	Arsitek awan
Opsi 2 - Bangun atau gunakan file JAR aplikasi sampel yang disediakan.	Pola ini menyediakan file JAR sampel prebuilt. File ini mengeluarkan variabel lingkungan aplikasi `STDOUT` sebelum tidur selama 30 detik dan keluar.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	File ini diberi nama `bluAgeSample.jar` dan terletak di folder docker repositori. GitHub Jika Anda ingin mengubah kode dan membangun versi file JAR Anda sendiri, gunakan kode sumber yang terletak di. /java_ sample/src/sample _java_app.java di repositori. GitHub Anda dapat menggunakan skrip build di. /java_sample/build.sh untuk mengkompilasi sumber Java dan membangun JAR fie baru.	

Bangun wadah Blu Age

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori.	Kloning repositori kode sampel dengan menggunakan perintah: `git clone https://github.com/aws-samples/aws-blu-age-sample-container`	AWS DevOps
Gunakan Docker untuk membangun wadah.	Gunakan Docker untuk membangun wadah sebelum Anda mendorongnya ke registri Docker seperti Amazon ECR:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Dari terminal yang Anda pilih, arahkan ke `docker` folder di dalam GitHub repositori lokal Anda. 2. Gunakan perintah ini untuk membangun wadah: `docker build -t <tag> .` di `<tag>` mana nama wadah yang ingin Anda gunakan.	
Uji wadah Blu Age.	(Opsional) Jika perlu, uji wadah secara lokal dengan menggunakan perintah: `docker run -it <tag> / bin/bash`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Otentikasi ke repositori Docker Anda.	Jika Anda berencana untuk menggunakan Amazon ECR, ikuti petunjuk dalam dokumentasi Amazon ECR untuk menginstal dan mengonfigurasi AWS CLI dan mengautentikasi CLI Docker ke registri default Anda. Kami menyarankan Anda menggunakan get-login-password perintah untuk otentikasi. **Note** Konsol Amazon ECR menyediakan versi pra-isi dari perintah ini jika Anda menggunakan tombol perintah tekan Lihat. Untuk informasi selengkapnya, lihat dokumentasi Amazon ECR. `aws ecr get-login -password --region <region> | docker login --username AWS --password-stdin <account>.dkr.ecr.<region>.amazonaws.com`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jika Anda tidak berencana untuk menggunakan Amazon ECR, ikuti petunjuk yang diberikan untuk sistem registri kontainer Anda.	
Buat repositori kontainer.	Buat repositori di Amazon ECR. Untuk petunjuknya, lihat pola Menerapkan lingkungan untuk aplikasi Blu Age dalam kontainer menggunakan Terraform. Jika Anda menggunakan sistem registri kontainer lain, ikuti instruksi yang diberikan untuk sistem itu.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tandai dan dorong wadah Anda ke repositori target.	Jika Anda menggunakan Amazon ECR:
	1. Tandai gambar Docker lokal dengan registri dan repositori Amazon ECR, sehingga Anda dapat mendorongnya ke repositori jarak jauh Anda:
	`docker tag <tag>:latest <account>.dkr.ecr.<region>.amazonaws.com/<repository>:<versionNumber>`
	2. Dorong gambar ke repositori jarak jauh:
	`docker push <account>.dkr.ecr.<region>.amazonaws.com/<repository>:<versionNumber>`
	Untuk informasi selengkapnya, lihat Mendorong gambar Docker di Panduan Pengguna Amazon ECR.
		AWS DevOps

Sumber daya terkait

Sumber daya AWS

- Repositori wadah sampel AWS Blu Age

- Menjalankan beban kerja mainframe Blu Age yang dimodernisasi pada infrastruktur AWS tanpa server
- Menerapkan lingkungan untuk aplikasi Blu Age dalam kontainer dengan menggunakan Terraform
- Menggunakan Amazon ECR dengan AWS CLI (Panduan Pengguna Amazon ECR)
- Otentikasi registri pribadi (Panduan Pengguna Amazon ECR)
- Dokumentasi Amazon ECS
- Dokumentasi Amazon EKS

Sumber daya tambahan

- Situs Blu Age
- Situs Docker

Mengonversi dan membongkar data EBCDIC ke ASCII di AWS dengan menggunakan Python

Luis Gustavo Dantas, Amazon Web Services

Ringkasan

Karena mainframe biasanya meng-host data bisnis penting, memodernisasi data adalah salah satu tugas terpenting saat memigrasikan data ke Amazon Web Services (AWS) Cloud atau lingkungan American Standard Code for Information Interchange (ASCII) lainnya. Pada mainframe, data biasanya dikodekan dalam format Extended Binary-Coded Decimal Interchange Code (EBCDIC). Mengekspor database, Metode Akses Penyimpanan Virtual (VSAM), atau file datar umumnya menghasilkan file EBCDIC biner yang dikemas, yang lebih kompleks untuk dimigrasi. Solusi migrasi database yang paling umum digunakan adalah change data capture (CDC), yang, dalam banyak kasus, secara otomatis mengubah pengkodean data. Namun, mekanisme CDC mungkin tidak tersedia untuk database, VSAM, atau file datar ini. Untuk file-file ini, pendekatan alternatif diperlukan untuk memodernisasi data.

Pola ini menjelaskan cara memodernisasi data EBCDIC dengan mengubahnya menjadi format ASCII. Setelah konversi, Anda dapat memuat data ke database terdistribusi atau memiliki aplikasi di cloud memproses data secara langsung. Pola menggunakan skrip konversi dan file sampel dalam mainframe-data-utilities GitHub repositori.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- File input EBCDIC dan copybook bahasa berorientasi bisnis umum (COBOL) yang sesuai. Contoh file EBCDIC dan copybook COBOL disertakan dalam repositori. mainframe-data-utilities GitHub Untuk informasi lebih lanjut tentang copybook COBOL, lihat Enterprise COBOL untuk z/OS 6.4 Panduan Pemrograman di situs web IBM.

Batasan

- Layout file yang didefinisikan di dalam program COBOL tidak didukung. Mereka harus tersedia secara terpisah.

Versi produk

- Python versi 3.8 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Data EBCDIC pada mainframe
- Copybook COBOL

Tumpukan teknologi target

- Instans Amazon Elastic Compute Cloud (Amazon EC2) di cloud pribadi virtual (VPC)
- Amazon Elastic Block Store (Amazon EBS)
- Python dan paket yang diperlukan, JavaScript Object Notation (JSON), sys, dan datetime
- File datar ASCII siap dibaca oleh aplikasi modern atau dimuat dalam tabel database relasional

Arsitektur target

Diagram arsitektur menunjukkan proses konversi file EBCDIC ke file ASCII pada sebuah instance: EC2

1. Menggunakan skrip parse_copybook_to_json.py, Anda mengonversi copybook COBOL ke file JSON.
2. Menggunakan file JSON dan skrip extract_ebcdic_to_ascii.py, Anda mengonversi data EBCDIC ke file ASCII.

Otomatisasi dan skala

Setelah sumber daya yang diperlukan untuk konversi file manual pertama tersedia, Anda dapat mengotomatiskan konversi file. Pola ini tidak termasuk instruksi untuk otomatisasi. Ada beberapa cara untuk mengotomatiskan konversi. Berikut ini adalah ikhtisar dari satu pendekatan yang mungkin:

1. Merangkum perintah AWS Command Line Interface (AWS CLI) dan perintah skrip Python ke dalam skrip shell.

2. Buat fungsi AWS Lambda yang secara asinkron mengirimkan tugas skrip shell ke dalam sebuah instance. EC2 Untuk informasi selengkapnya, lihat Menjadwalkan lowongan SSH menggunakan AWS Lambda.
3. Buat pemicu Amazon Simple Storage Service (Amazon S3) yang memanggil fungsi Lambda setiap kali file lama diunggah. Untuk informasi selengkapnya, lihat Menggunakan pemicu Amazon S3 untuk menjalankan fungsi Lambda.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan, dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon Elastic Compute Cloud (Amazon). EC2
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

Alat lainnya

- GitHubadalah layanan hosting kode yang menyediakan alat kolaborasi dan kontrol versi.
- Python adalah bahasa pemrograman tingkat tinggi.

Repositori kode

Kode untuk pola ini tersedia di mainframe-data-utilities GitHub repositori.

Epik

Siapkan EC2 instance

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan sebuah EC2 instance.	EC2 Instans harus memiliki akses internet keluar. Hal ini memungkinkan instance untuk mengakses kode sumber Python yang tersedia di. GitHub Untuk membuat instance: 1. Buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2. 2. Luncurkan instance EC2 Linux. Gunakan alamat IP publik dan izinkan akses masuk melalui port 22. Pastikan bahwa ukuran penyimpanan instance setidaknya dua kali ukuran file data EBCDIC. Untuk petunjuk, lihat EC2 dokumentasi Amazon.	AWS Umum
Instal Git.	1. Menggunakan klien shell aman (SSH), sambungkan ke EC2 instance yang baru saja Anda luncurkan. Untuk informasi selengkapnya, lihat Connect ke instans Linux Anda. 2. Di EC2 konsol Amazon, jalankan perintah berikut.	AWS Umum, Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Ini menginstal Git pada EC2 instance.
	`sudo yum install git`
	3. Jalankan perintah berikut dan konfirmasikan bahwa Git telah berhasil diinstal.
	`git --version`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal Python.	1. Di EC2 konsol Amazon, jalankan perintah berikut. Ini menginstal Python pada EC2 instance.
	`sudo yum install python3`
	2. Di EC2 konsol Amazon, jalankan perintah berikut. Ini menginstal Pip3 pada instance. EC2
	`sudo yum install python3-pip`
	3. Di EC2 konsol Amazon, jalankan perintah berikut. Ini menginstal AWS SDK for Python (Boto3) pada instance. EC2
	`sudo pip3 install boto3`
	4. Di EC2 konsol Amazon, jalankan perintah berikut, di `<us-east-1>` mana kode untuk Wilayah AWS Anda. Untuk daftar lengkap kode Wilayah, lihat Wilayah yang Tersedia di EC2 dokumentasi Amazon.
		AWS Umum, Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`export AWS_DEFAU LT_REGION=<us-east -1>`	
Kloning GitHub repositori.	1. Di EC2 konsol Amazon, jalankan perintah berikut. Ini mengkloning mainframe -data-utilitiesrepositori dari GitHub dan membuka lokasi salinan default, folder. home `git clone https://g ithub.com/aws-samp les/mainframe-data- utilities.git` 2. Di home folder, konfirmas ikan bahwa `mainframe- data-utilities` folder tersebut ada.	AWS Umum, GitHub

Buat file ASCII dari data EBCDIC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Parse copybook COBOL ke dalam file layout JSON.	Di dalam `mainframe- data-utilities` folder, jalankan skrip parse_cop ybook_to_json.py. Modul otomatisasi ini membaca tata letak file dari copybook COBOL dan membuat file JSON. File JSON berisi informasi yang diperluka	AWS Umum, Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	n untuk menafsirkan dan mengekstrak data dari file sumber. Ini menciptakan metadata JSON dari copybook COBOL.
	
	Perintah berikut mengkonve rsi copybook COBOL ke file JSON.
	
	`python3 parse_cop ybook_to_json.py \ -copybook LegacyRef erence/COBPACK2.cpy \ -output sample-data/ cobpack2-list.json \ -dict sample-data/ cobpack2-dict.json \ -ebcdic sample-data/ COBPACK.OUTFILE.txt \ -ascii sample-data/ COBPACK.ASCII.txt \ -print 10000`
	
	Script mencetak argumen yang diterima.
	
	`---------- ---------- ---------- ---------- Copybook file..... .....| LegacyRef erence/COBPACK2.cpy Parsed copybook (JSON List).| sample-data/ cobpack2-list.json`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>JSON Dict (document
	ation)...| sample-da
	ta/cobpack2-dict.json
	ASCII file.....
	.....| sample-
	data/COBPACK.ASCII.t
	xt
	EBCDIC file.....
	.....| sample-
	data/COBPACK.OUTFILE
	.txt
	Print each.....
	.....| 10000
	---------- 	
	---------- 	
	---------- 	
	----------</pre> Untuk informasi selengkap nya tentang argumen, lihat file README di GitHub repositori.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa file tata letak JSON.	1. Arahkan ke jalur output yang ditentukan dalam skrip parse_copybook_to_json.py. 2. Periksa waktu pembuatan file sample-data/cobpack2-list.json untuk mengonfirmasi bahwa Anda telah memilih file tata letak JSON yang sesuai. 3. Periksa file JSON dan konfirmasikan bahwa isinya mirip dengan yang berikut ini. `"input": "extract-ebcdic-to-ascii/COBPACK.OUTFILE.txt", "output": "extract-ebcdic-to-ascii/COBPACK.ASCII.txt", "max": 0, "skip": 0, "print": 10000, "lrecl": 150, "rem-low-values": true, "separator": "|", "transf": [ { "type": "ch", "bytes": 19, "name": "OUTFILE-TEXT" }`	AWS Umum, JSON

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Atribut yang paling penting dari file layout JSON adalah: • `input`— Berisi jalur file EBCDIC yang akan dikonversi • `output`— Mendefinisikan jalur di mana file ASCII akan dihasilkan • `lrecl`- Menentukan ukuran dalam byte panjang catatan logis • `transf`— Daftar semua bidang dan ukurannya dalam byte Untuk informasi selengkapnya tentang file tata letak JSON, lihat file README di repositori. GitHub	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file ASCII.	Jalankan skrip extract_e bcdic_to_ascii.py, yang termasuk dalam GitHub repositori kloning. Skrip ini membaca file EBCDIC dan menulis file ASCII yang dikonversi dan dapat dibaca.
	`python3 extract_e bcdic_to_ascii.py - local-json sample-da ta/cobpack2-list.json`
	Saat skrip memproses data EBCDIC, ia mencetak pesan untuk setiap batch 10.000 catatan. Lihat contoh berikut ini.
	`---------- ---------- ---------- ---------- 2023-05-15 21:21:46. 322253 | Local Json file | -local-json | sample-data/cobpack2- list.json 2023-05-15 21:21:47. 034556 | Records processed | 10000 2023-05-15 21:21:47. 736434 | Records processed | 20000 2023-05-15 21:21:48. 441696 | Records processed | 30000`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>2023-05-15 21:21:49.
	173781 | Records
	 processed | 40000
	2023-05-15 21:21:49.
	874779 | Records
	 processed | 50000
	2023-05-15 21:21:50.
	705873 | Records
	 processed | 60000
	2023-05-15 21:21:51.
	609335 | Records
	 processed | 70000
	2023-05-15 21:21:52.
	292989 | Records
	 processed | 80000
	2023-05-15 21:21:52.
	938366 | Records
	 processed | 89280
	2023-05-15 21:21:52.
	938448 Seconds 6.616232</pre>
	Untuk informasi tentang cara mengubah frekuensi cetak, lihat file README di GitHub repositori.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa file ASCII.	1. Periksa waktu pembuatan file extract-ebcdic-to-ascii/cobpack.ascii.txt untuk memverifikasi bahwa itu baru saja dibuat.
	2. Di EC2 konsol Amazon, masukkan perintah berikut. Ini membuka catatan pertama dari file ASCII.
	`head sample-data/COBPACK.ASCII.txt -n 1| xxd`
	3. Periksa isi catatan pertama. Karena file EBCDIC biasanya biner, mereka tidak memiliki karakter khusus carriage return dan line feed (CRLF). Skrip extract_ebcdic_to_ascii.py menambahkan karakter pipa sebagai pemisah kolom, yang didefinisikan dalam parameter skrip.
	
	Jika Anda menggunakan contoh file EBCDIC yang disediakan, berikut ini adalah catatan pertama dalam file ASCII.
	`00000000: 2d30 3030 3030 3030 3030 3130 3030 3030  -000000000100000`
		AWS Umum, Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	00000010: 3030 307c 3030 3030 3030 3030 3031 3030 000|00000 0000100 00000020: 3030 3030 3030 7c2d 3030 3030 3030 3030 000000|-0 0000000 00000030: 3031 3030 3030 3030 3030 7c30 7c30 7c31 0100000000| 0|0|1 00000040: 3030 3030 3030 3030 7c2d 3130 3030 3030 00000000| -100000 00000050: 3030 307c 3130 3030 3030 3030 307c 2d31 000|10000 0000|-1 00000060: 3030 3030 3030 3030 7c30 3030 3030 7c30 00000000| 00000|0 00000070: 3030 3030 7c31 3030 3030 3030 3030 7c2d 0000|1000 00000|- 00000080: 3130 3030 3030 3030 307c 3030 3030 3030 100000000| 000000 00000090: 3030 3030 3130 3030 3030 3030 307c 2d30 000010000 0000|-0 000000a0: 3030 3030 3030 3030 3031 3030 3030 3030 000000000 1000000	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`000000b0: 3030 7c41 7c41 7c0a 00|A|A|.`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Evaluasi file EBCDIC.	Di EC2 konsol Amazon, masukkan perintah berikut. Ini membuka catatan pertama dari file EBCDIC.
	`head sample-data/COBPACK.OUTFILE.txt -c 150 | xxd`
	Jika Anda menggunakan contoh file EBCDIC, berikut ini adalah hasilnya.
	`00000000: 60f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 f0f0 f0f0 `........ .... 00000010: f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 f0f0 .... 00000020: f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 .... 00000030: f0f0 f0f0 f0f0 d000 0000 0005 f5e1 00fa .... 00000040: 0a1f 0000 0000 0005 f5e1 00ff ffff fffa .... 00000050: 0a1f 0000 000f 0000 0c10 0000 000f 1000 .... 00000060: 0000 0d00 0000 0000 1000 0000`
		AWS Umum, Linux, EBCDIC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	0f00 0000 .... 00000070: 0000 1000 0000 0dc1 c100 0000 0000 0000 .... 00000080: 0000 0000 0000 0000 0000 0000 0000 0000 .... 00000090: 0000 0000 0000	
	Untuk mengevaluasi kesetaraan antara sumber dan file target, diperlukan pengetahuan komprehensif tentang EBCDIC. Misalnya, karakter pertama dari contoh file EBCDIC adalah tanda hubung (). - Dalam notasi heksadesimal dari file EBCDIC, karakter ini diwakili oleh, dan dalam notasi heksadesimal dari file ASCII60, karakter ini diwakili oleh. 2D Untuk tabel EBCDIC-to-ASCII konversi, lihat EBCDIC ke ASCII di situs web IBM.	

Sumber daya terkait

Referensi

- Set karakter EBCDIC (dokumentasi IBM)

- EBCDIC ke ASCII (dokumentasi IBM)
- COBOL (dokumentasi IBM)
- Konsep dasar JCL (dokumentasi IBM)
- Connect ke instans Linux Anda (EC2 dokumentasi Amazon)

Tutorial

- Menjadwalkan pekerjaan SSH menggunakan AWS Lambda (postingan blog AWS)
- Menggunakan pemicu Amazon S3 untuk menjalankan fungsi Lambda (dokumentasi AWS Lambda)

Konversi file mainframe dari format EBCDIC ke format ASCII yang dibatasi karakter di Amazon S3 menggunakan AWS Lambda

Luis Gustavo Dantas, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara meluncurkan AWS Lambda fungsi yang secara otomatis mengonversi file mainframe Extended Binary Coded Decimal Interchange Code (EBCDIC) ke file American Standard Code for Information Interchange (ASCII) yang dibatasi karakter. Fungsi Lambda berjalan setelah file ASCII diunggah ke bucket Amazon Simple Storage Service (Amazon S3). Setelah konversi file, Anda dapat membaca file ASCII pada beban kerja berbasis x86 atau memuat file ke dalam database modern.

Pendekatan konversi file yang ditunjukkan dalam pola ini dapat membantu Anda mengatasi tantangan bekerja dengan file EBCDIC di lingkungan modern. File yang dikodekan dalam EBCDIC sering berisi data yang direpresentasikan dalam format desimal biner atau dikemas, dan bidang memiliki panjang tetap. Karakteristik ini menciptakan hambatan karena beban kerja berbasis x86 modern atau lingkungan terdistribusi umumnya bekerja dengan data yang dikodekan ASCII dan tidak dapat memproses file EBCDIC.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Bucket Amazon S3
- Pengguna AWS Identity and Access Management (IAM) dengan izin administratif
- AWS CloudShell
- Python 3.8.0 atau yang lebih baru
- File datar yang dikodekan dalam EBCDIC dan struktur data yang sesuai dalam copybook bahasa berorientasi bisnis umum (COBOL)

> **Note**
>
> Pola ini menggunakan contoh file EBCDIC (Client.ebcdic.txt) dan copybook COBOL yang sesuai (COBKS05.cpy). Kedua file tersedia di GitHub mainframe-data-utilitiesrepositori.

Batasan

- Copybook COBOL biasanya memiliki beberapa definisi tata letak. mainframe-data-utilitiesProyek ini dapat mengurai jenis copybook ini tetapi tidak dapat menyimpulkan tata letak mana yang harus dipertimbangkan pada konversi data. Ini karena copybook tidak memegang logika ini (yang tetap pada program COBOL sebagai gantinya). Akibatnya, Anda harus secara manual mengkonfigurasi aturan untuk memilih tata letak setelah Anda mengurai copybook.
- Pola ini tunduk pada kuota Lambda.

Arsitektur

Tumpukan teknologi sumber

- IBM z/OS, IBM i, dan sistem EBCDIC lainnya
- File berurutan dengan data yang dikodekan dalam EBCDIC (seperti IBM Db2 unload)
- Copybook COBOL

Tumpukan teknologi target

- Amazon S3
- Pemberitahuan acara Amazon S3
- IAM
- Fungsi Lambda
- Python 3.8 atau yang lebih baru
- Utilitas Data Mainframe
- Metadata JSON
- File ASCII yang dibatasi karakter

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk mengkonversi file mainframe EBCDIC ke file ASCII.

Diagram menunjukkan alur kerja berikut:

1. Pengguna menjalankan skrip parser copybook, yang mengubah copybook COBOL menjadi file JSON.
2. Pengguna mengunggah metadata JSON ke bucket Amazon S3. Ini membuat metadata dapat dibaca oleh fungsi Lambda konversi data.
3. Pengguna atau proses otomatis mengunggah file EBCDIC ke bucket Amazon S3.
4. Peristiwa notifikasi Amazon S3 memicu fungsi Lambda konversi data.
5. AWS memverifikasi izin baca-tulis bucket Amazon S3 untuk fungsi Lambda.
6. Lambda membaca file dari bucket Amazon S3 dan mengonversi file secara lokal dari EBCDIC ke ASCII.
7. Lambda mencatat status proses di Amazon. CloudWatch
8. Lambda menulis file ASCII kembali ke Amazon S3.

> **Note**
>
> Skrip parser copybook berjalan satu kali untuk melakukan konversi metadata ke format JSON, yang kemudian disimpan dalam bucket Amazon S3. Setelah konversi awal, semua file EBCDIC berikutnya yang mereferensikan file JSON yang sama di bucket Amazon S3 akan menggunakan konfigurasi metadata yang ada.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS CloudShelladalah shell berbasis browser yang dapat Anda gunakan untuk mengelola Layanan AWS dengan menggunakan AWS Command Line Interface (AWS CLI) dan berbagai alat pengembangan yang sudah diinstal sebelumnya.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya bila diperlukan dan menskalakan secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Alat-alat lainnya

- GitHubadalah layanan hosting kode yang menyediakan alat kolaborasi dan kontrol versi.
- Python adalah bahasa pemrograman tingkat tinggi.

Kode

Kode untuk pola ini tersedia di GitHub mainframe-data-utilitiesrepositori.

Praktik terbaik

Pertimbangkan praktik terbaik berikut:

- Tetapkan izin yang diperlukan di tingkat Amazon Resource Name (ARN).
- Selalu berikan izin hak istimewa paling sedikit untuk kebijakan IAM. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan di IAM dalam dokumentasi IAM.

Epik

Buat variabel lingkungan dan folder kerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat variabel lingkungan.	Salin variabel lingkungan berikut ke editor teks, lalu ganti `<placeholder>` nilai dalam contoh berikut dengan nilai sumber daya Anda:
	`bucket=<your_bucket_name>`
	`account=<your_account_number>`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`region=<your_reg ion_code>` **Note** Anda akan membuat referensi ke bucket Amazon S3 Anda, Akun AWS, dan Wilayah AWS yang lebih baru. Untuk menentukan variabel lingkungan, buka CloudShell konsol, lalu salin dan tempel variabel lingkungan Anda yang diperbarui ke baris perintah. **Note** Anda harus mengulangi langkah ini setiap kali CloudShell sesi dimulai ulang.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat folder yang berfungsi.	Untuk menyederhanakan proses pembersihan sumber daya nanti, buat folder kerja CloudShell dengan menjalankan perintah berikut: `mkdir workdir; cd workdir` **Note** Anda harus mengubah direktori ke direktori kerja (`workdir`) setiap kali Anda kehilangan koneksi ke CloudShell sesi Anda.	AWS Umum

Mendefinisikan peran dan kebijakan IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan kepercayaan untuk fungsi Lambda.	Konverter EBCDIC berjalan dalam fungsi Lambda. Fungsi tersebut harus memiliki peran IAM. Sebelum Anda membuat peran IAM, Anda harus menentukan dokumen kebijakan kepercayaan yang memungkinkan sumber daya untuk mengambil kebijakan tersebut.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Dari folder CloudShell kerja, buat dokumen kebijakan dengan menjalankan perintah berikut:
	
	`E2ATrustPol=$(cat <<EOF`
	`{`
	`"Version": "2012-10-17",`
	`"Statement": [`
	`{`
	`"Effect": "Allow",`
	`"Principal": {`
	`"Service": "lambda.amazonaws.com"`
	`},`
	`"Action": "sts:AssumeRole"`
	`}`
	`]`
	`}`
	`EOF`
	`)`
	`printf "$E2ATrustPol" > E2ATrustPol.json`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM untuk konversi Lambda.	Untuk membuat peran IAM, jalankan AWS CLI perintah berikut dari folder CloudShell kerja: `aws iam create-role --role-name E2AConvLambdaRole --assume-role-policy-document file://E2ATrustPol.json`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat dokumen kebijakan IAM untuk fungsi Lambda.	Fungsi Lambda harus memiliki akses baca-tulis ke bucket Amazon S3 dan izin tulis untuk Amazon Log. CloudWatch
	
	Untuk membuat kebijakan IAM, jalankan perintah berikut dari folder CloudShell kerja:
	
	`E2APolicy=$(cat <<EOF`
	`{`
	`"Version": "2012-10-17",`
	`"Statement": [`
	`{`
	`"Sid": "Logs",`
	`"Effect": "Allow",`
	`"Action": [`
	`"logs:PutLogEvents",`
	`"logs:CreateLogStream",`
	`"logs:CreateLogGroup"`
	`],`
	`"Resource": [`
	`"arn:aws:logs:*:*:log-group:*",`
	`"arn:aws:logs:*:*:log-group:*:log-stream:*"`
	`]`
	`},`
	`{`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"Sid": "S3",
	"Effect":
	"Allow",
	"Action": [
	
	"s3:GetObject",
	
	"s3:PutObject",
	
	"s3:GetObjectVersion"
	],
	"Resource":
	[
	
	"arn:aws:s3:::%s/*",
	
	"arn:aws:s3:::%s"
	]
	}
	]
	}
	EOF
	)
	printf "$E2APolicy"
	"$bucket" "$bucket"
	> E2AConvLambdaPolic
	y.json</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan dokumen kebijakan IAM ke peran IAM.	Untuk melampirkan kebijakan IAM ke peran IAM, masukkan perintah berikut dari folder CloudShell kerja Anda: `aws iam put-role-policy --role-name E2AConvLambdaRole --policy-name E2AConvLambdaPolicy --policy-document file://E2AConvLambdaPolicy.json`	AWS Umum

Buat fungsi Lambda untuk konversi EBCDIC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kode sumber konversi EBCDIC.	Dari folder CloudShell kerja, jalankan perintah berikut untuk mengunduh kode mainframe -data-utilities sumber dari GitHub: `git clone https://github.com/aws-samples/mainframe-data-utilities.git mdu`	AWS Umum
Buat paket ZIP.	Dari folder CloudShell kerja, masukkan perintah berikut untuk membuat paket ZIP yang membuat fungsi Lambda untuk konversi EBCDIC:	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`cd mdu; zip ../mdu.zip *.py; cd ..`	
Buat fungsi Lambda.	Dari folder CloudShell kerja, masukkan perintah berikut untuk membuat fungsi Lambda untuk konversi EBCDIC: `aws lambda create-function \ --function-name E2A \ --runtime python3.9 \ --zip-file fileb://mdu.zip \ --handler extract_ebcdic_to_ascii.lambda_handler \ --role arn:aws:iam::$account:role/E2AConvLambdaRole \ --timeout 10 \ --environment "Variables={layout=$bucket/layout/}"` **Note** Tata letak variabel lingkungan memberi tahu fungsi Lambda tempat metadata JSON berada.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan berbasis sumber daya untuk fungsi Lambda.	Dari folder CloudShell kerja, masukkan perintah berikut untuk mengizinkan pemberitahuan peristiwa Amazon S3 Anda memicu fungsi Lambda untuk konversi EBCDIC: `aws lambda add-permission \ --function-name E2A \ --action lambda:InvokeFunction \ --principal s3.amazonaws.com \ --source-arn arn:aws:s3:::$bucket \ --source-account $account \ --statement-id 1`	AWS Umum

Buat notifikasi acara Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dokumen konfigurasi untuk notifikasi acara Amazon S3.	Pemberitahuan peristiwa Amazon S3 memulai fungsi Lambda konversi EBCDIC saat file ditempatkan di folder input. Dari folder CloudShell kerja, jalankan perintah berikut untuk membuat dokumen JSON untuk pemberitahuan acara Amazon S3:	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>S3E2AEvent=$(cat <<EOF
	{
	"LambdaFunctionConf
	igurations": [
	    {
	      "Id": "E2A",
	      "LambdaFu
	nctionArn": "arn:aws:
	lambda:%s:%s:funct
	ion:E2A",
	      "Events":
	 ["s3:ObjectCreated:
	Put"],
	      "Filter": {
	        "Key": {
	          "FilterRu
	les": [
	            {
	              "Name":
	 "prefix",
	              "Value":
	 "input/"
	            }
	          ]
	        }
	      }
	    }
	  ]
	}
	EOF
	)
	printf "$S3E2AEvent"
	 "$region" "$account" >
	 S3E2AEvent.json</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat notifikasi acara Amazon S3.	Dari folder CloudShell kerja, masukkan perintah berikut untuk membuat pemberita huan acara Amazon S3: `aws s3api put-bucket-notification-configuration --bucket $bucket --notification-configuration file://S3E2AEvent.json`	AWS Umum

Buat dan unggah metadata JSON

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mengurai copybook COBOL.	Dari folder CloudShell kerja, masukkan perintah berikut untuk mengurai contoh copybook COBOL ke dalam file JSON (yang mendefini sikan cara membaca dan mengiris file data dengan benar):
	`python3 mdu/parse_copybook_to_json.py \`
	`-copybook mdu/LegacyReference/COBKS05.cpy \`
	`-output CLIENT.json \`
	`-output-s3key CLIENT.ASCII.txt \`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`-output-s3bkt $bucket \`
	`-output-type  s3 \`
	`-print  25`	

Tambahkan aturan transform asi.	File data sampel dan copybook COBOL yang sesuai adalah file multi-tata letak. Ini berarti bahwa konversi harus mengiris data berdasarkan aturan tertentu. Dalam hal ini, byte pada posisi 3 dan 4 di setiap baris menentukan tata letak.
	
	Dari folder CloudShell kerja, edit `CLIENT.json` file dan ubah konten dari `"transf-rule": [],` yang berikut:
	
	`"transf-rule": [`
	`{`
	`"offset": 4,`
	`"size": 2,`
	`"hex": "0002",`
	`"transf": "transf1"`
	`},`
	`{`
	`"offset": 4,`
	`"size": 2,`
	`"hex": "0000",`
	`"transf": "transf2"`
	`}`
	`],`
		AWS Umum, IBM Mainframe, Cobol

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah metadata JSON ke bucket Amazon S3.	Dari folder CloudShell kerja, masukkan AWS CLI perintah berikut untuk mengunggah metadata JSON ke bucket Amazon S3 Anda: `aws s3 cp CLIENT.json s3://$bucket/layout/CLIENT.json`	AWS Umum

Konversikan file EBCDIC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim file EBCDIC ke bucket Amazon S3.	Dari folder CloudShell kerja, masukkan perintah berikut untuk mengirim file EBCDIC ke bucket Amazon S3: `aws s3 cp mdu/sample-data/CLIENT.EBCDIC.txt s3://$bucket/input/` **Note** Kami menyarankan Anda mengatur folder yang berbeda untuk file input (EBCDIC) dan output (ASCII) agar tidak memanggil fungsi konversi Lambda lagi saat file	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ASCII diunggah ke bucket Amazon S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa outputnya.	Dari folder CloudShell kerja, masukkan perintah berikut untuk memeriksa apakah file ASCII dihasilkan di bucket Amazon S3 Anda:
	`aws s3 ls s3://$bucket/`
	Note
	Konversi data dapat memakan waktu beberapa detik untuk terjadi. Kami menyarankan Anda memeriksa file ASCII beberapa kali.
	Setelah file ASCII tersedia, masukkan perintah berikut untuk melihat konten file yang dikonversi di bucket Amazon S3. Sesuai kebutuhan, Anda dapat mengunduhnya atau menggunakannya langsung dari bucket Amazon S3:
	`aws s3 cp s3://$bucket/ CLIENT.ASCII.txt - | head`
	Periksa konten file ASCII:
	`0|0|220|`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1|1|HERBERT MOHAMED|1 958-08-31|BACHELOR| 0010000.00| 1|2|36|THE ROE AVENUE| 2|1|JAYLEN GEORGE|19 69-05-29|ELEMENTARY| 0020000.00| 2|2|365|HEATHFIELD ESPLANADE| 3|1|MIKAEEL WEBER|198 2-02-17|MASTER|003 0000.00| 3|2|4555|MORRISON STRAND| 4|1|APRIL BARRERA|1 967-01-12|DOCTOR|0 030000.00| 4|2|1311|MARMION PARK| 5|1|ALEEZA PLANT|198 5-03-01|BACHELOR|0 008000.00|	

Bersihkan lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Siapkan variabel dan folder.	Jika Anda kehilangan koneksi dengan CloudShel l, sambungkan kembali dan kemudian masukkan perintah berikut untuk mengubah direktori ke folder kerja: `cd workdir` Pastikan bahwa variabel lingkungan didefinisikan:	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`bucket=<your_bucket_name>`
	`account=<your_account_number>`
	`region=<your_region_code>`	

Hapus konfigurasi notifikasi untuk bucket.	Dari folder CloudShell kerja, jalankan perintah berikut untuk menghapus konfigurasi pemberitahuan acara Amazon S3:
	`aws s3api put-bucket-notification-configuration \`
	`--bucket=$bucket \`
	`--notification-configuration="{}"`
		AWS Umum

Hapus fungsi Lambda.	Dari folder CloudShell kerja, masukkan perintah berikut untuk menghapus fungsi Lambda untuk konverter EBCDIC:
	`aws lambda delete-function \`
	`--function-name E2A`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hapus peran dan kebijakan IAM.	Dari folder CloudShell kerja, masukkan perintah berikut untuk menghapus peran dan kebijakan konverter EBCDIC:
	`aws iam delete-role-policy \`
	`--role-name E2AConvLambdaRole \`
	`--policy-name E2AConvLambdaPolicy`
	
	`aws iam delete-role \`
	`--role-name E2AConvLambdaRole`
		AWS Umum

Hapus file yang dihasilkan di bucket Amazon S3.	Dari folder CloudShell kerja, masukkan perintah berikut untuk menghapus file yang dihasilkan di bucket Amazon S3:
	`aws s3 rm s3://$bucket/layout --recursive`
	`aws s3 rm s3://$bucket/input --recursive`
	`aws s3 rm s3://$bucket/CLIENT.ASCII.txt`
		AWS Umum

Hapus folder kerja.	Dari folder CloudShell kerja, masukkan perintah berikut untuk menghapus `workdir` dan isinya: `cd ..; rm -Rf workdir`	AWS Umum

Sumber daya terkait

- Utilitas Data Mainframe README () GitHub
- Set karakter EBCDIC (dokumentasi IBM)
- EBCDIC ke ASCII (dokumentasi IBM)
- COBOL (dokumentasi IBM)
- Menggunakan pemicu Amazon S3 untuk menjalankan fungsi Lambda (dokumentasi)AWS Lambda

Mengkonversi file data mainframe dengan layout rekaman yang kompleks menggunakan Micro Focus

Peter West, Amazon Web Services

Ringkasan

Catatan: AWS Mainframe Modernization Layanan (pengalaman Lingkungan Runtime Terkelola) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Pola ini menunjukkan kepada Anda cara mengonversi file data mainframe dengan data non-teks dan tata letak rekaman kompleks dari EBCDIC (Extended Binary Coded Decimal Interchange Code) pengkodean karakter ke pengkodean karakter ASCII (American Standard Code for Information Interchange) dengan menggunakan file struktur Fokus Mikro. Untuk menyelesaikan konversi file, Anda harus melakukan hal berikut:

1. Siapkan file sumber tunggal yang menjelaskan semua item data dan rekam tata letak di lingkungan mainframe Anda.
2. Buat file struktur yang berisi tata letak catatan data dengan menggunakan Editor File Data Fokus Mikro sebagai bagian dari Alat File Data Klasik Fokus Mikro atau Alat File Data. File struktur mengidentifikasi data non-teks sehingga Anda dapat mengonversi file mainframe dengan benar dari EBCDIC ke ASCII.
3. Uji file struktur dengan menggunakan Classic Data File Tools atau Data File Tools.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Micro Focus Enterprise Developer untuk Windows, tersedia melalui AWS Mainframe Modernisasi

Versi produk

- Micro Focus Enterprise Server 7.0 dan yang lebih baru

Alat

- Micro Focus Enterprise Developer menyediakan lingkungan berjalan untuk aplikasi yang dibuat dengan varian lingkungan pengembangan terintegrasi (IDE) dari Enterprise Developer.
- Alat File Data Micro Focus Classic membantu Anda mengonversi, menavigasi, mengedit, dan membuat file data. Alat File Data Klasik termasuk Data File Converter, Record Layout Editor, dan Data File Editor.
- Alat File Data Fokus Mikro membantu Anda membuat, mengedit, dan memindahkan file data. Alat File Data termasuk Editor File Data, Utilitas Konversi File, dan Utilitas Baris Perintah Struktur File Data.

Epik

Siapkan file sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi komponen sumber.	Identifikasi semua tata letak rekaman yang mungkin untuk file, termasuk definisi ulang yang berisi data non-teks. Jika Anda memiliki tata letak yang berisi definisi ulang, Anda harus memfaktorkan tata letak ini ke tata letak unik yang menggambarkan setiap kemungkinan permutasi struktur data. Biasanya, tata letak catatan file data dapat dijelaskan oleh arketipe berikut: • Rekam tata letak hanya dengan data teks • Rekam tata letak dengan data non-teks	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Rekam tata letak dengan data non-teks yang berada di bawah klausa REDEFINES Untuk informasi selengkap nya tentang membuat tata letak rekaman yang diratakan untuk file yang berisi tata letak rekaman kompleks, lihat Menghosting ulang aplikasi EBCDIC di lingkungan ASCII untuk migrasi mainframe.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Identifikasi kondisi tata letak rekaman.	Untuk file dengan beberapa tata letak rekaman atau file yang berisi tata letak kompleks dengan klausa REDEFINES, identifikasi data dan kondisi dalam catatan yang dapat Anda gunakan untuk menentuka n tata letak mana yang akan digunakan selama konversi. Kami menyarankan Anda mendiskusikan tugas ini dengan pakar materi pelajaran (UKM) yang memahami program yang memproses file-file ini.
	
	Misalnya, file mungkin berisi dua jenis rekaman yang berisi data non-teks. Anda dapat memeriksa sumbernya dan mungkin menemukan kode yang mirip dengan yang berikut ini:
	
	`MOVE "M" TO PART-TYPE`
	`MOVE "MAIN ASSEMBLY" TO`
	`PART-NAME`
	`MOVE "S" TO PART-TYPE`
	`MOVE "SUB ASSEMBLY 1"`
	`TO PART-NAME`
	
	Kode ini membantu Anda mengidentifikasi hal-hal berikut:
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Bidang “PART-TYPE” digunakan untuk menentukan jenis rekaman • Nilai “M” digunakan untuk “M-PART-RECORD” • Nilai “S” digunakan untuk “S-PART-RECORD” Anda dapat mendokumentasikan nilai yang digunakan oleh bidang ini untuk mengaitkan tata letak rekaman dengan catatan data yang benar dalam file.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membangun file sumber.	Jika file dijelaskan melalui beberapa file sumber atau jika tata letak rekaman berisi data non-teks yang berada di bawah klausa REDEFINES , maka buat file sumber baru yang berisi tata letak rekaman. Program baru tidak perlu mendeskripsikan file menggunakan pernyataan SELECT dan FD. Program ini hanya dapat berisi deskripsi catatan sebagai 01 tingkat dalam Working-Storage. **Note** Anda dapat membuat file sumber untuk setiap file data atau membuat file sumber master yang menjelaskan semua file data.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kompilasi file sumber.	Kompilasi file sumber untuk membangun kamus data. Kami menyarankan Anda mengkompilasi file sumber dengan menggunakan set karakter EBCDIC. Jika direktif IBMCOMP atau arahan ODOSLIDE sedang digunakan, maka Anda harus menggunakan arahan ini dalam file sumber juga. **Note** IBMCOMP mempengaruhi penyimpanan byte bidang COMP dan ODOSLIDE mempengaruhi padding pada struktur YANG BERBEDA TERJADI. Jika arahan ini tidak disetel dengan benar, maka alat konversi tidak akan membaca catatan data dengan benar. Ini menghasilkan data yang buruk dalam file yang dikonversi.	Pengembang aplikasi

(Opsi A) Buat file struktur menggunakan Alat File Data Klasik

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai alat dan muat kamus.	1. Pilih ikon menu Start Windows, cari dan pilih Micro Focus Enterprise Developer, lalu pilih Classic Data File Tools. 2. Pilih File, lalu pilih Record Layout. 3. Dalam Pilih file untuk membangun tata letak dari kotak dialog, untuk Nama file, pilih file IDY (`.idy`) yang dibuat saat Anda mengkompilasi file sumber sebelumnya. Kemudian, pilih Buka. 4. Untuk mengonfirmasi bahwa Classic Data File Tools menggunakan EBCDIC, di kotak dialog Data File Tools, pilih YES jika file IDY diatur ke EBCDIC dan Datatools diatur ke ANSI.	Pengembang aplikasi
Buat tata letak catatan default.	Gunakan tata letak rekaman default untuk semua catatan yang tidak cocok dengan tata letak bersyarat apa pun. 1. Di jendela Layout, perluas struktur data, dan kemudian temukan level 01 yang	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	digunakan untuk tata letak default. 2. Klik kanan item 01, lalu pilih New Layout. 3. Di kotak dialog New Record Layout Wizard, pilih Tata Letak Default, lalu pilih Berikutnya. 4. Pilih Selesai. Tata letak default muncul di panel Layouts dan dapat diidentifikasi oleh ikon folder merah.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tata letak catatan bersyarat.	Gunakan tata letak catatan bersyarat bila ada lebih dari satu tata letak rekaman dalam sebuah file. 1. Di panel Layouts, perluas struktur data, lalu temukan level 01 yang digunakan untuk tata letak bersyarat. 2. Klik kanan item 01, lalu pilih New Layout. 3. Di kotak dialog New Record Layout Wizard, pilih Tata Letak Bersyarat, lalu pilih Berikutnya. 4. Pilih Selesai. Tata letak bersyarat muncul di panel Layouts dan dapat diidentif ikasi dengan ikon folder kuning. 5. Perluas tata letak bersyarat , klik kanan bidang tempat Anda harus menempatkan kondisi, lalu pilih Properties. 6. Dalam kotak dialog Field Properties, masukkan kondisi. Konfirmasikan bahwa set karakter diatur ke EBCDIC, lalu pilih OK. Tanda centang muncul di sebelah bidang yang memiliki kondisi yang ditetapkan.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Ulangi langkah 5—6 untuk bidang lain yang memerlukan kondisi untuk tata letak ini. 8. Ulangi langkah 1—6 untuk tata letak bersyarat lainnya yang harus ditambahkan. 9. Pilih File, pilih Save As, dan kemudian simpan file struktur ke disk.	

(Opsi B) Buat file struktur menggunakan Alat File Data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai alat dan muat kamus.	1. Pilih ikon menu Start Windows, cari dan pilih Pengembang Perusahaan Fokus Mikro, lalu pilih Alat File Data. 2. Pilih File, Baru, File Struktur. 3. Di kotak dialog Buka, untuk nama File, pilih file IDY (`.idy`) yang dibuat saat Anda mengkompilasi file sumber sebelumnya. Kemudian, pilih Buka. 4. Untuk mengonfirmasi bahwa Alat File Data menggunakan EBCDIC, konfirmasikan bahwa menu	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tarik-turun di bagian File Debug diatur ke EBCDIC.	
Buat tata letak catatan default.	Gunakan tata letak catatan default untuk semua catatan yang tidak cocok dengan tata letak bersyarat apa pun. 1. Di bagian Tata Letak yang Tersedia di panel kiri, perluas struktur data, lalu temukan level 01 yang digunakan untuk tata letak default. 2. Klik kanan item 01, lalu pilih Buat Tata Letak Default. Tata letak default muncul di panel Layouts dan dapat diidentifikasi dengan ikon “D” biru.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tata letak catatan bersyarat.	Gunakan tata letak catatan bersyarat bila ada lebih dari satu tata letak rekaman dalam sebuah file. 1. Di bagian Tata Letak Terpilih di panel kanan, perluas struktur data, lalu temukan level 01 yang digunakan untuk tata letak bersyarat. 2. Klik kanan item 01, lalu pilih Buat Tata Letak Bersyarat. Tata letak bersyarat muncul di panel Layouts di sisi kanan dan dapat diidentif ikasi dengan ikon hijau “C”. 3. Perluas tata letak bersyarat , klik kanan bidang tempat Anda harus menempatkan kondisi, lalu pilih Properties. 4. Dalam kotak dialog Field Properties, masukkan kondisi. Konfirmasikan bahwa set karakter diatur ke EBCDIC, lalu pilih OK. Ikon “IF” merah muncul di sebelah bidang yang memiliki set kondisi. 5. Ulangi langkah 3—4 untuk bidang lain yang memerluka n kondisi untuk tata letak ini.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Ulangi langkah 1-4 untuk tata letak bersyarat lainnya yang harus ditambahkan. 7. Pilih File, pilih Save As, dan kemudian simpan file struktur ke disk.	

(Opsi A) Uji file struktur menggunakan Alat File Data Klasik

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji file data EBCDIC.	Konfirmasikan bahwa Anda dapat menggunakan file struktur Anda untuk melihat file data uji EBCDIC dengan benar. 1. Pilih ikon menu Start Windows, temukan dan pilih Micro Focus Enterprise Developer, lalu pilih Classic Data Tools. 2. Pilih File, lalu pilih Buka. 3. Di kotak dialog Buka, untuk nama file, pilih dataset EBCDIC, lalu pilih Buka. 4. Pilih File, Editor File Data, Muat Tata Letak Rekaman. 5. Di kotak dialog Buka, untuk Nama file, pilih file struktur, lalu pilih Buka. 6. Untuk mengonfirmasi bahwa mode set karakter	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	diatur ke EBCDIC, konfirmasikan bahwa menu tarik-turun diatur ke EBCDIC. Anda dapat melihat data rekaman mentah di panel kiri dan data yang diformat di panel kanan. 7. Pilih berbagai catatan untuk memastikan bahwa semua format dirender dengan tata letak yang benar.	

(Opsi B) Uji file struktur menggunakan Alat File Data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji file data EBCDIC.	Konfirmasikan bahwa Anda dapat menggunakan file struktur Anda untuk melihat file data uji EBCDIC dengan benar. 1. Pilih ikon menu Start Windows, cari dan pilih Pengembang Perusahaan Fokus Mikro, lalu pilih Alat File Data. 2. Pilih File, Buka, File Data. 3. Di kotak dialog Buka File Data, pada tab Lokal, untuk Nama File, pilih Jelajahi untuk menemukan lokasi file uji EBCDIC.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk File Struktur (opsional), pilih Browse untuk menemukan lokasi file struktur. 5. Di bagian Rincian File, masukkan detail file dan konfirmasikan bahwa Pengkodean diatur ke EBCDIC. 6. Pilih mode Buka Bersama atau Buka Eksklusif tergantung pada kebutuhan Anda. 7. Konfirmasikan bahwa menu drop-down di bagian Appearance pada toolbar diatur ke EBCDIC. Anda akan melihat data catatan mentah di panel kiri dan data yang diformat di panel kanan. 8. Pilih berbagai catatan untuk memastikan bahwa semua format dirender dengan tata letak yang benar.	

Uji konversi file data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji Konversi Berkas EBCDIC.	1. Pilih ikon menu Start Windows, cari dan pilih Pengembang Perusahaan	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Fokus Mikro, lalu pilih Alat Data Klasik. 2. Pilih Tools, lalu pilih Convert. 3. Dalam kotak dialog Konversi File Data, di bagian File input, untuk Nama file, pilih Jelajahi untuk menemukan dan memilih file input EBCDIC. Konfirmasikan bahwa set Karakter diatur ke EBCDIC. 4. Di Konversi Set Karakter bagian, pilih Konversi set karakter dan Catatan berisi item data non-teks kotak centang. Pilih Pilih tata letak untuk konversi, lalu pilih Browse untuk menemukan dan memilih file struktur. 5. Di bagian File baru, untuk Nama File, masukkan jalur dan nama file dari file keluaran ASCII yang ingin Anda buat. Secara default, alat konversi default ke format yang sama dengan file input. Untuk pengujian, biarkan opsi disetel ke nilai defaultnya. 6. Pilih Konversi. 7. Ikuti langkah-langkah di bagian (Opsi A) Uji file struktur menggunakan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Alat File Data Klasik atau (Opsi B) Uji file struktur menggunakan bagian Alat File Data, tetapi muat file keluaran ASCII alih-alih file EBCDIC. 8. Muat file EBCDIC dan ASCII ke dalam Editor File Data, lalu bandingkan file secara berdampingan untuk memeriksa keakuratan konversi.	

Sumber daya terkait

- Fokus Mikro (dokumentasi Fokus Mikro)
- Mainframe dan kode lama (postingan Blog AWS)
- AWS Prescriptive Guidance (dokumentasi AWS)
- Dokumentasi AWS (dokumentasi AWS)
- Referensi Umum AWS (dokumentasi AWS)
- Glosarium AWS (dokumentasi AWS)

Menerapkan lingkungan untuk aplikasi Blu Age kontainer dengan menggunakan Terraform

Richard Milner-Watts, Amazon Web Services

Ringkasan

Migrasi beban kerja mainframe lama ke arsitektur cloud modern dapat menghilangkan biaya pemeliharaan mainframe—biaya yang hanya meningkat seiring bertambahnya usia lingkungan. Namun, migrasi pekerjaan dari mainframe dapat menimbulkan tantangan unik. Sumber daya internal mungkin tidak akrab dengan logika pekerjaan, dan kinerja tinggi mainframe pada tugas-tugas khusus ini bisa sulit untuk ditiru jika dibandingkan dengan komoditas, digeneralisasi. CPUs Menulis ulang pekerjaan ini bisa menjadi usaha besar dan membutuhkan upaya yang signifikan.

Blu Age mengubah beban kerja mainframe lama menjadi kode Java modern, yang kemudian dapat Anda jalankan sebagai wadah.

Pola ini menyediakan contoh arsitektur tanpa server untuk menjalankan aplikasi kontainer yang telah dimodernisasi dengan alat Blu Age. File HashiCorp Terraform yang disertakan akan membangun arsitektur aman untuk orkestrasi kontainer Blu Age, mendukung tugas batch dan layanan waktu nyata.

Untuk informasi selengkapnya tentang memodernisasi beban kerja Anda dengan menggunakan layanan Blu Age dan AWS, lihat publikasi AWS Prescriptive Guidance ini:

- Menjalankan beban kerja mainframe yang telah dimodernisasi dengan Blu Age pada infrastruktur tanpa server AWS
- Kontainerisasi beban kerja mainframe yang telah dimodernisasi oleh Blu Age

Untuk bantuan menggunakan Blu Age untuk memodernisasi beban kerja mainframe Anda, hubungi tim Blu Age dengan memilih Hubungi pakar kami di situs web Blu Age. Untuk bantuan dalam memigrasikan beban kerja modern Anda ke AWS, mengintegrasikannya dengan layanan AWS, dan memindahkannya ke produksi, hubungi manajer akun AWS Anda atau isi formulir AWS Professional Services.

Prasyarat dan batasan

Prasyarat

- Contoh aplikasi Blu Age dalam wadah yang disediakan oleh beban kerja mainframe Containerize yang telah dimodernisasi oleh pola Blu Age. Aplikasi sampel menyediakan logika untuk menangani pemrosesan input dan output untuk aplikasi modern, dan dapat berintegrasi dengan arsitektur ini.
- Terraform diperlukan untuk menyebarkan sumber daya ini.

Batasan

- Amazon Elastic Container Service (Amazon ECS) membatasi sumber daya tugas yang dapat disediakan untuk wadah. Sumber daya ini termasuk CPU, RAM, dan penyimpanan. Misalnya, saat menggunakan Amazon ECS dengan AWS Fargate, batas sumber daya tugas berlaku.

Versi produk

Solusi ini diuji dengan versi berikut:

- Terraform 1.3.6
- Penyedia AWS Terraform 4.46.0

Arsitektur

Tumpukan teknologi sumber

- Usia Blu
- Terraform

Tumpukan teknologi target

- Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL
- AWS Backup
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- AWS Identity and Access Management Service (IAM)
- Server Manajemen Kunci AWS (AWS KMS)
- AWS Secrets Manager
- Amazon Simple Notification Service (Amazon SNS)

- Amazon Simple Storage Service (Amazon S3)
- AWS Step Functions
- AWS Systems Manager

Arsitektur target

Diagram berikut menunjukkan arsitektur solusi.

1. Solusinya menerapkan peran IAM berikut:
   - Peran tugas batch
   - Peran eksekusi tugas Batch
   - Peran tugas layanan
   - Peran eksekusi tugas layanan
   - Peran Step Functions
   - Peran AWS Backup
   - Peran Pemantauan yang Ditingkatkan RDS.

   Peran tersebut sesuai dengan prinsip akses yang paling tidak diistimewakan.
2. Amazon ECR digunakan untuk menyimpan gambar kontainer yang diatur oleh pola ini.
3. AWS Systems Manager Parameter Store menyediakan data konfigurasi tentang setiap lingkungan ke definisi tugas Amazon ECS saat runtime.
4. AWS Secrets Manager menyediakan data konfigurasi sensitif tentang lingkungan ke definisi tugas Amazon ECS saat runtime. Data telah dienkripsi oleh AWS KMS.
5. Modul Terraform membuat definisi tugas Amazon ECS untuk semua tugas real-time dan batch.
6. Amazon ECS menjalankan tugas batch dengan menggunakan AWS Fargate sebagai mesin komputasi. Ini adalah tugas berumur pendek, dimulai seperti yang dipersyaratkan oleh AWS Step Functions.
7. Amazon Aurora PostgreSQL kompatibel menyediakan database untuk mendukung aplikasi modern. Ini menggantikan database mainframe seperti IBM Db2 atau IBM IMS DB.
8. Amazon ECS menjalankan layanan berumur panjang untuk memberikan beban kerja real-time yang dimodernisasi. Aplikasi stateless ini berjalan secara permanen dengan kontainer yang tersebar di Availability Zone.

9. Network Load Balancer digunakan untuk memberikan akses ke beban kerja real-time. Network Load Balancer mendukung protokol sebelumnya, seperti IBM CICS. Atau, Anda dapat menggunakan Application Load Balancer dengan beban kerja berbasis HTTP.
10. Amazon S3 menyediakan penyimpanan objek untuk input dan output pekerjaan. Wadah harus menangani operasi tarik dan dorong ke Amazon S3 untuk menyiapkan direktori kerja untuk aplikasi Blu Age.
11. Layanan AWS Step Functions digunakan untuk mengatur menjalankan tugas Amazon ECS untuk memproses beban kerja batch.
12. Topik SNS untuk setiap beban kerja batch digunakan untuk mengintegrasikan aplikasi modern dengan sistem lain, seperti email, atau untuk memulai tindakan tambahan, seperti mengirimkan objek keluaran dari Amazon S3 ke FTP.

> **Note**
>
> Secara default, solusinya tidak memiliki akses ke internet. Pola ini mengasumsikan bahwa virtual private cloud (VPC) akan terhubung ke jaringan lain menggunakan layanan seperti AWS Transit Gateway. Dengan demikian, beberapa titik akhir VPC antarmuka digunakan untuk memberikan akses ke layanan AWS yang digunakan oleh solusi. Untuk mengaktifkan akses internet langsung, Anda dapat menggunakan sakelar di modul Terraform untuk mengganti titik akhir VPC dengan gateway internet dan sumber daya terkait.

Otomatisasi dan skala

Penggunaan sumber daya tanpa server di seluruh pola ini membantu memastikan bahwa, dengan memperkecil, ada beberapa batasan pada skala desain ini. Ini mengurangi kekhawatiran tetangga yang bising, seperti persaingan untuk sumber daya komputasi yang mungkin dialami pada mainframe asli. Batch task dapat dijadwalkan untuk berjalan secara bersamaan sesuai kebutuhan.

Wadah individu dibatasi oleh ukuran maksimum yang didukung oleh Fargate. Untuk informasi selengkapnya, lihat bagian CPU tugas dan memori di dokumentasi Amazon ECS.

Untuk menskalakan beban kerja real-time secara horizontal, Anda dapat menambahkan kontainer.

Alat

Layanan AWS

- Amazon Aurora PostgreSQL Compatible Edition adalah mesin database relasional yang dikelola sepenuhnya dan sesuai dengan ACID yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL.
- AWS Backup adalah layanan terkelola penuh yang membantu Anda memusatkan dan mengotomatiskan perlindungan data di seluruh Layanan AWS, di cloud, dan di tempat.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Step Functions adalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi yang penting bagi bisnis.
- AWS Systems Manager Parameter Store menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia.

Layanan lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud. Pola ini menggunakan Terraform untuk membuat arsitektur sampel.

Repositori kode

Kode sumber untuk pola ini tersedia di repositori GitHub Blu Age Sample ECS Infrastructure (Terraform).

Praktik terbaik

- Untuk lingkungan pengujian, gunakan fitur seperti `forceDate` opsi untuk mengonfigurasi aplikasi modern untuk menghasilkan hasil pengujian yang konsisten dengan selalu berjalan selama periode waktu yang diketahui.
- Sesuaikan setiap tugas satu per satu untuk mengkonsumsi jumlah sumber daya yang optimal. Anda dapat menggunakan Amazon CloudWatch Container Insights untuk mendapatkan panduan tentang potensi kemacetan.

Epik

Mempersiapkan lingkungan untuk penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning kode sumber solusi.	Kloning kode solusi dari GitHub proyek.	DevOps insinyur
Bootstrap lingkungan dengan menerapkan sumber daya untuk menyimpan status Terraform.	1. Buka jendela terminal, dan konfirmasikan bahwa Terraform diinstal dan AWS kredensialnya tersedia. 2. Arahkan ke `bootstrap-terraform` folder. 3. Edit file `main.tf` jika Anda ingin mengubah nama bucket S3 (`<accountId>-terraform-backend`) dan tabel Amazon `terraform-lock` DynamoDB ().	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Jalankan `terraform apply` perintah untuk menyebarkan sumber daya. Catat bucket S3 dan nama tabel DynamoDB.	

Menyebarkan infrastruktur solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau dan perbarui konfigura si Terraform.	Di direktori root, buka file `main.tf`, tinjau isinya, dan pertimbangkan untuk melakukan pembaruan berikut: 1. Perbarui Wilayah AWS dengan mencari dan mengganti string `eu-west-1` dengan Wilayah yang diinginkan yang ingin Anda gunakan. 2. Perbarui nama bucket di `Terraform Backend` blok jika default diubah dalam epik sebelumnya. 3. Perbarui `dynamodb_table` nilai jika default diubah dalam epik sebelumnya. 4. Perbarui nilai `stack_prefix` variabel ke string yang Anda inginkan. String ini akan ditambahkan ke	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nama semua sumber daya yang dibuat oleh pola ini. 5. Perbarui nilai `vpc_cidr` Ini harus setidaknya rentang `/24` alamat. 6. Tinjau `Locals` bagian ini. Ini digunakan untuk mendefinisikan tugas-tugas Zaman Blu yang akan digunakan. Solusinya akan mengulangi objek daftar`bluage_batch_modules` , membuat sumber daya terkait (mesin status Step Functions, definisi tugas, dan topik SNS) untuk setiap elemen daftar. Dalam beberapa kasus, Anda mungkin ingin menyesuaikan variabel untuk lingkungan yang berbeda. Misalnya, untuk memaksa runtime di lingkungan pengujian, Anda dapat mengubah nilai `force_execution_time` variabel. 7. Untuk mengaktifkan akses internet, ubah nilai `direct_internet_access_required` dari `false` ke`true`. Ini akan menyebarkan gateway	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	internet, bersama dengan gateway NAT dan tabel rute yang mengaktifkan akses internet publik untuk infrastruktur. Secara default, solusinya akan menyebarkan titik akhir VPC antarmuka ke VPC tanpa akses internet langsung. 8. Untuk memberikan akses ke beban kerja client-server yang dilayani melalui Elastic Load Balancing, perbarui nilai `additional_nlb_igress_cidrs` dengan jaringan CIDR yang seharusnya diizinkan.	
Terapkan file Terraform.	Dari terminal Anda, jalankan `terraform apply` perintah untuk menyebarkan semua sumber daya. Tinjau perubahan yang dihasilkan oleh Terraform, dan masukkan ya untuk memulai build. Perhatikan bahwa dibutuhkan waktu lebih dari 15 menit untuk menyebarkan infrastruktur ini.	DevOps insinyur

(Opsional) Menyebarkan aplikasi kontainer Blu Age yang valid

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dorong gambar wadah Blu Age ke Amazon ECR.	Dorong wadah ke repositori Amazon ECR yang Anda buat di epik sebelumnya. Untuk petunjuk, lihat dokumentasi Amazon ECR. Buat catatan URI gambar kontainer.	DevOps insinyur
Perbarui Terraform untuk mereferensikan gambar wadah Blu Age.	Perbarui file `main.tf` untuk mereferensikan gambar kontainer yang Anda unggah.	DevOps insinyur
Menerapkan ulang file Terraform.	Dari terminal Anda, jalankan `terraform apply` untuk menyebarkan semua sumber daya. Tinjau pembaruan yang disarankan dari Terraform, lalu masukkan ya untuk melanjutk an penerapan.	DevOps insinyur

Sumber daya terkait

- Usia Blu
- Menjalankan beban kerja mainframe yang telah dimodernisasi dengan Blu Age pada infrastruktur tanpa server AWS
- Kontainerisasi beban kerja mainframe yang telah dimodernisasi oleh Blu Age

Hasilkan wawasan z/OS data Db2 dengan menggunakan dan AWS Mainframe Modernization Amazon Q di Quick Sight

Shubham Roy, Roshna Razack, dan Santosh Kumar Singh, Amazon Web Services

Ringkasan

Catatan: AWS Mainframe Modernization Layanan (pengalaman Lingkungan Runtime Terkelola) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Jika organisasi Anda menghosting data penting bisnis di lingkungan mainframe IBM Db2, mendapatkan wawasan dari data tersebut sangat penting untuk mendorong pertumbuhan dan inovasi. Dengan membuka kunci data mainframe, Anda dapat membangun intelijen bisnis yang lebih cepat, aman, dan terukur untuk mempercepat pengambilan keputusan, pertumbuhan, dan inovasi berbasis data di Amazon Web Services () Cloud.AWS

Pola ini menyajikan solusi untuk menghasilkan wawasan bisnis dan membuat narasi yang dapat dibagikan dari data mainframe di IBM Db2 untuk tabel. z/OS Perubahan data mainframe dialirkan ke topik Amazon Managed Streaming for Apache Kafka (Amazon MSK AWS Mainframe Modernization) menggunakan Replikasi Data dengan Tepat. Menggunakan konsumsi streaming Amazon Redshift, data topik MSK Amazon disimpan di tabel gudang data Amazon Redshift Tanpa Server untuk analitik di Amazon Quick Sight.

Setelah data tersedia di Quick Sight, Anda dapat menggunakan petunjuk bahasa alami dengan Amazon Q di Quick Sight untuk membuat ringkasan data, mengajukan pertanyaan, dan menghasilkan cerita data. Anda tidak perlu menulis kueri SQL atau mempelajari alat intelijen bisnis (BI).

Konteks bisnis

Pola ini menyajikan solusi untuk analisis data mainframe dan kasus penggunaan wawasan data. Dengan menggunakan pola, Anda membangun dasbor visual untuk data perusahaan Anda. Untuk mendemonstrasikan solusinya, pola ini menggunakan perusahaan perawatan kesehatan yang menyediakan rencana medis, gigi, dan penglihatan kepada anggotanya di AS. Dalam contoh ini,

demografi anggota dan informasi rencana disimpan di IBM Db2 untuk tabel data. z/OS Dasbor visual menunjukkan hal berikut:

- Distribusi anggota berdasarkan wilayah
- Distribusi anggota berdasarkan jenis kelamin
- Distribusi anggota berdasarkan usia
- Distribusi anggota berdasarkan jenis rencana
- Anggota yang belum menyelesaikan imunisasi preventif

Untuk contoh distribusi anggota menurut wilayah dan anggota yang belum menyelesaikan imunisasi preventif, lihat bagian Informasi tambahan.

Setelah membuat dasbor, Anda menghasilkan cerita data yang menjelaskan wawasan dari analisis sebelumnya. Cerita data memberikan rekomendasi untuk meningkatkan jumlah anggota yang telah menyelesaikan imunisasi preventif.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS. Solusi ini dibangun dan diuji di Amazon Linux 2 di Amazon Elastic Compute Cloud (Amazon EC2).
- Virtual Private Cloud (VPC) dengan subnet yang dapat diakses oleh sistem mainframe Anda.
- Database mainframe dengan data bisnis. Untuk contoh data yang digunakan untuk membangun dan menguji solusi ini, lihat bagian Lampiran.
- Ubah pengambilan data (CDC) diaktifkan pada tabel Db2 z/OS . Untuk mengaktifkan CDC pada Db2 z/OS, lihat dokumentasi IBM.
- Tepatnya Connect CDC untuk z/OS diinstal pada z/OS sistem yang menghosting database sumber. CDC Tepat Connect untuk z/OS gambar disediakan sebagai file zip dalam AWS Mainframe Modernization - Replikasi Data untuk IBM z/OS Amazon Machine Image (AMI). Untuk menginstal Connected Connect CDC z/OS di mainframe, lihat dokumentasi Instalasi yang tepat.

Batasan

- Data Db2 mainframe Anda harus berada dalam tipe data yang didukung oleh Accepently Connect CDC. Untuk daftar tipe data yang didukung, lihat dokumentasi CDC Connect dengan tepat.

- Data Anda di Amazon MSK harus dalam tipe data yang didukung oleh Amazon Redshift. Untuk daftar tipe data yang didukung, lihat dokumentasi Amazon Redshift.
- Amazon Redshift memiliki perilaku dan batas ukuran yang berbeda untuk tipe data yang berbeda. Untuk informasi selengkapnya, lihat dokumentasi Amazon Redshift.
- Data hampir real-time di Quick Sight bergantung pada interval penyegaran yang disetel untuk database Amazon Redshift.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Amazon Q in Quick Sight saat ini tidak tersedia di setiap Wilayah yang mendukung Quick Sight. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Versi produk

- AWS Mainframe Modernization Replikasi Data dengan Tepatnya versi 4.1.44
- Python versi 3.6 atau yang lebih baru
- Apache Kafka versi 3.5.1

Arsitektur

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk menghasilkan wawasan bisnis dari data mainframe dengan menggunakan Replikasi AWS Mainframe Modernization Data dengan Tepatnya dan Amazon Q di Quick Sight.

Diagram menunjukkan alur kerja berikut:

1. Agen Pembaca Log Tepatnya membaca data dari log Db2 dan menulis data ke penyimpanan sementara pada sistem file OMVS di mainframe.
2. Agen Penerbit membaca log Db2 mentah dari penyimpanan sementara.
3. Daemon pengontrol lokal mengautentikasi, mengotorisasi, memantau, dan mengelola operasi.
4. Agen Terapkan diterapkan di Amazon EC2 dengan menggunakan AMI yang telah dikonfigurasi sebelumnya. Ini terhubung dengan Agen Penerbit melalui daemon pengontrol dengan menggunakan TCP/IP. Agen Terapkan mendorong data ke MSK Amazon menggunakan beberapa pekerja untuk throughput tinggi.

5. Para pekerja menulis data ke topik MSK Amazon dalam format JSON. Sebagai target perantara untuk pesan yang direplikasi, Amazon MSK menyediakan kemampuan failover yang sangat tersedia dan otomatis.
6. Amazon Redshift streaming ingestion menyediakan latensi rendah, konsumsi data berkecepatan tinggi dari Amazon MSK ke database Amazon Redshift Serverless. Prosedur tersimpan di Amazon Redshift melakukan rekonsiliasi data perubahan mainframe (insert/update/deletes) ke dalam tabel Amazon Redshift. Tabel Amazon Redshift ini berfungsi sebagai sumber analisis data untuk Quick Sight.
7. Pengguna mengakses data di Quick Sight untuk analisis dan wawasan. Anda dapat menggunakan Amazon Q di Quick Sight untuk berinteraksi dengan data dengan menggunakan petunjuk bahasa alami.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat menskalakannya atau masuk.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- Amazon Managed Streaming for Apache Kafka (Amazon MSK) adalah layanan yang dikelola sepenuhnya yang membantu Anda membangun dan menjalankan aplikasi yang menggunakan Apache Kafka untuk memproses data streaming.
- Amazon Quick Sight adalah layanan intelijen bisnis skala cloud (BI) yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda dalam satu dasbor. Pola ini menggunakan kemampuan BI generatif Amazon Q di Quick Sight.
- Amazon Redshift Serverless adalah opsi tanpa server Amazon Redshift yang membuatnya lebih efisien untuk menjalankan dan menskalakan analitik dalam hitungan detik tanpa perlu menyiapkan dan mengelola infrastruktur gudang data.
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.

Alat-alat lainnya

- Connect CDC mengumpulkan dan mengintegrasikan data dari sistem lama ke dalam platform cloud dan data.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Mainframe_ DataInsights _change_data_reconciliation. Kode ini adalah prosedur yang disimpan di Amazon Redshift. Prosedur tersimpan ini merekonsiliasi perubahan data mainframe (menyisipkan, memperbarui, dan menghapus) dari Amazon MSK ke dalam tabel Amazon Redshift. Tabel Amazon Redshift ini berfungsi sebagai sumber analisis data untuk Quick Sight.

Praktik terbaik

- Ikuti praktik terbaik saat menyiapkan kluster MSK Amazon Anda.
- Ikuti praktik terbaik penguraian data Amazon Redshift untuk meningkatkan kinerja.
- Saat Anda membuat peran AWS Identity and Access Management (IAM) untuk pengaturan Tepatnya, ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.

Epik

Mengatur Replikasi AWS Mainframe Modernization Data dengan Tepat di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Siapkan grup keamanan.	Untuk terhubung ke daemon pengontrol dan kluster MSK Amazon, buat grup keamanan untuk instance tersebut. EC2 Tambahkan aturan masuk dan keluar berikut:
	• Aturan masuk 1:
	  • Untuk Type, pilih Custom TCP.
	  • Untuk Protokol, pilih TCP.
		DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk rentang Port, pilih 2626 (port default untuk daemon pengontrol Tepatnya) atau nomor port daemon pengontrol yang berjalan di mainframe. • Untuk Sumber, pilih blok CIDR. • Aturan masuk 2: • Untuk Jenis, pilih TCP kustom. • Untuk Protokol, pilih SSH. • Untuk rentang Port, pilih 22. • Untuk Sumber, pilih alamat IP atau daftar awalan. • Aturan masuk 3: • Untuk Jenis, pilih TCP kustom. • Untuk Protokol, pilih TCP. • Untuk rentang Port, pilih 9092-9098. • Untuk Sumber, pilih blok CIDR. • Aturan keluar 1: • Untuk Jenis, pilih TCP kustom. • Untuk Protokol, pilih TCP. • Untuk rentang Port, pilih 9092-9098.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• Untuk Sumber, pilih blok CIDR.
	• Aturan keluar 2:
	  • Untuk Jenis, pilih TCP kustom.
	  • Untuk Protokol, pilih TCP.
	  • Untuk rentang Port, pilih 2626 (port default untuk daemon pengontrol Tepatnya) atau nomor port daemon pengontrol yang berjalan di mainframe.
	  • Untuk Sumber, pilih blok CIDR.
	
	Perhatikan nama grup keamanan. Anda perlu mereferensikan nama saat meluncurkan EC2 instance dan mengonfigurasi cluster MSK Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM dan peran IAM.	1. Untuk membuat kebijakan IAM dan peran IAM, ikuti petunjuk dalam dokumentasi.AWS Kebijakan IAM memberikan akses untuk membuat topik di klaster MSK Amazon dan mengirim data ke topik tersebut. 2. Setelah Anda membuat peran IAM, kaitkan kebijakan dengannya. Perhatikan nama peran IAM. Peran ini akan digunakan sebagai profil instans IAM saat Anda meluncurkan EC2 instance.	DevOps insinyur, administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan sebuah EC2 instance.	Untuk menyediakan EC2 instans untuk menjalankan Excently CDC dan terhubung ke Amazon MSK, lakukan hal berikut: 1. Masuk ke AWS Marketplace, dan berlangganan - AWS Mainframe Modernization Replikasi Data untuk IBM z/OS. 2. Pilih AMI dari langganan terkelola, dan pilih Launch new instance. 3. Berikan detail konfigurasi lainnya, seperti nama instance, tipe instance, key pair, VPC, dan subnet. Untuk informasi selengkapnya, lihat EC2 dokumentasi Amazon. 4. Dalam daftar dropdown, pilih grup keamanan yang Anda buat sebelumnya. 5. Di bawah Detail lanjutan, profil instans IAM, Anda harus memilih peran yang Anda buat sebelumnya. 6. Pilih Luncurkan instans.	Administrator AWS, DevOps insinyur

Siapkan Amazon MSK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster MSK Amazon.	Untuk membuat cluster MSK Amazon, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan navigasikan ke Amazon MSK. 2. Pilih Buat klaster. 3. Untuk metode pembuatan cluster, pilih Custom create, dan untuk tipe cluster, pilih Provisioned. 4. Berikan nama untuk cluster. 5. Perbarui pengaturan Cluster sesuai kebutuhan , dan pertahankan default untuk pengaturan lainnya. 6. Perhatikan versi < Kafka>. Anda akan membutuhk annya selama pengaturan klien Kafka. 7. Pilih Berikutnya. 8. Pilih VPC dan subnet yang sama dengan yang Anda gunakan untuk EC2 instance Tepat, dan pilih grup keamanan yang Anda buat sebelumnya. 9. Di bagian Pengatura n keamanan, aktifkan otentikasi berbasis peran	AWS DevOps, Administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	SASL/SCRAM dan IAM. Secara tepat Connect menggunakan CDC SASL/SCRAM (Sederhana Authentication and Security Layer/Salted Challenge Response Mechanism), dan IAM diperlukan untuk menghubungkan ke Amazon Redshift. 10. Pilih Berikutnya. 11. Untuk meninjau, pilih metode pengiriman log Monitoring dan Broker. 12. Pilih Next, lalu pilih Create the cluster. Cluster yang disediakan khas membutuhkan waktu hingga 15 menit untuk membuatny a. Setelah cluster dibuat, statusnya berubah dari Creating menjadi Active.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur SASL/SCRAM otentikasi.	Untuk menyiapkan SASL/SCRAM otentikasi untuk kluster MSK Amazon, lakukan hal berikut: 1. Untuk mengatur rahasia di Secrets Manager, ikuti instruksi dalam AWS dokumentasi. 2. Buka konsol MSK Amazon, dan pilih cluster MSK Amazon yang Anda buat sebelumnya. 3. Pilih tab Properti. 4. Pilih Rahasia asosiasi, Pilih rahasia, pilih kunci rahasia yang Anda buat, lalu pilih Rahasia asosiasi. Anda akan melihat pesan sukses yang mirip dengan yang berikut ini: `Successfully associated 1 secret for cluster <chosen cluster name>` 5. Pilih nama cluster. 6. Dalam ringkasan klaster, pilih Lihat informasi klien. 7. Perhatikan string koneksi endpoint pribadi untuk jenis otentikasi SASL/SCRAM.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat topik MSK Amazon.	Untuk membuat topik MSK Amazon, lakukan hal berikut:
	1. Connect ke EC2 instance yang Anda buat sebelumnya, dan instal pembaruan terbaru dengan menjalankan perintah berikut:
	`sudo yum update -y`
	2. Instal perpustakaan Java dan Kafka dengan menjalankan perintah berikut:
	`sudo yum install -y java-11 librdkafka librdkafka-devel`
	3. Untuk membuat folder bernama `kafka/home/ec2-user` , navigasikan ke folder itu, dan jalankan perintah berikut:
	`mkdir kafka;cd kafka`
	4. Unduh pustaka `kafka` klien ke `kafka` folder, ganti `<YOUR MSK VERSION>` dengan versi Kafka yang Anda catat selama pembuatan cluster MSK Amazon:
		Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`wget https://archive.apache.org/dist/kafka//kafka_2.13-<YOUR MSK VERSION>.tgz`
	5. Untuk mengekstrak file yang diunduh, jalankan perintah berikut, gantiYOUR MSK VERSION>:
	`tar -xzf kafka_2.13-<YOUR MSK VERSION>.tgz`
	6. Untuk menavigasi ke `kafka libs` direktori dan mengunduh file Java IAM Authentication Java Archive (JAR), jalankan perintah berikut, ganti: <YOUR MSK VERSION>
	`cd kafka_2.13-<YOUR MSK VERSION>/libs`
	`wget https://github.com/aws/aws-msk-iam-auth/releases/download/v1.1.1/aws-msk-iam-auth-1.1.1-all.jarkafka`
	7. Untuk menavigasi ke `bin` direktori Kafka dan membuat `client.properties` file, jalankan perintah berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cd /home/ec2-user/kafka/kafka_2.13-<YOUR MSK VERSION>/bin`
	`cat >client.properties`
	
	8. Perbarui `client.properties` file dengan konten berikut:
	
	`security.protocol=SASL_SSL`
	`sasl.mechanism=AWS_MSK_IAM`
	`sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required;`
	`sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler`
	
	9. Untuk membuat topik Kafka, navigasikan ke tempat sampah Kafka dan jalankan perintah berikut, ganti `<kafka broker>` dengan titik akhir pribadi server bootstrap IAM yang Anda catat selama pembuatan cluster MSK Amazon:
	
	`./kafka-topics.sh --bootstrap-server <kafka broker> --`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`command-config client.properties --create --replication-factor 3 –partitions 6 --topic <topic name>` 10. Saat pesan `Created topic <topic name>` muncul, perhatikan nama topik.	

Konfigurasikan Mesin Terapkan dengan Tepat di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan skrip Tepat untuk mereplikasi perubahan data.	Untuk menyiapkan skrip CDC Connect yang tepat untuk mereplikasi data yang diubah dari mainframe ke topik MSK Amazon, lakukan hal berikut: 1. Untuk membuat nama folder dengan tepat dan mengubah ke folder itu, jalankan perintah berikut: `mkdir /home/ec2-user/precisely;cd /home/ec2-user/precisely` 2. Untuk membuat dua folder di dalam tepat dipanggil `scripts` dan`ddls`, dan kemudian ubah ke	Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`scripts` folder, jalankan perintah berikut:
	`mkdir scripts;mkdir ddls;cd scripts`
	3. Untuk membuat file bernama `sqdata_kafka_producer.conf` dalam `scripts` folder, jalankan perintah berikut:
	`cat >sqdata_kafka_producer.conf`
	4. Perbarui `sqdata_kafka_producer.conf` file dengan konten berikut:
	`builtin.features=SASL_SCRAM`
	`security.protocol=SASL_SSL`
	`sasl.mechanism=SCRAM-SHA-512`
	`sasl.username=<User Name>`
	`sasl.password=<Password>`
	`metadata.broker.list=<SASL/SCRAM Bootstrap servers>`
	Note
	Perbarui `<SASL/SCRAM Bootstrap`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`servers>` dengan daftar SASL/SCRAM broker MSK Amazon yang sebelumnya Anda konfigurasikan. Perbarui `<User Name>` dan `<Password>` dengan nama pengguna dan kata sandi yang Anda atur sebelumnya di Secrets Manager.
	
	5. Buat `script.sqd` file di `scripts` folder.
	
	`cat >script.sqd`
	
	Apply Engine digunakan `script.sqd` untuk memproses data sumber dan mereplikasi data sumber ke target. Untuk contoh Menerapkan skrip Engine, lihat bagian Informasi tambahan.
	
	6. Untuk mengubah ke `ddls` folder dan membuat file.ddl untuk setiap tabel Db2, jalankan perintah berikut:
	
	`cd /home/ec2-user/precisely/ddls`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>cat >mem_details.ddl
	cat >mem_plans.ddl</pre>
	Misalnya file.ddl, lihat bagian Informasi tambahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hasilkan kunci ACL jaringan.	Untuk menghasilkan kunci network access control list (network ACL), lakukan hal berikut:
	
	1. Untuk mengekspor jalur `sqdata` instalasi, jalankan perintah berikut:
	
	`export PATH=$PATH:/usr/sbin:/opt/precisely/di/sqdata/bin`
	
	2. Untuk mengubah ke `/home/ec2-user` direktori dan menghasilkan kunci ACL jaringan, jalankan perintah berikut:
	
	`cd /home/ec2-user`
	`sqdutil keygen --force`
	
	Setelah kunci publik dan pribadi dihasilkan, pesan berikut ditampilkan:
	
	`SQDUT04I Generating a private key in file /home/ec2-user/.nacl/id_nacl`
	`SQDC017I sqdutil(pid=27344) terminated successfully`
		Arsitek cloud, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Perhatikan kunci publik yang dihasilkan yang disimpan di `.nacl` folder.	

Siapkan lingkungan sumber mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan default di layar ISPF.	Untuk mengonfigurasi pengaturan default di Fasilitas Produktivitas Sistem Interakti f (ISPF), ikuti petunjuk dalam dokumentasi Tepatnya.	Administrator sistem mainframe
Konfigurasikan daemon pengontrol.	Untuk mengkonfigurasi daemon pengontrol, lakukan hal berikut: 1. Pada layar Menu Utama SQData z/OS, pilih opsi 2. 2. Pada layar Tambahkan Daemon ke Daftar, di bidang Nama Daemon, masukkan nama untuk daemon, lalu tekan Enter.	Administrator sistem mainframe
Mengonfigurasi penerbit.	Untuk mengkonfigurasi penerbit, lakukan hal berikut: 1. Pada layar Menu Utama SQData z/OS, pilih opsi 3. Ini membawa Anda ke layar Ringkasan Capture/P ublisher.	Administrator sistem mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Pilih opsi untuk menambahkan file CAB. Ini akan membawa Anda ke Tambahkan File CAB ke Daftar layar. 3. Di bidang Nama, masukkan nama untuk file CAB. Untuk Db2, masukkan Type as. `D` 4. Tekan Enter. Ini membawa Anda ke layar Create New Db2 Capture CAB File. 5. Di bidang ZfS Dir, tentukan titik pemasangan penyimpanan. 6. Tekan Enter untuk menyimpan dan melanjutkan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file konfigurasi daemon.	Untuk memperbarui detail penerbit di file konfigurasi daemon pengontrol, lakukan hal berikut: 1. Pada layar Menu Utama SQData z/OS, pilih opsi 2. 2. Masukkan S di dekat daemon yang Anda buat untuk melihat detail daemon. 3. Masuk1, lalu tekan Enter untuk mengedit file agen. 4. Tambahkan detail file CAB Anda. Contoh berikut menunjukkan rincian untuk file CAB bernama`DB2ZT0MSK` . Gunakan ID pengguna mainframe Anda alih-alih. `<userid>` `ÝDB2ZT0MSK¨ type=capture cab=/u/<userid>/sqdata/DB2ZT0MSK.cab` 5. Tekan F3. 6. Masukkan 2 untuk mengedit file ACL. Tambahkan Anda `userid` ke file `acl` konfigurasi seperti yang ditunjukkan pada contoh berikut:	Administrator sistem mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Ýacls¨ prod=admin,<userid>` 7. Tekan F3 untuk menyimpan dan keluar.	
Buat pekerjaan untuk memulai daemon controller.	Untuk membuat pekerjaan, lakukan hal berikut: 1. Di Opsi, masukkan`G`. 2. Masukkan Kartu JOB, pustaka pekerjaan dan proc, dan detail `Db2 load` perpustakaan. 3. Masukkan detail file ACL jaringan, dan masukkan opsi 2 untuk menghasil kan file bahasa kontrol pekerjaan (JCL) di pustaka pekerjaan yang ditentukan.	Administrator sistem mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan file JCL penerbit tangkapan.	Untuk membuat file JCL penerbit tangkapan, lakukan hal berikut: 1. Pada layar Menu Utama SQData z/OS, pilih opsi 3. Ini membawa Anda ke layar Ringkasan Capture/Publisher. 2. Masukkan `S` di sebelah file CAB untuk memilihnya. Ini membawa Anda ke layar Capture/Publisher Detail Db2. 3. Di Opsi, `G` masukkan opsi untuk menghasilkan `capture/publisher` pekerjaan. 4. Masukkan pustaka kartu JOB, pekerjaan, dan prosedur, dan detail pustaka muat Db2. 5. Untuk membuat pekerjaan, pilih opsi 4. Pekerjaan dibuat di pustaka pekerjaan yang ditentukan di perpustakaan pekerjaan.	Administrator sistem mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Periksa dan perbarui CDC.	1. Periksa `DATACAPTURE` bendera untuk tabel Db2 dengan menjalankan kueri berikut, ubah `<table name>` ke nama tabel Db2 Anda:
	`SELECT DATACAPTURE FROM SYSIBM.SYSTABLES WHERE NAME='<table name>';`
	Konfirmasikan bahwa hasilnya menunjukkan `DATACAPTURE` sebagai`Y`.
	2. Jika `DATACAPTURE` tidak`Y`, jalankan kueri berikut untuk mengaktifkan CDC pada tabel Db2, ubah `<table name>` ke nama tabel Db2 Anda:
	`ALTER TABLE <table name> DATA CAPTURE CHANGES;`
		Administrator sistem mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim file JCL.	Kirimkan file JCL berikut yang Anda konfigurasikan pada langkah sebelumnya: • File JCL untuk memulai daemon pengontrol • File JCL untuk mulai menangkap dan mempublik asikan Setelah Anda mengirimkan file JCL, Anda dapat memulai Apply Engine di Tepat pada EC2 instance.	Administrator sistem mainframe

Jalankan dan validasi CDC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai Apply Engine dan validasi CDC.	Untuk memulai Apply Engine pada EC2 instance dan memvalidasi CDC, lakukan hal berikut: 1. Untuk terhubung ke EC2 instance, ikuti instruksi dalam AWS dokumentasi. 2. Ubah ke direktori yang berisi `script.sqd` file: `cd /home/ec2-user/precisely/scripts`	Arsitek cloud, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Untuk memulai Apply Engine, jalankan perintah `sqdeng` start berikut: `sqdeng -s script.sqd --identity=/home/ec2-user/.nacl/id_nacl` Apply Engine akan mulai menunggu pembaruan dari sumber mainframe. 4. Untuk menguji CDC, buat beberapa sisipan catatan atau pembaruan di tabel Db2. 5. Validasi bahwa log Apply Engine menunjukkan jumlah catatan yang diambil dan ditulis ke target.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi catatan tentang topik MSK Amazon.	Untuk membaca pesan dari topik Kafka, lakukan hal berikut:
	1. Untuk mengubah ke `bin` direktori jalur instalasi klien Kafka pada EC2 instance, jalankan perintah berikut, ganti `<Kafka version>` dengan versi Anda:
	`cd /home/ec2-user/kafka/kafka_2.13-<Kafka version>/bin`
	2. Untuk memvalidasi CDC Db2 yang ditulis sebagai pesan dalam topik Kafka, jalankan perintah berikut, ganti `<kafka broker>` dan `<Topic Name>` dengan topik yang Anda buat sebelumnya:
	`./kafka-console-consumer.sh --bootstrap-server <kafka broker>:9098 --topic <Topic Name> --from-beginning --consumer.config client.properties`
	3. Validasi bahwa pesan cocok dengan jumlah
		Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	catatan yang diperbarui dalam tabel Db2.	

Menyimpan data perubahan mainframe di gudang data Amazon Redshift Tanpa Server

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Amazon Redshift Tanpa Server.	Untuk membuat gudang data Amazon Redshift Tanpa Server, ikuti petunjuk dalam dokumentasi.AWS Di dasbor Amazon Redshift Tanpa Server, validasi bahwa namespace dan workgroup telah dibuat dan tersedia. Untuk contoh pola ini, prosesnya mungkin memakan waktu 2-5 menit.	Insinyur data
Siapkan peran IAM dan kebijakan kepercayaan yang diperlukan untuk streaming konsumsi.	Untuk mengatur konsumsi streaming Amazon Redshift Tanpa Server dari Amazon MSK, lakukan berikut: 1. Buat kebijakan IAM untuk Amazon Redshift untuk mengakses Amazon MSK. Mengganti `[region]` dengan MSK Wilayah AWS untuk Amazon, `[account-id]` dengan Akun AWS ID Anda, dan `[msk-cluster-name]` dengan	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nama cluster MSK Amazon, jalankan kode berikut: `{"Version": "2012-10-17", "Statement": [{"Sid": "MSKIAMpolicy","Effect": "Allow","Action": ["kafka-cluster:ReadData","kafka-cluster:DescribeTopic","kafka-cluster:Connect"],"Resource": ["arn:aws:kafka:[region]:[account-id]:cluster/[msk-cluster-name]/*","arn:aws:kafka:[region]:[account-id]:topic/[msk-cluster-name]/*"]},{"Effect": "Allow","Action": ["kafka-cluster:AlterGroup","kafka-cluster:DescribeGroup"],"Resource": ["arn:aws:kafka:[region]:[account-id]:group/[msk-cluster-name]/*"]}]}` Anda dapat menemukan nama cluster dan Amazon	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Resource Name (ARN) di konsol Amazon MSK. Di konsol, pilih Ringkasan cluster, lalu pilih ARN. 2. Untuk membuat peran IAM dan melampirkan kebijakan , ikuti instruksi dalam AWS dokumentasi. 3. Untuk melampirkan peran IAM ke namespace Amazon Redshift Tanpa Server, lakukan berikut: a. Masuk ke konsol, dan buka konsol Amazon Redshift di. https://console.aws.amazon.com/redshiftv2/ b. Pilih dasbor Tanpa Server. c. Pilih Namespace. d. Pilih tab Keamanan dan enkripsi. e. Pilih Izin, dan lampirkan peran IAM yang Anda buat. 4. Di grup keamanan Amazon Redshift Tanpa Server, buat aturan masuk dengan detail berikut: • Untuk Jenis, pilih TCP kustom. • Untuk Protokol, pilih TCP.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk rentang Port, pilih 9098, 9198. • Untuk Sumber, pilih grup keamanan MSK Amazon. 5. Di grup keamanan MSK Amazon Anda, buat aturan masuk dengan detail berikut: • Untuk Jenis, pilih TCP kustom. • Untuk Protokol, pilih TCP. • Untuk rentang Port, pilih 9098, 9198. • Untuk Sumber, pilih grup keamanan Amazon Redshift. Pola ini menggunakan port untuk otentikasi IAM untuk konfigurasi Amazon Redshift dan Amazon MSK. Untuk informasi selengkap nya, lihat AWS dokumentasi (langkah 2). 6. Aktifkan perutean VPC yang disempurnakan untuk workgroup Amazon Redshift Serverless. Lihat informasi yang lebih lengkap dalam dokumentasi AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hubungkan Amazon Redshift Tanpa Server ke Amazon MSK.	Untuk terhubung ke topik MSK Amazon, buat skema eksternal di Amazon Redshift Tanpa Server. Di Amazon Redshift query editor v2, jalankan perintah SQL berikut, ganti `'iam_role_arn'` dengan peran yang Anda buat sebelumnya dan ganti `'MSK_cluster_arn` 'dengan ARN untuk cluster Anda. `CREATE EXTERNAL SCHEMA member_schema FROM MSK IAM_ROLE 'iam_role _arn' AUTHENTICATION iam URI 'MSK_cluster_arn';`	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tampilan yang terwujud.	Untuk menggunakan data dari topik MSK Amazon di Amazon Redshift Tanpa Server, buat tampilan terwujud. Di Amazon Redshift query editor v2, jalankan perintah SQL berikut, ganti <MSK_Topic_name> dengan nama topik MSK Amazon Anda. `CREATE MATERIALIZED VIEW member_view AUTO REFRESH YES AS SELECT kafka_partition, kafka_offset, refresh_time, json_parse(kafka_value) AS Data FROM member_schema.<MSK_Topic_name> WHERE CAN_JSON_PARSE(kafka_value);`	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat tabel target di Amazon Redshift.	Tabel Amazon Redshift memberikan masukan untuk Quick Sight. Pola ini menggunakan tabel `member_dtls` dan`member_plans` , yang cocok dengan tabel sumber Db2 pada mainframe.
	
	Untuk membuat dua tabel di Amazon Redshift, jalankan perintah SQL berikut di editor kueri Amazon Redshift v2:
	
	`-- Table 1: members_dtls`
	`CREATE TABLE members_dtls (`
	`memberid INT ENCODE AZ64,`
	`member_name VARCHAR(100) ENCODE ZSTD,`
	`member_type VARCHAR(50) ENCODE ZSTD,`
	`age INT ENCODE AZ64,`
	`gender CHAR(1) ENCODE BYTEDICT,`
	`email VARCHAR(100) ENCODE ZSTD,`
	`region VARCHAR(50) ENCODE ZSTD`
	`) DISTSTYLE AUTO;`
	
	`-- Table 2: member_plans`
	`CREATE TABLE member_plans (`
	`memberid INT ENCODE AZ64,`
		Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>medical_plan CHAR(1)
	ENCODE BYTEDICT,
	dental_plan CHAR(1)
	ENCODE BYTEDICT,
	vision_plan CHAR(1)
	ENCODE BYTEDICT,
	preventive_immuniz
	ation VARCHAR(50)
	ENCODE ZSTD
	) DISTSTYLE AUTO;</pre>	

Buat prosedur tersimpan di Amazon Redshift.	Pola ini menggunak an prosedur tersimpan untuk menyinkro nkan data perubahan (`INSERT`,`UPDATE`,`DELETE`) dari mainframe sumber ke tabel gudang data Amazon Redshift target untuk analitik di Quick Sight. Untuk membuat prosedur tersimpan di Amazon Redshift, gunakan editor kueri v2 untuk menjalankan kode prosedur tersimpan yang ada di repositori. GitHub	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Baca dari tampilan terwujud streaming dan muat ke tabel target.	Prosedur tersimpan membaca perubahan data dari tampilan streaming yang terwujud dan memuat perubahan data ke tabel target. Untuk menjalankan prosedur yang disimpan, gunakan perintah berikut: `call SP_Members_Load();` Anda dapat menggunakan Amazon EventBridge untuk menjadwalkan pekerjaan di gudang data Amazon Redshift untuk memanggil prosedur tersimpan ini berdasarkan persyaratan latensi data Anda. EventBridge menjalankan pekerjaan pada interval tetap. Untuk memantau apakah panggilan sebelumnya ke prosedur selesai, Anda mungkin perlu menggunakan mekanisme seperti mesin AWS Step Functionsstatus. Untuk informasi selengkapnya, lihat sumber daya berikut: • Membuat EventBridge aturan Amazon yang berjalan sesuai jadwal • Mempercepat orkestrasi proses ELT menggunakan dan Amazon AWS Step	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Functions Redshift Data API. Pilihan lain adalah menggunakan editor kueri Amazon Redshift v2 untuk menjadwalkan penyegaran. Untuk informasi selengkapnya, lihat Menjadwalkan kueri dengan editor kueri v2.	

Connect Quick Sight ke data di Amazon Redshift

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur Quick Sight.	Untuk mengatur Quick Sight, ikuti instruksi dalam AWS dokumentasi.	Insinyur migrasi
Siapkan koneksi aman antara Quick Sight dan Amazon Redshift.	Untuk mengatur koneksi yang aman antara Quick Sight dan Amazon Redshift, lakukan hal berikut 1. Untuk mengotorisasi koneksi dari Quick Sight ke Amazon Redshift, buka konsol Amazon Redshift, dan tambahkan aturan masuk di grup keamanan Amazon Redshift. Aturan harus mengizinkan lalu lintas ke port 5439 (port Redshift default) dari rentang CIDR tempat Anda	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengatur Quick Sight. Untuk daftar Wilayah AWS dan alamat IP mereka, lihat Didukung Wilayah AWS untuk Penglihatan Cepat. Di konsol Amazon Redshift, pilih Workgroup, Akses data, Jaringan, dan keamanan, dan aktifkan akses Publik.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kumpulan data untuk Quick Sight.	Untuk membuat kumpulan data Quick Sight dari Amazon Redshift, lakukan hal berikut: 1. Pada konsol Quick Sight, di panel navigasi, pilih Datasets. 2. Pada halaman Datasets, pilih Kumpulan data baru. 3. Pilih sambungan Manual Redshift. 4. Di jendela sumber data Redshift Baru, masukkan informasi koneksi: • Untuk nama sumber Data, masukkan nama untuk sumber data Amazon Redshift. • Untuk server Database, masukkan titik akhir cluster Amazon Redshift. Anda bisa mendapatkan nilai endpoint dari bidang Endpoint di bagian Informasi umum untuk workgroup cluster di dasbor Amazon Redshift Serverless. Alamat server adalah bagian pertama dari titik akhir sebelum titik dua, seperti yang ditunjukkan pada contoh berikut:	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`mfdata-insights.NNNNNNNN.us-east-1.redshift-serverless.amazonaws.com:5439/dev` • Untuk port, masukkan `5439` (port default untuk Amazon Redshift). • Masukkan nama database (setelah garis miring di titik akhir). Dalam hal ini, nama database adalah`dev`. • Untuk Nama Pengguna dan Kata Sandi, masukkan nama pengguna dan kata sandi untuk database Amazon Redshift. 5. Pilih Validasi Koneksi. Jika berhasil, Anda akan melihat tanda centang hijau, yang menunjukkan validasi. Jika validasi gagal, lihat bagian Pemecahan Masalah. 6. Pilih Buat sumber data.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bergabunglah dengan kumpulan data.	Untuk membuat analitik di Quick Sight, gabungkan dua tabel dengan mengikuti instruksi dalam AWS dokumentasi. Di panel Gabung Konfigura si, pilih Left for Join type. Di bawah klausa Gabung, gunakan. `memberid from member_plans = memberid from members_details`	Insinyur migrasi

Dapatkan wawasan bisnis dari data mainframe dengan menggunakan Amazon Q di Quick Sight

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Amazon Q di Quick Sight.	Untuk menyiapkan Amazon Q dalam kemampuan BI generatif Quick Sight, ikuti petunjuk dalam AWS dokumentasi.	Insinyur migrasi
Analisis data mainframe dan buat dasbor visual.	Untuk menganalisis dan memvisualisasikan data Anda di Quick Sight, lakukan hal berikut: 1. Untuk membuat analisis data mainframe, ikuti instruksi dalam AWS dokumentasi. Untuk kumpulan data, pilih	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kumpulan data yang Anda buat. 2. Pada halaman analisis pilih Build visual. 3. Di jendela Buat topik untuk analisis, pilih Perbarui topik yang ada. 4. Dalam daftar tarik-turun Pilih topik, pilih topik yang Anda buat sebelumnya. 5. Pilih Penautan topik. 6. Setelah Anda menautkan topik, pilih Build visual untuk membuka jendela Amazon Q Build a Visual. 7. Di bilah prompt. Tulis pertanyaan analisis Anda. Contoh pertanyaan yang digunakan untuk pola ini adalah sebagai berikut: • Tampilkan distribusi anggota berdasarkan wilayah • Tampilkan distribusi anggota berdasarkan usia • Tampilkan distribusi anggota berdasarkan jenis kelamin • Tampilkan distribusi anggota berdasarkan jenis paket	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Tampilkan anggota belum menyelesaikan imunisasi preventif Setelah memasukkan pertanyaan, pilih Build. Amazon Q di Quick Sight menciptakan visual. 8. Untuk menambahkan visual ke dasbor visual Anda, pilih TAMBAHKAN KE ANALISIS. Setelah selesai, Anda dapat mempublikasikan dasbor untuk dibagikan dengan orang lain di organisasi Anda. Sebagai contoh, lihat Dasbor visual Mainframe di bagian Informasi tambahan.	

Buat cerita data dengan Amazon Q di Quick Sight dari data mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cerita data.	Buat cerita data untuk menjelaskan wawasan dari analisis sebelumnya, dan buat rekomendasi untuk meningkatkan imunisasi preventif bagi anggota:	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Untuk membuat cerita data, ikuti instruksi dalam AWS dokumentasi. 2. Untuk prompt cerita data, gunakan yang berikut ini: `Build a data story about Region with most numbers of members. Also show the member distribution by medical plan, vision plan, dental plan. Recommend how to motivate members to complete immunizat ion. Include 4 points of supportin g data for this pattern.` Anda juga dapat membuat prompt Anda sendiri untuk menghasilkan cerita data untuk wawasan bisnis lainnya. 3. Pilih Tambahkan visual, dan tambahkan visual yang relevan dengan cerita data. Untuk pola ini, gunakan visual yang Anda buat sebelumnya. 4. Pilih Build.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Misalnya keluaran cerita data, lihat Output cerita data di bagian Informasi tambahan.	
Lihat cerita data yang dihasilkan.	Untuk melihat cerita data yang dihasilkan, pilih cerita itu di halaman Cerita data.	Insinyur migrasi
Edit cerita data yang dihasilkan.	Untuk mengubah format, tata letak, atau visual dalam cerita data, ikuti petunjuk dalam dokumentasi.AWS	Insinyur migrasi
Bagikan cerita data.	Untuk berbagi cerita data, ikuti instruksi dalam AWS dokumentasi.	Insinyur migrasi

Pemecahan masalah

Isu	Solusi
Untuk pembuatan kumpulan data Quick Sight ke Amazon Redshift, `Validate Connection` telah salah.	1. Konfirmasikan bahwa grup keamanan yang dilampirkan ke instans Amazon Redshift Tanpa Server memungkinkan lalu lintas masuk dari rentang alamat IP yang terkait dengan Wilayah tempat Anda mengatur Quick Sight. 2. Konfirmasikan bahwa VPC tempat Amazon Redshift Serverless digunakan tersedia untuk umum. 3. Konfirmasikan bahwa Anda menggunakan nama pengguna dan kata sandi yang benar untuk Amazon Redshift. Anda dapat

Isu	Solusi
	mengatur ulang nama pengguna dan kata sandi di konsol Amazon Redshift.
Mencoba menyalakan mesin Apply pada EC2 instance mengembalikan kesalahan berikut: `-bash: sqdeng: command not found`	Ekspor jalur `sqdata` instalasi dengan menjalankan perintah berikut: `export PATH=$PATH:/usr/sbin:/opt/precisely/di/sqdata/bin`
Mencoba memulai Apply Engine mengembalikan salah satu kesalahan koneksi berikut: • `SQDD018E Cannot connect to transfer socket(rc==0x18468). Agent:<Agent Name > Socket:/u/./sqdata/.DB2ZTOMSK.cab.data` • `SQDUR06E Error opening url cdc://<VPC end point name>:2626/DB2ZTOMSK/DB2ZTOMSK : errno:1128 (Unknown error 1128)`	Periksa spool mainframe untuk memastikan bahwa pekerjaan daemon controller berjalan.

Sumber daya terkait

- Hasilkan wawasan dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight (pola)
- Hasilkan wawasan data dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight (demo)
- AWS Mainframe Modernization - Replikasi Data untuk IBM z/OS
- Konsumsi streaming Amazon Redshift ke tampilan yang terwujud

Informasi tambahan

Contoh file.ddl

members_details.ddl

```
CREATE TABLE MEMBER_DTLS (
memberid INTEGER NOT NULL,
member_name VARCHAR(50),
member_type VARCHAR(20),
age INTEGER,
gender CHAR(1),
email VARCHAR(100),
region VARCHAR(20)
);
```

member_plans.ddl

```
CREATE TABLE MEMBER_PLANS (
memberid INTEGER NOT NULL,
medical_plan CHAR(1),
dental_plan CHAR(1),
vision_plan CHAR(1),
preventive_immunization VARCHAR(20)
);
```

Contoh file.sqd

Ganti **<kafka topic name>**dengan nama topik MSK Amazon Anda.

script.sqd

```
-- Name: DB2ZTOMSK: DB2z To MSK JOBNAME DB2ZTOMSK;REPORT EVERY 1;OPTIONS
 CDCOP('I','U','D');-- Source Descriptions
JOBNAME DB2ZTOMSK;
REPORT EVERY 1;
OPTIONS CDCOP('I','U','D');

-- Source Descriptions
BEGIN GROUP DB2_SOURCE;
DESCRIPTION DB2SQL /var/precisely/di/sqdata/apply/DB2ZTOMSK/ddl/mem_details.ddl AS
 MEMBER_DTLS;
DESCRIPTION DB2SQL /var/precisely/di/sqdata/apply/DB2ZTOMSK/ddl/mem_plans.ddl AS
 MEMBER_PLANS;
END GROUP;
-- Source Datastore
DATASTORE cdc://<zos_host_name>/DB2ZTOMSK/DB2ZTOMSK
OF UTSCDC
```

```
AS CDCIN
DESCRIBED BY GROUP DB2_SOURCE ;
-- Target Datastore(s)
DATASTORE 'kafka:///<kafka topic name>/key'
OF JSON
AS TARGET
DESCRIBED BY GROUP DB2_SOURCE;
PROCESS INTO TARGET
SELECT
{
REPLICATE(TARGET)
}
FROM CDCIN;
```

Dasbor visual mainframe

Data visual berikut dibuat oleh Amazon Q di Quick Sight untuk pertanyaan analisis `show member distribution by region`.

Data visual berikut dibuat oleh Amazon Q di Quick Sight untuk pertanyaan tersebut`show member distribution by Region who have not completed preventive immunization, in pie chart`.

Keluaran cerita data

Tangkapan layar berikut menunjukkan bagian dari cerita data yang dibuat oleh Amazon Q di Quick Sight untuk prompt`Build a data story about Region with most numbers of members. Also show the member distribution by age, member distribution by gender. Recommend how to motivate members to complete immunization. Include 4 points of supporting data for this pattern`.

Dalam pendahuluan, cerita data merekomendasikan memilih wilayah dengan anggota terbanyak untuk mendapatkan dampak terbesar dari upaya imunisasi.

Cerita data memberikan analisis nomor anggota untuk empat wilayah. Wilayah Timur Laut, Barat Daya, dan Tenggara memiliki anggota terbanyak.

Kisah data menyajikan analisis anggota berdasarkan usia.

Kisah data berfokus pada upaya imunisasi di Midwest.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Hasilkan wawasan data dengan menggunakan AWS Mainframe Modernization dan Amazon Q di Quick Sight

Shubham Roy, Roshna Razack, dan Santosh Kumar Singh, Amazon Web Services

Ringkasan

Catatan: AWS Mainframe Modernization Layanan (pengalaman Lingkungan Runtime Terkelola) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Jika organisasi Anda menyimpan data penting bisnis di lingkungan mainframe, mendapatkan wawasan dari data tersebut sangat penting untuk mendorong pertumbuhan dan inovasi. Dengan membuka kunci data mainframe, Anda dapat membangun intelijen bisnis yang lebih cepat, aman, dan terukur untuk mempercepat pengambilan keputusan, pertumbuhan, dan inovasi berbasis data di Amazon Web Services () Cloud.AWS

Pola ini menyajikan solusi untuk menghasilkan wawasan bisnis dan membuat narasi yang dapat dibagikan dari data mainframe dengan menggunakan transfer AWS Mainframe Modernization file dengan BMC dan Amazon Q di Quick Sight. Kumpulan data mainframe ditransfer ke Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dengan menggunakan transfer file dengan BMC. AWS Mainframe Modernization AWS Lambda Fungsi memformat dan menyiapkan file data mainframe untuk dimuat ke Quick Sight.

Setelah data tersedia di Quick Sight, Anda dapat menggunakan petunjuk bahasa alami dengan Amazon Q di Quick Sight untuk membuat ringkasan data, mengajukan pertanyaan, dan menghasilkan cerita data. Anda tidak perlu menulis kueri SQL atau mempelajari alat intelijen bisnis (BI).

Konteks bisnis

Pola ini menyajikan solusi untuk analisis data mainframe dan kasus penggunaan wawasan data. Dengan menggunakan pola, Anda membangun dasbor visual untuk data perusahaan Anda. Untuk mendemonstrasikan solusinya, pola ini menggunakan perusahaan perawatan kesehatan yang menyediakan rencana medis, gigi, dan penglihatan kepada anggotanya di AS. Dalam contoh ini,

demografi anggota dan informasi rencana disimpan dalam kumpulan data mainframe. Dasbor visual menunjukkan hal berikut:

- Distribusi anggota berdasarkan wilayah
- Distribusi anggota berdasarkan jenis kelamin
- Distribusi anggota berdasarkan usia
- Distribusi anggota berdasarkan jenis rencana
- Anggota yang belum menyelesaikan imunisasi preventif

Setelah membuat dasbor, Anda menghasilkan cerita data yang menjelaskan wawasan dari analisis sebelumnya. Cerita data memberikan rekomendasi untuk meningkatkan jumlah anggota yang telah menyelesaikan imunisasi preventif.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Kumpulan data mainframe dengan data bisnis
- Akses untuk menginstal agen transfer file di mainframe

Batasan

- File data mainframe Anda harus dalam salah satu format file yang didukung oleh Quick Sight. Untuk daftar format file yang didukung, lihat Sumber data yang didukung.
- Pola ini menggunakan fungsi Lambda untuk mengonversi file mainframe menjadi format yang didukung oleh Quick Sight.

Arsitektur

Diagram berikut menunjukkan arsitektur untuk menghasilkan wawasan bisnis dari data mainframe dengan menggunakan transfer AWS Mainframe Modernization file dengan BMC dan Amazon Q di Quick Sight.

Diagram menunjukkan alur kerja berikut:

1. Dataset mainframe yang berisi data bisnis ditransfer ke Amazon S3 dengan AWS Mainframe Modernization menggunakan transfer file dengan BMC.
2. Fungsi Lambda mengonversi file yang ada di bucket S3 tujuan transfer file ke format nilai yang dipisahkan koma (CSV).
3. Fungsi Lambda mengirimkan file yang dikonversi ke bucket S3 kumpulan data sumber.
4. Data dalam file dicerna oleh Quick Sight.
5. Pengguna mengakses data di Quick Sight. Anda dapat menggunakan Amazon Q di Quick Sight untuk berinteraksi dengan data dengan menggunakan petunjuk bahasa alami.

Alat

Layanan AWS

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Mainframe Modernization transfer file dengan BMC mengonversi dan mentransfer kumpulan data mainframe ke Amazon S3 untuk kasus penggunaan modernisasi, migrasi, dan augmentasi mainframe.
- Amazon Quick Sight adalah layanan BI skala cloud yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda dalam satu dasbor. Pola ini menggunakan kemampuan BI generatif Amazon Q di Quick Sight.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Praktik terbaik

- Saat Anda membuat peran AWS Identity and Access Management (IAM) untuk transfer AWS Mainframe Modernization file dengan BMC dan fungsi Lambda, ikuti prinsip hak istimewa paling sedikit.
- Pastikan kumpulan data sumber Anda telah mendukung tipe data untuk Quick Sight. Jika kumpulan data sumber Anda berisi tipe data yang tidak didukung, konversikan ke tipe data yang didukung. Untuk informasi tentang tipe data mainframe yang tidak didukung dan cara mengonversinya menjadi tipe data yang didukung oleh Amazon Q di Quick Sight, lihat bagian Sumber daya terkait.

Epik

Mengatur transfer AWS Mainframe Modernization file dengan BMC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal agen transfer file.	Untuk menginstal agen transfer AWS Mainframe Modernization file, ikuti instruksi dalam AWS dokumentasi.	Administrator sistem mainframe
Buat bucket S3 untuk transfer file mainframe.	Buat bucket S3 untuk menyimpan file output dari transfer AWS Mainframe Modernization file dengan BMC. Dalam diagram arsitektur, ini adalah bucket tujuan transfer file.	Insinyur migrasi
Buat titik akhir transfer data.	1. Buat bucket S3 untuk mementaskan file mainframe input untuk transfer AWS Mainframe Modernization file dengan BMC. 2. Untuk membuat titik akhir transfer data mainframe, ikuti petunjuk dalam dokumentasi.AWS	Spesialis Modernisasi AWS Mainframe

Konversi ekstensi nama file mainframe untuk integrasi Quick Sight

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket S3 untuk fungsi Lambda untuk menyalin file	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mainframe yang dikonversi dari sumber ke bucket tujuan akhir.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda.	Untuk membuat fungsi Lambda yang mengubah ekstensi file dan menyalin file mainframe ke bucket tujuan, lakukan hal berikut: 1. Masuk ke Konsol Manajemen AWS, dan navigasikan ke AWS Lambda konsol. 2. Pilih Create function, lalu pilih Author dari awal. 3. Untuk nama Fungsi, masukkan nama untuk fungsi Anda. 4. Dalam daftar dropdown Runtime, pilih Python.3.X. 5. Perluas Ubah peran eksekusi default, lalu pilih Buat peran baru dengan izin Lambda dasar. 6. Pilih Buat fungsi. 7. Pilih tab Kode, lalu tempel kode `S3CopyLambda.py` Python yang disediakan di bagian Informasi tambahan. Kode Python dihasilkan dengan menggunakan Amazon Q di lingkungan pengembangan terintegrasi Microsoft Visual Studio (IDE).	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. Edit `destination_bucket_name` ke nama bucket S3 yang Anda buat sebelumnya, dan `change destination_file_key` ke nama file mainframe. 9. Menyebarkan fungsi Lambda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pemicu Amazon S3 untuk menjalankan fungsi Lambda.	Untuk mengonfigurasi pemicu yang memanggil fungsi Lambda, lakukan hal berikut: 1. Di konsol Lambda, buka halaman Fungsi. 2. Pilih fungsi Lambda. 3. Dalam Ikhtisar fungsi, pilih Tambahkan pemicu. 4. Dalam daftar dropdown konfigurasi Trigger, pilih S3. 5. Di bidang Bucket, masukkan nama bucket sumber Anda. 6. Dalam daftar dropdown jenis acara, pilih Semua objek membuat acara. 7. Pilih kotak centang I mengakui bahwa menggunakan bucket S3 yang sama untuk input dan output tidak disarankan, lalu pilih Tambah. Untuk informasi selengkap nya, lihat Tutorial: Menggunak an pemicu Amazon S3 untuk menjalankan fungsi Lambda.	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan izin IAM untuk fungsi Lambda.	Izin IAM diperlukan untuk fungsi Lambda untuk mengakses tujuan transfer file dan bucket S3 kumpulan data sumber. Perbarui kebijakan yang terkait dengan peran eksekusi fungsi Lambda dengan mengizinkan `s3:GetObject` dan `s3:DeleteObject` izin untuk bucket S3 tujuan transfer file dan `s3:PutObject` akses untuk bucket S3 kumpulan data sumber. Untuk informasi selengkapnya, lihat bagian Membuat kebijakan izin di Tutorial: Menggunakan pemicu Amazon S3 untuk menjalankan fungsi Lambda.	Pimpin migrasi

Tentukan tugas transfer data mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas transfer untuk menyalin file mainframe ke bucket S3.	Untuk membuat tugas transfer file mainframe, ikuti instruksi dalam AWS Mainframe Modernization dokumentasi. **Note** Tentukan pengkodean halaman kode sumber	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sebagai IBM1047 dan pengkodean halaman kode target sebagai UTF-8.	
Verifikasi tugas transfer.	Untuk memverifikasi bahwa transfer data berhasil, ikuti instruksi dalam AWS Mainframe Modernization dokumentasi. Konfirmasikan bahwa file mainframe ada di bucket S3 tujuan transfer file.	Pimpin migrasi
Verifikasi fungsi salinan Lambda.	Verifikasi bahwa fungsi Lambda dimulai dan file tersebut disalin dengan ekstensi.csv ke bucket S3 kumpulan data sumber. File csv yang dibuat oleh fungsi Lambda adalah file data input untuk Quick Sight. Misalnya data, lihat `Sample-data-member-healthcare-APG` file di bagian Lampiran.	Pimpin migrasi

Connect Quick Sight ke data mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengatur Quick Sight.	Untuk mengatur Quick Sight, ikuti instruksi dalam AWS dokumentasi.	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kumpulan data untuk Quick Sight.	Untuk membuat kumpulan data untuk Quick Sight, ikuti petunjuk dalam AWS dokumentasi. File data input adalah file mainframe yang dikonversi yang dibuat saat Anda menentukan tugas transfer data mainframe.	Pimpin migrasi

Dapatkan wawasan bisnis dari data mainframe dengan menggunakan Amazon Q di Quick Sight

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Amazon Q di Quick Sight.	Kemampuan ini membutuhkan Enterprise Edition. Untuk mengatur Amazon Q di Quick Sight, lakukan hal berikut: 1. Untuk mendapatkan add-on Amazon Q, ikuti petunjuk Langkah 1: Dapatkan add-on Q dalam dokumentasi.AWS 2. Untuk menggunakan kemampuan BI generatif di Amazon Q, tingkatkan akun pengguna Anda. Ikuti instruksi dalam AWS dokumentasi. 3. Buat topik Amazon Q dengan menggunakan kumpulan data yang Anda buat sebelumnya. Ikuti	Pimpin migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	instruksi dalam AWS dokumentasi. 4. Untuk mengonfigurasi metadata topik agar ramah bahasa alami, ikuti petunjuk dalam dokumentasi.AWS	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Analisis data mainframe dan buat dasbor visual.	Untuk menganalisis dan memvisualisasikan data Anda di Quick Sight, lakukan hal berikut: 1. Untuk membuat analisis data mainframe, ikuti instruksi dalam AWS dokumentasi. Untuk kumpulan data, pilih kumpulan data yang dibuat pada langkah sebelumnya. 2. Pada halaman analisis pilih Build visual. 3. Di jendela Buat topik untuk analisis, pilih Perbarui topik yang ada. 4. Dalam daftar tarik-turun Pilih topik, pilih topik yang Anda buat sebelumnya. 5. Pilih Penautan topik. 6. Setelah Anda menautkan topik, pilih Build visual untuk membuka jendela Amazon Q Build a Visual. 7. Di bar prompt. Tulis pertanyaan analisis Anda. Contoh pertanyaan yang digunakan untuk pola ini adalah sebagai berikut: • Tampilkan distribusi anggota berdasarkan wilayah	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Tampilkan distribusi anggota berdasarkan usia • Tampilkan distribusi anggota berdasarkan jenis kelamin • Tampilkan distribusi anggota berdasarkan jenis paket • Tampilkan anggota belum menyelesaikan imunisasi preventif Setelah memasukkan pertanyaan, pilih Build. Amazon Q di Quick Sight menciptakan visual. 8. Untuk menambahkan visual ke dasbor visual Anda, pilih TAMBAHKAN KE ANALISIS. Setelah selesai, Anda dapat mempublikasikan dasbor untuk dibagikan dengan orang lain di organisasi Anda. Sebagai contoh, lihat Dasbor visual Mainframe di bagian Informasi tambahan.	

Buat cerita data dengan Amazon Q di Quick Sight dari data mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cerita data.	Buat cerita data untuk menjelaskan wawasan dari analisis sebelumnya, dan buat rekomendasi untuk meningkatkan imunisasi preventif bagi anggota: 1. Untuk membuat cerita data, ikuti instruksi dalam AWS dokumentasi. 2. Untuk prompt cerita data, gunakan yang berikut ini: `Build a data story about Region with most numbers of members. Also show the member distribution by medical plan, vision plan, dental plan. Recommend how to motivate members to complete immunization. Include 4 points of supporting data for this pattern.` Anda juga dapat membuat prompt Anda sendiri untuk menghasilkan cerita data	Insinyur migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk wawasan bisnis lainnya. 3. Pilih Tambahkan visual, dan tambahkan visual yang relevan dengan cerita data. Untuk pola ini, gunakan visual yang Anda buat sebelumnya. 4. Pilih Build. 5. Misalnya keluaran cerita data, lihat Output cerita data di bagian Informasi tambahan.	
Lihat cerita data yang dihasilkan.	Untuk melihat cerita data yang dihasilkan, ikuti petunjuk dalam AWS dokumentasi.	Pimpin migrasi
Edit cerita data yang dihasilkan.	Untuk mengubah format, tata letak, atau visual dalam cerita data, ikuti petunjuk dalam dokumentasi.AWS	Pimpin migrasi
Bagikan cerita data.	Untuk berbagi cerita data, ikuti instruksi dalam AWS dokumentasi.	Insinyur migrasi

Pemecahan masalah

Isu	Solusi
Tidak dapat menemukan file mainframe atau kumpulan data yang dimasukkan dalam Data menetapkan kriteria pencarian untuk Buat	1. Pertama, periksa koneksi dengan memilih Titik akhir transfer data pada AWS Mainframe Modernization Transfer dengan

Isu	Solusi
tugas transfer dalam transfer AWS Mainframe Modernization file dengan BMC.	konsol BMC. Jika waktu detak jantung terakhir lebih dari dua menit, koneksi untuk transfer file belum ditetapkan. Jika waktu detak jantung terakhir kurang dari 2 menit untuk agen yang berjalan di mainframe , koneksi ke agen berhasil. Lanjutkan ke langkah 2. 2. Periksa AWS Secrets Manager pengaturannya. Kunci rahasia harus dikonfigurasi di Secrets Manager dengan kunci `userId` (huruf besar I) dengan nilai ID pengguna mainframe dan kunci `password` dengan nilai kata sandi mainframe. Kunci `userId` dan `password` rahasia peka huruf besar/kecil dan harus dimasukkan apa adanya.

Sumber daya terkait

Untuk mengonversi tipe data mainframe seperti PACKED-DECIMAL (COMP-3) atau BINARY (COMP atau COMP-4) ke tipe data yang didukung oleh Quick Sight, lihat pola berikut:

- Konversi dan bongkar data EBCDIC ke ASCII dengan menggunakan Python AWS
- Konversi file mainframe dari format EBCDIC ke format ASCII yang dibatasi karakter di Amazon S3 menggunakan AWS Lambda

Informasi tambahan

S3 CopyLambda .py

Kode Python berikut dihasilkan dengan menggunakan prompt dengan Amazon Q di IDE:

```
#Create a lambda function triggered by S3. display the S3 bucket name and key
import boto3
s3 = boto3.client('s3')
def lambda_handler(event, context):
```

```
print(event)
bucket = event['Records'][0]['s3']['bucket']['name']
key = event['Records'][0]['s3']['object']['key']
print(bucket, key)
#If key starts with object_created, skip copy, print "copy skipped". Return lambda with
 key value.
if key.startswith('object_created'):
print("copy skipped")
return {
'statusCode': 200,
'body': key
}
# Copy the file from the source bucket to the destination bucket.
 Destination_bucket_name = 'm2-filetransfer-final-opt-bkt'. Destination_file_key =
 'healthdata.csv'
copy_source = {'Bucket': bucket, 'Key': key}
s3.copy_object(Bucket='m2-filetransfer-final-opt-bkt', Key='healthdata.csv',
 CopySource=copy_source)
print("file copied")
#Delete the file from the source bucket.
s3.delete_object(Bucket=bucket, Key=key)
return {
'statusCode': 200,
'body': 'Copy Successful'
}
```

Dasbor visual mainframe

Data visual berikut dibuat oleh Amazon Q di Quick Sight untuk pertanyaan analisis `show member distribution by region`.

Data visual berikut dibuat oleh Amazon Q di Quick Sight untuk pertanyaan tersebut`show member distribution by Region who have not completed preventive immunization, in pie chart`.

Keluaran cerita data

Tangkapan layar berikut menunjukkan bagian dari cerita data yang dibuat oleh Amazon Q di Quick Sight untuk prompt `Build a data story about Region with most numbers of members. Also show the member distribution by medical plan, vision plan,`

```
dental plan. Recommend how to motivate members to complete immunization.
Include 4 points of supporting data.
```

Dalam pendahuluan, cerita data merekomendasikan memilih wilayah dengan anggota terbanyak untuk mendapatkan dampak terbesar dari upaya imunisasi.

Kisah data memberikan analisis jumlah anggota untuk tiga wilayah teratas, dan menyebut Southwest sebagai wilayah terkemuka untuk berfokus pada upaya imunisasi.

> **Note**
>
> Wilayah Barat Daya dan Timur Laut masing-masing memiliki delapan anggota. Namun, Southwest memiliki lebih banyak anggota yang tidak sepenuhnya divaksinasi, sehingga memiliki lebih banyak potensi untuk mendapatkan manfaat dari inisiatif untuk meningkatkan tingkat imunisasi.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Menerapkan otentikasi berbasis Microsoft Entra ID dalam aplikasi mainframe modern AWS Blu Age

Vishal Jaswani dan Rimpy Tewani, Amazon Web Services

Ringkasan

> **Note**
>
> AWS Mainframe Modernization Layanan (Managed Runtime Environment experience) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Aplikasi mainframe yang dimodernisasi dengan menggunakan pola refactoring, seperti yang oleh AWS Mainframe Modernization Refactor with AWS Blu Age, memerlukan integrasi mekanisme otentikasi yang cermat ke dalam arsitektur aplikasi baru. Integrasi ini biasanya ditujukan sebagai kegiatan pasca-modernisasi. Tugas dapat menjadi kompleks dan sering melibatkan migrasi atau eksternalisasi sistem otentikasi yang ada untuk menyelaraskan dengan standar keamanan modern dan praktik cloud-native. Pengembang perlu mempertimbangkan cara menerapkan otentikasi secara efektif saat mereka bekerja dalam batasan lingkungan runtime dan pustaka aplikasi yang dimodernisasi. Setelah modernisasi, AWS menyediakan cara untuk memudahkan Anda mengintegrasikan kode modern AWS Blu Age Anda dengan identitas dan sistem manajemen akses seperti Amazon Cognito dan Microsoft Entra ID (sebelumnya dikenal sebagai Azure AD).

Pola ini menjelaskan cara menerapkan mekanisme otentikasi dalam aplikasi modern Anda ketika penyedia otentikasi adalah Microsoft Entra ID, tanpa menghabiskan waktu untuk penelitian dan uji coba. Pola ini menyediakan:

- Pustaka Angular yang teruji lapangan dan relevan dari Microsoft Authentication Library (MSAL) dan dokumentasi Microsoft Entra ID lainnya yang penting untuk implementasi otentikasi.
- Konfigurasi yang diperlukan pada AWS Blu Age Runtime untuk mengaktifkan Spring Security dengan menggunakan 2.0. OAuth
- Pustaka yang menangkap identitas pengguna yang diautentikasi dan meneruskannya ke Blu Age AWS Runtime.

- Langkah-langkah keamanan yang kami rekomendasikan untuk diterapkan.
- Kiat pemecahan masalah untuk masalah yang umum ditemui dengan pengaturan Microsoft Entra ID.

> **Note**
>
> Pola ini menggunakan pustaka OAuth ekstensi AWS Blu Age, yang disediakan untuk pelanggan sebagai bagian dari keterlibatan Layanan AWS Profesional mereka. Perpustakaan ini bukan bagian dari AWS Blu Age Runtime.

Prasyarat dan batasan

Prasyarat

- Aplikasi modern yang diproduksi oleh alat refactoring modernisasi mainframe AWS Blu Age. Pola ini digunakan CardDemosebagai contoh aplikasi mainframe open source.
- Perpustakaan OAuth ekstensi AWS Blu Age, yang disediakan oleh tim AWS Blu Age selama keterlibatan Anda dengan Layanan AWS Profesional.
- Aktif Akun AWS untuk menyebarkan dan menguji aplikasi modern.
- Keakraban dengan file konfigurasi AWS Blu Age dan dasar-dasar Microsoft Entra ID.

Batasan

- Pola ini mencakup otentikasi OAuth 2.0 dan alur otorisasi berbasis token dasar. Skenario otorisasi lanjutan dan mekanisme kontrol akses berbutir halus tidak dalam cakupan.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

Pola ini dikembangkan dengan menggunakan:

- AWS Blu Age Runtime versi 4.1.0 (polanya juga berfungsi dengan versi yang lebih baru yang kompatibel ke belakang)

- Pustaka MSAL versi 3.0.23
- Kit Pengembangan Java (JDK) versi 17
- Versi sudut 16.1

Arsitektur

Tumpukan teknologi sumber

Dalam lingkungan mainframe yang khas, otentikasi diimplementasikan melalui profil pengguna. Profil ini mengidentifikasi pengguna ke sistem, menentukan siapa yang dapat masuk, dan menentukan fungsi mana yang dapat dilakukan pengguna pada sumber daya sistem. Profil pengguna dikelola oleh petugas keamanan atau administrator keamanan.

Tumpukan teknologi target

- ID Microsoft Entra
- Backend berbasis Java Spring Boot yang dimodernisasi
- AWS Blu Age Runtime
- Keamanan Musim Semi dengan OAuth 2.0
- Aplikasi satu halaman sudut (SPA)

Arsitektur target

AWS Blu Age runtime mendukung otentikasi OAuth berbasis 2.0 secara default, sehingga pola menggunakan standar itu untuk melindungi backend. APIs

Diagram berikut menggambarkan aliran proses.

> **Note**
>
> Diagram mencakup Amazon Aurora sebagai contoh modernisasi basis data meskipun Aurora tidak termasuk dalam langkah-langkah untuk pola ini.

di mana:

1. Pengguna mencoba mengautentikasi dengan Microsoft Entra ID.
2. Microsoft Entra ID mengembalikan token refresh, akses, dan ID yang digunakan aplikasi dalam panggilan berikutnya.
3. Pencegat MSAL menyertakan token akses di `Authorization` header permintaan HTTPS untuk memanggil AWS Blu Age Runtime.
4. `extension-oauth`Pustaka AWS Blu Age mengekstrak informasi pengguna dari header dengan menggunakan file konfigurasi AWS Blu Age Runtime (`application-main.yml`) dan menempatkan informasi ini dalam `SharedContext` objek sehingga logika bisnis dapat menggunakannya.

   > **Note**
   >
   > `SharedContext`adalah komponen runtime yang disediakan oleh AWS Blu Age yang mengelola konteks aplikasi dan informasi status di seluruh aplikasi modern. Untuk informasi selengkapnya tentang komponen dan pembaruan AWS Blu Age Runtime, lihat catatan rilis AWS Blu Age di dokumentasi. AWS Mainframe Modernization Untuk informasi selengkapnya tentang `application-main.yml` file, lihat Mengatur konfigurasi untuk AWS Blu Age Runtime dalam dokumentasi. AWS Mainframe Modernization

5. AWS Blu Age Runtime memeriksa apakah token ada.
   a. Jika token ada, ia memeriksa validitas token dengan berkomunikasi dengan Microsoft Entra ID.
   b. Jika token tidak ada, AWS Blu Age Runtime mengembalikan kesalahan dengan kode status HTTP 403.
6. Jika token valid, AWS Blue Age Runtime memungkinkan logika bisnis untuk melanjutkan. Jika token tidak valid, AWS Blu Age Runtime mengembalikan kesalahan dengan kode status HTTP 403.

OAuth Alur kerja 2.0

Untuk diagram tingkat tinggi alur kerja OAuth 2.0, lihat dokumentasi Microsoft Entra.

Alat

Layanan AWS

AWS Mainframe Modernizationmenyediakan alat dan sumber daya untuk membantu Anda merencanakan dan mengimplementasikan migrasi dan modernisasi dari mainframe ke lingkungan

runtime AWS terkelola. Anda dapat menggunakan fitur refactoring layanan ini, yang disediakan oleh AWS Blu Age, untuk mengonversi dan memodernisasi aplikasi mainframe lama Anda.

> **Note**
>
> AWS Mainframe Modernization Layanan (Managed Runtime Environment experience) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Repositori kode

CardDemo Aplikasi telah diperbarui untuk menunjukkan integrasi dengan Microsoft Entra ID. Anda dapat mengakses kode dari GitHub repositori untuk pola ini.

Konfigurasi backend

Pola ini memerlukan perubahan pada file `application-main.yml` konfigurasi untuk mengaktifkan Spring Security dengan menggunakan OAuth 2.0 pada aplikasi backend. `.yml`File terlihat seperti ini:

```
gapwalk-application.security: enabled
gapwalk-application:
  security:
    identity: oauth
    issuerUri: ${issuerUrl}
    claim:
      claims:
        -
          claimName: upn
          claimMapValue: username
spring:
  autoconfigure:
    exclude:
     -
 org.springframework.boot.autoconfigure.security.oauth2.client.servlet.OAuth2ClientAutoConfigur
     -
 org.springframework.boot.autoconfigure.security.oauth2.resource.servlet.OAuth2ResourceServerAu
  security:
    oauth2:
```

```
      client:
        registration:
          azure:
            client-id: {clientId}
            client-secret: ${clientSecret}
            provider: azure
            authorization-grant-type: authorization_code
            redirect-uri: ${redirectUri}
            scope: openid

        provider:
          azure:
            authorization-uri: ${gapwalk-application.security.issuerUri}/oauth2/v2.0/
authorize
            token-uri:  ${gapwalk-application.security.issuerUri}/oauth2/v2.0/token
            jwk-set-uri: ${gapwalk-application.security.issuerUri}/discovery/v2.0/keys
      resourceserver:
        jwt:
          jwk-set-uri: ${gapwalk-application.security.issuerUri}/discovery/v2.0/keys
```

AWS Pustaka filter OAuth ekstensi Blu Age

Perpustakaan OAuth ekstensi AWS Blu Age disediakan oleh tim AWS Blu Age selama keterlibatan Anda dengan Layanan AWS Profesional.

Pustaka ini membaca `claim.claims` konfigurasi dalam `application-main.yml` fie yang ditampilkan di blok kode sebelumnya. Konfigurasi ini adalah daftar. Setiap item dalam daftar memberikan dua nilai: `claimName` dan`claimMapValue`. `claimName`merupakan nama kunci dalam JSON Web Token (JWT) yang dikirim oleh frontend, dan `claimMapValue` merupakan nama kunci dalam. `SharedContext` Misalnya, jika Anda ingin menangkap ID pengguna di backend, atur `claimName` ke nama kunci di JWT yang menyimpan `userId` yang disediakan oleh Microsoft Entra ID, dan atur `claimMapValue` ke nama kunci untuk mengambil ID pengguna dalam kode backend.

Misalnya, jika Anda mengatur `UserIdclaimMapValue`, Anda dapat menggunakan kode berikut untuk mengekstrak ID pengguna:

```
SharedContext.get().getValue("userId", [UserId]);
```

Praktik terbaik

Dalam penerapan pola ini, pertimbangkan pertimbangan keamanan penting berikut.

> **Important**
>
> Pola ini memberikan dasar untuk integrasi otentikasi. Kami menyarankan Anda menerapkan langkah-langkah keamanan selain yang dibahas di bagian ini berdasarkan persyaratan bisnis Anda sebelum Anda menyebarkannya ke produksi.

- AWS keamanan konfigurasi.Pindahkan nilai konfigurasi sensitif dari `application-main.yml` ke AWS Secrets Manager. Misalnya, konfigurasikan properti berikut dengan menggunakan Secrets Manager:

```
security:
    oauth2:
      client:
        registration:
          azure:
            client-id: {clientId}
            client-secret: ${clientSecret}
```

  Untuk informasi selengkapnya tentang cara menggunakan Secrets Manager untuk mengonfigurasi parameter AWS Blu Age, lihat Rahasia AWS Blu Age Runtime di dokumentasi. AWS Mainframe Modernization
- Perlindungan lingkungan runtime. Konfigurasikan lingkungan aplikasi modern dengan kontrol AWS keamanan yang tepat:

```
server:
  tomcat:
    remoteip:
     protocol-header: X-Forwarded-Proto
     remote-ip-header: X-Forwarded-For
  forward-headers-strategy: NATIVE
```

- CloudWatch Pencatatan Amazon. Pertimbangkan untuk menambahkan file`logback-spring.xml` to `src/main/resources`:

```
<configuration>
 <appender name="CLOUDWATCH"
 class="com.amazonaws.services.logs.logback.CloudWatchAppender">
   <logGroup>/aws/bluage/application</logGroup>
   <logStream>${AWS_REGION}-${ENVIRONMENT}</logStream>
```

```
    <layout>
     <pattern>%d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} - %msg%n</
pattern>
    </layout>
  </appender>

  <root level="INFO">
  <appender-ref ref="CLOUDWATCH"/>
  </root>
</configuration>
```

Untuk informasi tentang mengaktifkan penelusuran dengan CloudWatch, lihat Mengaktifkan jejak untuk mencatat korelasi dalam dokumentasi. CloudWatch

- Konfigurasi dan penanganan token. Konfigurasikan masa pakai token di Microsoft Entra ID agar selaras dengan persyaratan keamanan Anda. Tetapkan token akses untuk kedaluwarsa dalam 1 jam dan refresh token untuk kedaluwarsa dalam 24 jam. Dalam konfigurasi AWS Blu Age Runtime (`application-main.yml`), pastikan validasi JWT dikonfigurasi dengan benar dengan URI penerbit dan nilai audiens yang tepat dari pendaftaran aplikasi Entra ID Anda.

  Saat token kedaluwarsa dan di-refresh:

  1. Pencegat kesalahan aplikasi Angular menangani respons 401 dengan mendapatkan token baru melalui MSAL.
  2. Token baru dikirim dengan permintaan berikutnya.
  3. OAuth Filter AWS Blu Age Runtime memvalidasi token baru dan secara otomatis memperbarui `SharedContext` dengan informasi pengguna saat ini. Ini memastikan bahwa logika bisnis terus memiliki akses ke konteks pengguna yang valid melalui `SharedContext.get().getValue()` panggilan.

  Untuk informasi selengkapnya tentang komponen AWS Blu Age Runtime dan pembaruannya, lihat catatan rilis AWS Blu Age.

- Keamanan Runtime AWS Blu Age. `oauth2-ext`Pustaka yang disediakan oleh AWS Blu Age harus ditempatkan di lokasi direktori bersama yang benar (`{app-server-home}/shared/`) dengan izin file yang tepat. Verifikasi bahwa pustaka berhasil mengekstrak informasi pengguna JWTs dengan memeriksa populasi `SharedContext` objek di log Anda.
- Konfigurasi klaim khusus. Dalam`application-main.yml`, tentukan klaim yang Anda butuhkan dari Microsoft Entra ID secara eksplisit. Misalnya, untuk menangkap email dan peran pengguna, tentukan:

```
gapwalk-application:
  security:
    claim:
      claims:
        - claimName: upn
          claimMapValue: username
        - claimName: roles
          claimMapValue: userRoles
        - claimName: email
          claimMapValue: userEmail
```

- Penanganan kesalahan. Tambahkan penanganan kesalahan untuk mengatasi kegagalan otentikasi di aplikasi Angular Anda; misalnya:

```
@Injectable()
export class AuthErrorInterceptor implements HttpInterceptor {
  intercept(request: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>>
 {
    return next.handle(request).pipe(
      catchError((error: HttpErrorResponse) => {
        if (error.status === 401) {
          // Handle token expiration
          this.authService.login();
        }
        if (error.status === 403) {
          // Handle unauthorized access
          this.router.navigate(['/unauthorized']);
        }
        return throwError(() => error);
      })
    );
  }
}
```

- Konfigurasi batas waktu sesi. Konfigurasikan pengaturan batas waktu sesi di AWS Blu Age Runtime dan Microsoft Entra ID. Misalnya, tambahkan kode berikut ke `application-main.yml` file Anda:

```
server:
  servlet:
    session:
```

```
      timeout: 3600 # 1 hour in seconds
```

- MsalGuard. Anda harus menerapkan MsalGuard fitur untuk semua rute yang dilindungi untuk mencegah akses yang tidak sah. Contoh:

```
const routes: Routes = [
    { path: '', redirectTo: '/transaction-runner', pathMatch: 'full' },
    { path: 'transaction-runner', component: TransactionRunnerComponent,
 canActivate:guards },
    { path: 'user-info', component: UserInfoComponent, canActivate:guards },
    { path: 'term/:transid/:commarea', component: TermComponent,
 canActivate:guards },
 { path: 'code', component: TransactionRunnerComponent  }
];
```

  Rute yang tidak memiliki MsalGuard perlindungan akan dapat diakses tanpa otentikasi, berpotensi mengekspos fungsionalitas sensitif. Pastikan bahwa semua rute yang memerlukan otentikasi menyertakan penjaga dalam konfigurasinya.

Epik

Mengatur ID Microsoft Entra

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan akun Microsoft Azure untuk membuat ID Entra.	Untuk opsi dan instruksi, lihat situs web Microsoft Azure.	Pengembang aplikasi
Siapkan ID Microsoft Entra di aplikasi Anda.	Untuk mempelajari cara menambahkan otentikasi Microsoft Entra ID B2C (Azure AD B2C) ke Angular SPA Anda, lihat dokumentasi Microsoft. Secara khusus: 1. Daftarkan aplikasi Anda dan catat pengidentifikasi. 2. Paparkan titik akhir transaksi AWS Blu Age	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan menambahkan dan mengekspos cakupan khusus. 3. Setelah Anda menentukan cakupan dan URL API, konfigurasikan nilai-nilai ini di `apiUri` dan `apiScope` properti `environment.ts` file dalam repositori kode.	

Kloning repositori dan terapkan kode Blu Age Anda AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning GitHub repositori untuk mendapatkan kode Angular yang diperlukan untuk otentikasi.	Jalankan perintah berikut untuk mengkloning GitHub repositori yang disediakan dengan pola ini ke direktori kerja lokal Anda saat ini: `git clone https://github.com/aws-samples/sample-microsoft-entra-id-based-auth-in-aws-bluage-modernized-mainframe-app.git`	Pengembang aplikasi
Terapkan kode modern AWS Blu Age di server Tomcat untuk mengimplementasikan otentikasi.	Untuk mengatur lingkungan lokal yang mencakup Tomcat dan server pengembangan Angular, ikuti langkah-langkah instalasi yang disediakan oleh tim AWS Blu Age sebagai bagian dari keterliba	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tan pelanggan Anda dengan Layanan AWS Profesional.	

Bangun solusi otentikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan keamanan AWS Blu Age Runtime untuk melindungi titik akhir AWS Blu Age REST API.	Konfigurasikan `application-main.yml` file yang digunakan AWS Blu Age Runtime sebagai berikut. Untuk contoh file ini, lihat bagian Repositori kode sebelumnya dalam pola ini. • `spring:security:oauth2:client:*` mengaktifkan dukungan klien Spring Security OAuth 2.0. Ganti pengaturan dengan nilai yang khusus untuk pengaturan Microsoft Entra ID Anda. • `spring:security:oauth2:resourceserver` memungkinkan dukungan server sumber daya Spring Security OAuth 2.0. • `gapwalk-application.security.issuerUri` menentukan URL penyedia identitas untuk mengambil informasi	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	konfigurasi seperti titik akhir otorisasi, titik akhir token, dan sebagainya. • `gapwalk-application.security.identity` harus diatur `keoauth`. • `gapwalk-application.security.claim.claims` menentukan daftar klaim yang diperlukan untuk menangkap nama pengguna IDs dan pengguna. Sebagai contoh, lihat bagian Repositori kode sebelumnya dalam pola ini. • `gapwalk-application.security` harus diatur `keenabled`. • `spring.autoconfigure.exclude` harus dihapus untuk mengaktifkan otentikasi.	
Masukkan kode contoh dari lingkungan lokal Anda ke dalam basis kode Angular modern Blu Age Anda.	Untuk informasi tentang cara memasukkan contoh ke dalam basis kode Angular modern AWS Blu Age Anda, lihat bagian repositori Kode sebelumnya dalam pola ini.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tempatkan `oauth2-ext` perpustakaan di direktori bersama.	Tempatkan **`oauth2-ext`** perpustakaan di direktori bersama server aplikasi sehingga aplikasi modern AWS Blu Age Anda dapat menggunakannya.Jalankan perintah berikut:
	`cd oauth2-ext/target`
	`cp extension-oauth-filter-<version>.jar /{app-server-home}/shared/`
		Pengembang aplikasi

Menyebarkan solusi otentikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan aplikasi frontend.	Jalankan perintah berikut untuk memulai aplikasi frontend secara lokal:
	`npm install`
	`ng serve --ssl`
	`npm start`
	Note
	Menambahkan `--ssl` flag ke `ng serve` perintah memastikan bahwa server pengembangan menggunakan HTTPS, yang lebih aman
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	daripada protokol lain dan memberikan simulasi lingkungan produksi yang lebih baik.	
Mulai aplikasi backend.	Mulai server Tomcat di Eclipse.	Pengembang aplikasi

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji fungsionalitas login.	Akses aplikasi yang digunakan secara lokal di `http://localhost:4200` untuk memverifikasi bahwa pengguna diminta untuk mengonfirmasi identitas mereka. **Note** HTTP digunakan di sini untuk tujuan demonstrasi. Dalam produksi atau lingkungan lain yang dapat diakses publik, Anda harus menggunakan HTTPS untuk keamanan. Bahkan untuk pengemban	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	gan lokal, kami menyarankan Anda mengatur HTTPS jika memungkinkan. Prompt login Microsoft akan muncul, dan pengguna yang dikonfigurasi di Microsoft Entra ID harus diizinkan untuk mengakses aplikasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji header otorisasi dalam permintaan.	**Note** Langkah-langkah berikut menggunakan CardDemo aplikasi sebagai contoh. Langkah-langkah pengujian untuk aplikasi modern lainnya akan bervariasi. 1. Luncurkan transaksi `CC00` di CardDemo aplikasi. 2. Masuk dengan kredensi pengguna. Misalnya, jika Anda menggunakan lokakarya mandiri AWS Blu Age level 3, Anda dapat menggunakan nama `USER0001` dan kata sandi `PASSWORD` untuk kredensialnya. 3. Buka jendela Alat Pengembang di tab yang sama di browser Anda. 4. Buka tab Jaringan dan periksa permintaan yang dikirim ke backend `http://localhost:8080/gapwalk-application/transaction` .	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Periksa header permintaan dan verifikasi bahwa Anda melihat header otorisasi dengan nilai di `Bearer <token>` mana nilai yang `<token>` dihasilkan oleh Microsoft Entra ID.	
Uji fungsionalitas logout.	Pilih Keluar untuk keluar, dan coba akses aplikasi lagi. Ini harus menyajikan prompt login baru.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
Token yang dikeluarkan oleh Microsoft Entra ID tidak kompatibel dengan keamanan Spring Boot OAuth 2.0.	Untuk penyelesaian masalah ini, lihat Microsoft Entra ID OAuth Flow di OAuth blog.
Pertanyaan umum terkait token.	Untuk memecahkan kode dan melihat konten token JWT, gunakan situs web https://jwt.io/.

Sumber daya terkait

- Untuk informasi tentang refactoring aplikasi Anda dengan menggunakan AWS Blu Age, lihat dokumentasi.AWS Mainframe Modernization
- Untuk memahami cara kerja OAuth 2.0, lihat situs web OAuth 2.0.
- Untuk gambaran umum tentang Microsoft Authentication Library (MSAL), lihat dokumentasi Microsoft Entra.
- Untuk informasi tentang profil pengguna pada sistem AS/400, lihat IBM i (AS400) tutorial.

- Untuk alur otentikasi OAuth 2.0 dan OpenID Connect (OIDC) di platform identitas Microsoft, lihat dokumentasi Microsoft Entra.

Integrasikan Stonebranch Universal Controller dengan AWS Mainframe Modernisasi

Vaidy Sankaran dan Pablo Alonso Prieto, Amazon Web Services

Robert Lemieux dan Huseyin Gomleksizoglu, Cabang Batu

Ringkasan

Catatan: AWS Mainframe Modernization Layanan (pengalaman Lingkungan Runtime Terkelola) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Pola ini menjelaskan cara mengintegrasikan orkestrasi beban kerja Stonebranch Universal Automation Center (UAC) dengan layanan Modernisasi Mainframe Amazon Web Services (AWS). Layanan AWS Mainframe Modernisasi memigrasikan dan memodernisasi aplikasi mainframe ke AWS Cloud. Ini menawarkan dua pola: AWS Mainframe Modernization Replatform dengan teknologi Micro Focus Enterprise dan AWS Mainframe Modernization Automated Refactor dengan AWS Blu Age.

Stonebranch UAC adalah platform otomatisasi dan orkestrasi TI real-time. UAC dirancang untuk mengotomatiskan dan mengatur pekerjaan, aktivitas, dan alur kerja di seluruh sistem TI hybrid, dari lokal hingga AWS. Klien perusahaan yang menggunakan sistem mainframe beralih ke infrastruktur dan aplikasi modern yang berpusat pada cloud. Alat dan layanan profesional Stonebranch memfasilitasi migrasi penjadwal yang ada dan kemampuan otomatisasi ke AWS Cloud.

Saat memigrasikan atau memodernisasi program mainframe Anda ke AWS Cloud menggunakan layanan AWS Mainframe Modernization, Anda dapat menggunakan integrasi ini untuk mengotomatiskan penjadwalan batch, meningkatkan kelincahan, meningkatkan pemeliharaan, dan mengurangi biaya.

Pola ini memberikan instruksi untuk mengintegrasikan penjadwal Stonebranch dengan aplikasi mainframe yang dimigrasikan ke runtime layanan Modernisasi AWS Mainframe Micro Focus Enterprise. Pola ini untuk arsitek solusi, pengembang, konsultan, spesialis migrasi, dan lainnya yang bekerja dalam migrasi, modernisasi, operasi, atau. DevOps

Hasil yang ditargetkan

Pola ini berfokus pada penyediaan hasil target berikut:

- Kemampuan untuk menjadwalkan, mengotomatiskan, dan menjalankan pekerjaan batch mainframe yang berjalan di layanan AWS Mainframe Modernization (Microfocus runtime) dari Stonebranch Universal Controller.
- Pantau proses batch aplikasi dari Stonebranch Universal Controller.
- Start/Restart/Rerun/Stopproses batch secara otomatis atau manual dari Stonebranch Universal Controller.
- Ambil hasil proses batch AWS Mainframe Modernization.
- Tangkap CloudWatch log AWS dari pekerjaan batch di Stonebranch Universal Controller.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Aplikasi Micro Focus Bankdemo dengan file job control language (JCL), dan proses batch yang diterapkan di lingkungan AWS Mainframe Modernization service (Micro Focus runtime)
- Pengetahuan dasar tentang cara membangun dan menyebarkan aplikasi mainframe yang berjalan di Micro Focus Enterprise Server
- Pengetahuan dasar tentang Stonebranch Universal Controller
- Lisensi uji coba Stonebranch (hubungi Stonebranch)
- Instans Windows atau Linux Amazon Elastic Compute Cloud (Amazon EC2) (misalnya, xlarge) dengan minimal empat core, memori 8 GB, dan ruang disk 2 GB
- Apache Tomcat versi 8.5.x atau 9.0.x
- Oracle Java Runtime Environment (JRE) atau OpenJDK versi 8 atau 11
- Amazon Aurora MySQL — Edisi yang Kompatibel
- Bucket Amazon Simple Storage Service (Amazon S3) untuk repositori ekspor
- Amazon Elastic File System (Amaon EFS) untuk koneksi agen Stonebranch Universal Message Service (OMS) untuk ketersediaan tinggi (HA)
- Stonebranch Universal Controller 7.2 Agen Universal 7.2 File Instalasi
- Templat penjadwalan tugas Modernisasi AWS Mainframe (versi rilis terbaru dari file.zip)

Batasan

- Produk dan solusi telah diuji dan kompatibilitas divalidasi hanya dengan OpenJDK 8 dan 11.
- Template penjadwalan tugas aws-mainframe-modernization-stonebranch-integrasi hanya akan berfungsi dengan layanan AWS Mainframe Modernization.
- Template penjadwalan tugas ini hanya akan bekerja pada agen Stonebranch edisi Unix, Linux, atau Windows.
- Beberapa layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat layanan AWS menurut Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Arsitektur negara sasaran

Diagram berikut menunjukkan contoh lingkungan AWS yang diperlukan untuk pilot ini.

1. Stonebranch Universal Automation Center (UAC) mencakup dua komponen utama: Universal Controller dan Universal Agents. Stonebranch OMS digunakan sebagai bus pesan antara pengontrol dan agen individu.
2. Stonebranch UAC Database digunakan oleh Universal Controller. Database dapat MySQL, Microsoft SQL Server, Oracle, atau Aurora MySQL — kompatibel.
3. Layanan Modernisasi AWS Mainframe — Lingkungan runtime Fokus Mikro dengan aplikasi yang diterapkan. BankDemo File BankDemo aplikasi akan disimpan dalam ember S3. Bucket ini juga berisi file JCL mainframe.
4. Stonebranch UAC dapat menjalankan fungsi-fungsi berikut untuk batch run:
   a. Mulai pekerjaan batch menggunakan nama file JCL yang ada di bucket S3 yang ditautkan ke layanan modernisasi mainframe AWS.
   b. Dapatkan status pekerjaan batch yang dijalankan.
   c. Tunggu sampai batch job run selesai.
   d. Ambil log dari pekerjaan batch yang dijalankan.
   e. Jalankan kembali pekerjaan batch yang gagal.
   f. Batalkan pekerjaan batch saat pekerjaan sedang berjalan.
5. Stonebranch UAC dapat menjalankan fungsi-fungsi berikut untuk aplikasi:

a. Mulai Aplikasi
b. Dapatkan Status Aplikasi
c. Tunggu hingga Aplikasi dimulai atau dihentikan
d. Hentikan Aplikasi
e. Ambil Log operasi Aplikasi

Konversi pekerjaan Stonebranch

Diagram berikut mewakili proses konversi pekerjaan Stonebranch selama perjalanan modernisasi. Ini menjelaskan bagaimana jadwal pekerjaan dan definisi tugas diubah menjadi format yang kompatibel yang dapat menjalankan tugas batch AWS Mainframe Modernization.

1. Untuk proses konversi, definisi pekerjaan diekspor dari sistem mainframe yang ada.
2. File JCL dapat diunggah ke bucket S3 untuk aplikasi Modernisasi Mainframe sehingga file JCL ini dapat digunakan oleh layanan AWS Mainframe Modernization.
3. Alat konversi mengonversi definisi pekerjaan yang diekspor ke tugas UAC.
4. Setelah semua definisi tugas dan jadwal pekerjaan dibuat, objek ini akan diimpor ke Universal Controller. Tugas yang dikonversi kemudian menjalankan proses di layanan AWS Mainframe Modernization alih-alih menjalankannya di mainframe.

Arsitektur UAC Stonebranch

Diagram arsitektur berikut merupakan active-active-passive model ketersediaan tinggi (HA) Universal Controller. Stonebranch UAC digunakan di beberapa Availability Zone untuk menyediakan ketersediaan tinggi dan mendukung pemulihan bencana (DR).

Pengontrol Universal

Dua server Linux disediakan sebagai Universal Controller. Keduanya terhubung ke endpoint database yang sama. Setiap server memiliki aplikasi Universal Controller dan OMS. Versi terbaru dari Universal Controller digunakan pada saat itu disediakan.

Pengontrol Universal digunakan di webapp Tomcat sebagai ROOT dokumen dan disajikan pada port 80. Penerapan ini memudahkan konfigurasi penyeimbang beban frontend.

HTTP melalui TLS atau HTTPS diaktifkan menggunakan sertifikat wildcard Stonebranch (misalnya,). `https://customer.stonebranch.cloud` Ini mengamankan komunikasi antara browser dan aplikasi.

OMS

Agen Universal dan OMS (Layanan Pesan Opswise) berada di setiap server Universal Controller. Semua Agen Universal yang dikerahkan dari ujung pelanggan diatur untuk terhubung ke kedua layanan OMS. OMS bertindak sebagai layanan pesan umum antara Universal Agents dan Universal Controller.

Amazon EFS memasang direktori spool di setiap server. OMS menggunakan direktori spool bersama ini untuk menjaga koneksi dan informasi tugas dari pengontrol dan agen. OMS bekerja dalam mode ketersediaan tinggi. Jika OMS aktif turun, OMS pasif memiliki akses ke semua data, dan melanjutkan operasi aktif secara otomatis. Agen Universal mendeteksi perubahan ini dan secara otomatis terhubung ke OMS aktif baru.

Basis Data

Amazon Relational Database Service (Amazon RDS) menampung database UAC, dengan Amazon Aurora MySQL yang kompatibel dengan mesinnya. Amazon RDS membantu dalam mengelola dan menawarkan cadangan terjadwal secara berkala. Kedua instans Universal Controller terhubung ke endpoint database yang sama.

Penyeimbang beban

Application Load Balancer diatur untuk setiap instance. Penyeimbang beban mengarahkan lalu lintas ke pengontrol aktif pada saat tertentu. Nama domain instance Anda mengarah ke titik akhir penyeimbang beban masing-masing.

URLs

Setiap instance Anda memiliki URL, seperti yang ditunjukkan pada contoh berikut.

Lingkungan	Instans
Produksi	`customer.stonebranch.cloud`
Pengembangan (non-produksi)	`customerdev.stonebranch.cloud`

Pengujian (non-produksi)	`customertest.stonebranch.cloud`

> **Note**
>
> Nama instans non-produksi dapat diatur berdasarkan kebutuhan Anda.

Ketersediaan tinggi

Ketersediaan tinggi (HA) adalah kemampuan sistem untuk beroperasi terus menerus tanpa kegagalan untuk jangka waktu yang ditentukan. Kegagalan tersebut termasuk, namun tidak terbatas pada, penyimpanan, penundaan respons komunikasi server yang disebabkan oleh masalah CPU atau memori, dan konektivitas jaringan.

Untuk memenuhi persyaratan HA:

- Semua EC2 instans, database, dan konfigurasi lainnya dicerminkan di dua Availability Zone terpisah dalam Wilayah AWS yang sama.
- Pengontrol disediakan melalui Amazon Machine Image (AMI) pada dua server Linux di dua Availability Zones. Misalnya, jika Anda disediakan di Wilayah eu-west-1 Eropa, Anda memiliki Pengontrol Universal di Availability Zone eu-west-1a dan Availability Zone eu-west-1c.
- Tidak ada pekerjaan yang diizinkan untuk berjalan langsung di server aplikasi dan tidak ada data yang diizinkan untuk disimpan di server ini.
- Application Load Balancer menjalankan pemeriksaan kesehatan pada setiap Universal Controller untuk mengidentifikasi yang aktif dan mengarahkan lalu lintas ke sana. Jika satu server mengalami masalah, penyeimbang beban secara otomatis mempromosikan Universal Controller pasif ke status aktif. Load balancer kemudian mengidentifikasi instans Universal Controller aktif baru dari pemeriksaan kesehatan dan mulai mengarahkan lalu lintas. Failover terjadi dalam waktu empat menit tanpa kehilangan pekerjaan, dan URL frontend tetap sama.
- Layanan database yang kompatibel dengan Aurora MySQL menyimpan data Universal Controller. Untuk lingkungan produksi, kluster database dibangun dengan dua instance database di dua Availability Zone yang berbeda dalam satu Wilayah AWS. Kedua Pengontrol Universal menggunakan antarmuka Java Database Connectivity (JDBC) yang menunjuk ke titik akhir cluster database tunggal. Jika satu instance database menimbulkan masalah, titik akhir cluster database secara dinamis menunjuk ke instance sehat. Tidak diperlukan intervensi manual.

Backup dan bersihkan

Stonebranch Universal Controller diatur untuk mencadangkan dan membersihkan data lama mengikuti jadwal yang ditunjukkan pada tabel.

Jenis	Jadwal
Aktivitas	7 hari
Audit	90 hari
Sejarah	60 hari

Backup data yang lebih lama dari tanggal yang ditampilkan diekspor ke format.xml. dan disimpan dalam sistem file. Setelah proses pencadangan selesai, data lama akan dihapus dari database dan diarsipkan dalam bucket S3 hingga satu tahun untuk instance produksi.

Anda dapat menyesuaikan jadwal ini di antarmuka Universal Controller Anda. Namun, peningkatan kerangka waktu ini dapat menyebabkan waktu henti yang lebih lama selama pemeliharaan.

Alat

Layanan AWS

- AWS Mainframe Modernization adalah platform cloud-native AWS yang membantu Anda memodernisasi aplikasi mainframe ke lingkungan runtime yang dikelola AWS. Ini menyediakan alat dan sumber daya untuk membantu Anda merencanakan dan menerapkan migrasi dan modernisasi.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon. EC2
- Amazon Elastic File System (Amazon EFS) membantu Anda membuat dan mengonfigurasi sistem file bersama di AWS Cloud.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud. Pola ini menggunakan Amazon Aurora MySQL-Compatible Edition.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh EC2 instans Amazon, kontainer, dan alamat IP di satu atau beberapa Availability Zone. Pola ini menggunakan Application Load Balancer.

Stonebranch

- Universal Automation Center (UAC) adalah sistem produk otomatisasi beban kerja perusahaan. Pola ini menggunakan komponen UAC berikut:
  - Universal Controller, aplikasi web Java yang berjalan di wadah web Tomcat, adalah penjadwal pekerjaan perusahaan dan solusi broker otomatisasi beban kerja dari Universal Automation Center. Controller menyajikan antarmuka pengguna untuk membuat, memantau, dan mengkonfigurasi informasi Controller; menangani logika penjadwalan; memproses semua pesan ke dan dari Agen Universal; dan menyinkronkan banyak operasi ketersediaan tinggi Universal Automation Center.
  - Universal Agent adalah agen penjadwalan independen vendor yang berkolaborasi dengan penjadwal pekerjaan yang ada di semua platform komputasi utama, baik warisan maupun terdistribusi. Semua penjadwal yang berjalan diz/Series, i/Series, Unix, Linux, atau Windows didukung.
- Universal Agent adalah agen penjadwalan independen vendor yang berkolaborasi dengan penjadwal pekerjaan yang ada di semua platform komputasi utama, baik warisan maupun terdistribusi. Semua penjadwal yang berjalan diz/Series, i/Series, Unix, Linux, atau Windows didukung.
- Stonebranch aws-mainframe-modernization-stonebranch -integration AWS Mainframe Modernization Universal Extension adalah template integrasi untuk menjalankan, memantau, dan menjalankan kembali pekerjaan batch di platform AWS Mainframe Modernization.

Kode

Kode untuk pola ini tersedia di GitHub repositori aws-mainframe-modernization-stonebranch-integration.

Epik

Instal Pengontrol Universal dan Agen Universal di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh file instalasi.	Unduh instalasi dari server Stonebranch. Untuk mendapatkan file instalasi, kontak dengan Stonebranch.	Arsitek awan
Luncurkan EC2 instance.	Anda akan membutuhkan sekitar 3 GB ruang ekstra untuk instalasi Universal Controller dan Universal Agent. Jadi sediakan setidaknya 30 GB ruang disk untuk instance. Tambahkan port 8080 ke grup keamanan sehingga dapat diakses.	Arsitek awan

Periksa prasyarat.	Sebelum instalasi, lakukan hal berikut:
	
	1. Instal Java seperti yang dijelaskan dalam Mengunduh Java Runtime Environment.
	
	``
	$ sudo yum -y update
	$ sudo yum install
	 java-11-amazon-cor
	retto
	``
	
	Pastikan untuk menggunakan salah satu versi JAVA yang didukung.
		Administrator cloud, administrator Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Perintah sebelumnya harus menginstal java-11. Periksa versi Java dan pastikan Anda menggunakan versi 11 sebelum melanjutkan.
	2. Seperti yang dijelaskan dalam Menginstal Apache Tomcat, jalankan perintah berikut.
	`$ sudo yum install tomcat tomcat-admin-webapps`
	`$ sudo systemctl enable tomcat`
	`$ sudo systemctl start tomcat`
	3. Buat database Amazon Aurora seperti yang dijelaskan dalam Membuat dan menghubungkan ke cluster DB MySQL Aurora. Gunakan Amazon Aurora Edisi yang kompatibel dengan MySQL.
	Pilih nama pengguna Master dan kata sandi Master. Simpan nilai default untuk sisa pengaturan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal Pengontrol Universal.	1. Unggah file `universal-controller-7.2.0.0.tar` instalasi ke EC2 instance.
	2. Buka arsip file instalasi ke `temp` folder.
	`$ tar -xvf universal-controller-7.2.0.0.tar`
	3. Berikan izin menjalankan skrip instalasi.
	`$ chmod a+x install-controller.sh`
	4. Pasang pengendali. Contoh ini menggunakan perintah berikut untuk menginstal Universal Controller di bawah `/usr/share/tomcat`. Gunakan database Amazon Aurora yang Anda buat di langkah sebelumnya.
	`$ sudo ./install-controller.sh --tomcat-dir /usr/share/tomcat/ --controller-file universal-controller-7.2.0.0-build.145.war --dbuser admin --dbpass "****" --dbname uc --rdbms mysql --dburl`
		Arsitek cloud, administrator Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	jdbc:mysql://database-2-instance-1.cih63miincgy.us-east-1.rds.amazonaws.com:3306/ Baris terakhir dari output skrip harus “Instalasi selesai.” 5. Arahkan ke URL berikut dalam EC2 contoh. http://<public_ip>:8080/uc 6. Pada layar login, masukkan ops.admin di bagian Nama Pengguna, dan biarkan bidang Kata Sandi kosong. 7. Tetapkan kata sandi baru untuk `ops.admin` pengguna.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal Agen Universal.	1. Unggah file `sb-7.2.0.1-linux-3.10-x86_64.tar.Z` instalasi ke EC2 instance.
	2. Masuk ke EC2 instance.
	3. Buka arsip paket instalasi Agen Universal.
	`$ zcat sb-7.2.0.1-linux-3.10-x86_64.tar.Z | tar xvf -`
	4. Jalankan perintah berikut.
	`$ sudo ./unvinst --oms_servers 7878@localhost --oms_autostart yes --python yes`
	5. Buat file PAM.
	`$ cp /etc/pam.d/login /etc/pam.d/ucmd`
	6. Aktifkan Autostart untuk Agen Universal.
	`$ /sbin/restorecon -v /etc/rc.d/init.d/ubrokerd`
		Administrator cloud, administrator Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan OMS ke Universal Controller.	1. Masuk ke Universal Controller dengan `ops.admin` pengguna. 2. Pilih menu Layanan di sudut kiri atas layar, lalu pilih menu Server OMS di Sistem 3. Di bidang Alamat Server OMS, ketik`localhost` , lalu simpan. 4. Anda akan melihat status server OMS sebagai Terhubung dan Status Sesi sebagai Operasional.	Administrator Pengontrol Universal

Impor AWS Mainframe Modernization Universal Extension dan buat tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Templat Integrasi Impor.	Untuk langkah ini, Anda memerlukan AWS Mainframe Modernization Universal Extension. Pastikan versi rilis terbaru dari file.zip diunduh. 1. Masuk ke Universal Controller dengan `ops.admin` pengguna. 2. Arahkan ke Layanan, Template Integrasi Impor. 3. Pilih file Integration Template .zip (`aws_mainframe_modernization`	Administrator Pengontrol Universal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`_stonebranch_extension.zip`), dan pilih Impor. Setelah Template Integrasi diimpor, Anda akan melihat Tugas Modernisasi AWS Mainframe di bawah Layanan yang Tersedia.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan kredenal yang dapat diselesaikan.	1. Arahkan ke Layanan, Tugas Modernisasi AWS Mainframe. 2. Di panel kanan, isi bidang yang diperlukan: • Nama: Tugas Modernisasi Mainframe Baru • Agen: Pilih satu-satunya agen (AGNT0001). Di bawah Detail Modernisasi AWS Mainframe: • Tindakan: Daftar Lingkungan • AWS Credentials: Jika Anda memiliki peran AWS Identity and Access Management (IAM) yang ditambahkan ke EC2 instance, Anda dapat membiarkan bidang ini kosong. Jika Anda akan menggunakan `AWSAccessKeyID` dan`AWSSecretKey` , pilih ikon () di sebelah bidang. Di jendela Credential Details yang terbuka, masukkan informasi berikut dan kemudian simpan.	Administrator Pengontrol Universal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Nama: Kredensia l Modernisasi AWS Mainframe • Pengguna Runtime: Tulis ID kunci akses AWS di bidang ini. • Kata Sandi Runtime: Tulis kunci rahasia AWS di bidang ini. • Titik Akhir: Pastikan titik akhir memiliki Wilayah AWS yang benar. Nilai default-nya `https://m2.us-east-1.amazonaws.com` . • Wilayah: Masuk ke Wilayah layanan Modernisasi AWS Mainframe. Nilai default-nya `us-east-1` . 3. Simpan nilai default di bidang lainnya, dan simpan tugas.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan tugas.	1. Di bagian atas panel kanan, pilih Launch Task. 2. Di jendela Konfirmasi, pilih Luncurkan. Setelah itu, Universal Controller Console akan menampilkan pesan yang mirip dengan pesan berikut. 2022-08-24 10:11:49 Berhasil meluncurkan tugas Universal “Tugas Modernisasi Mainframe Baru” dengan instance tugas sys_id 1661291493634146313 E38 OZJY. NC8 DB8 3. Arahkan ke Instans Jika Anda tidak melihat tab Instans, pilih panah kanan untuk menggulir ke kanan. 4. Buka menu konteks (klik kanan) untuk instance tugas dalam daftar, pilih Ambil Output, lalu pilih Kirim di Ambil Output 5. Di jendela Retrieve Output, Anda akan melihat daftar lingkungan di STDOUT.	Administrator Pengontrol Universal

Uji memulai pekerjaan batch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tugas untuk pekerjaan batch.	1. Arahkan ke Layanan, Tugas Modernisasi AWS Mainframe. 2. Di panel kanan, isi bidang yang diperlukan: • Nama: Tugas Modernisasi Mainframe Baru • Agen: Pilih satu-satunya agen (AGNT0001). Di bawah Detail Modernisasi AWS Mainframe: • Tindakan: Mulai Batch (atau Mulai Batch dan Tunggu untuk menjalankan pekerjaan batch dan menunggu hingga tugas selesai di AWS) • AWS Credentials: Jika Anda memiliki peran IAM yang ditambahkan ke EC2 instance, Anda dapat membiarkan bidang ini kosong. Jika Anda akan menggunakan `AWSAccessKeyID` dan`AWSSecretKey` , pilih ikon () di sebelah bidang. • Titik Akhir: Pastikan titik akhir memiliki Wilayah AWS yang benar. Nilai	Administrator Pengontrol Universal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	default-nya https://m2.us-east-1.amazonaws.com. • Wilayah: Masuk ke Wilayah layanan Modernisasi AWS Mainframe. Nilai default-nya `us-east-1` . • Aplikasi: Pilih ikon di sebelah bidang (), dan pilih Kirim di Refresh Application Choices. Ini akan terhubung ke layanan AWS Mainframe Modernization dan mengembalikan daftar aplikasi. Sekarang Anda dapat memilih aplikasi dari daftar dropdown. Pilih aplikasi yang ingin Anda jalankan pekerjaan batch. • Nama File JCL: `RUNHELLO.jcl` • Tunggu Sukses atau Kegagalan: Jika opsi ini dipilih, tugas akan menunggu hingga status pekerjaan batch berhasil atau gagal. • Interval Polling: Ini adalah jumlah waktu antara setiap polling. • Ambil Log Eksekusi: Jika dipilih, log akan diambil	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	secara otomatis ketika pekerjaan batch telah selesai. • Format Log: Ini adalah format log yang akan dicetak. Ini bisa berupa format Teks atau JSON. 3. Simpan nilai default di bidang lainnya, dan simpan tugas.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan tugas.	1. Di bagian atas panel kanan, pilih Launch Task. 2. Di jendela Konfirmasi, pilih Luncurkan. Setelah itu, Universal Controller Console akan menampilkan pesan yang mirip dengan pesan berikut. 2022-08-24 11:11:59 Berhasil meluncurkan tugas Universal “Batch Mulai Modernisasi Mainframe ” dengan instance tugas sys_id. <sys id> 3. Arahkan ke Instans Jika Anda tidak melihat tab Instans, pilih panah kanan untuk menggulir ke kanan. 4. Buka menu konteks (klik kanan) untuk instance tugas dalam daftar, pilih Ambil Output, lalu pilih Kirim di Ambil Output 5. Di jendela Retrieve Output, Anda akan melihat daftar lingkungan di STDOUT.	Administrator Pengontrol Universal

Buat alur kerja untuk beberapa tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin tugas.	1. Buka menu konteks (klik kanan) untuk tugas yang ingin Anda buat salinannya, dan pilih Salin. 2. Di jendela Copy AWS Mainframe Modernization Task, masukkan nama baru berikut untuk tugas baru: Batch Mulai Modernisasi Mainframe -. RUNAWS2 3. Salin tugas lagi, menggunakan nama berikut: Modernisasi Mainframe Mulai Batch -. RUNAWS3 4. Salin dengan tugas lagi, menggunakan nama berikut: Modernisasi Mainframe Mulai Batch -. RUNAWS4 5. Salin tugas terakhir kali, menggunakan nama berikut: Modernisasi Mainframe Mulai Batch - FOOBAR.	Administrator Pengontrol Universal
Perbarui tugas.	1. Buka (klik dua kali) Batch Mulai Modernisa si Mainframe RUNAWS2 - tugas, ubah bidang `RUNAWS2.jcl` Nama File JCL menjadi, dan simpan.	Administrator Pengontrol Universal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buka (klik dua kali) Batch Mulai Modernisasi Mainframe RUNAWS3 - tugas, ubah bidang *RUNAWS3.jcl* Nama File JCL menjadi, dan simpan. 3. Buka (klik dua kali) Batch Mulai Modernisasi Mainframe RUNAWS4 - tugas, ubah bidang *RUNAWS4.jcl* Nama File JCL menjadi, dan simpan. 4. Buka (klik dua kali) tugas Batch Mulai Modernisasi Mainframe - FOOBAR, ubah bidang Nama File JCL menjadi, dan simpan. *MISSING.jcl* Tugas ini akan gagal karena nilai Nama File JCL salah.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat alur kerja.	1. Arahkan ke Layanan, Alur Kerja. 2. Di panel kanan, masukkan Alur Kerja Modernisasi Mainframe di bidang Nama, dan simpan. 3. Di panel kanan, pilih Edit Alur Kerja. 4. Pada Tab Editor Alur Kerja, tombol Tambahkan Tugas (+). 5. Di jendela Task Find, pilih Cari untuk melihat semua tugas di Universal Controlle r. 6. Klik ikon di samping Modernisasi Mainframe Start Batch Task, dan seret ikon ke tempat kosong di Workflow Editor. 7. Ulangi tindakan yang sama untuk tugas Modernisa si Mainframe lainnya dan letakkan seperti yang ditunjukkan di bagian Informasi tambahan. 8. Pilih tombol Connect (), dan hubungkan tugas bersama-sama. Untuk menghubun gkan tugas dengan tugas lain, klik di tengah tugas, dan seret ke tugas target.	Administrator Pengontrol Universal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	9. Hubungkan tugas seperti yang ditunjukkan di bagian Informasi tambahan, dan simpan alur kerja. 10. Klik kanan tempat kosong di Editor Alur Kerja, pilih Luncurkan Alur Kerja, lalu pilih OK.	
Periksa status alur kerja.	1. Di menu sebelah kiri, pilih Aktivitas 2. Di tengah jendela, pilih Mulai. Anda akan melihat daftar contoh tugas dalam daftar. 3. Buka (klik dua kali) Alur Kerja Modernisasi Mainframe dalam daftar, atau buka menu konteks (klik kanan) dan pilih Perintah Tugas Alur Kerja, Lihat Alur Kerja. Anda akan melihat tugas seperti yang ditunjukk an di bagian Informasi tambahan. Tugas kedua diharapkan gagal karena Anda menggunakan file JCL yang hilang.	Administrator Pengontrol Univeral

Memecahkan masalah pekerjaan batch yang gagal dan jalankan kembali

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbaiki tugas yang gagal dan jalankan kembali.	1. Buka (klik dua kali) tugas yang gagal untuk melihat kesalahan tugas. 2. Anda memiliki dua opsi untuk memperbaiki tugas yang gagal. • Perbaiki nama file JCL, dan atur ke`F00BAR.jcl` . • Tambahkan nama file JCL yang benar ke JCL File Name (Temp). Bidang ini akan menimpa bidang Nama File JCL. Untuk pilot ini, pilih opsi kedua, dan simpan instance tugas. 3. Di Workflow Monitor, buka menu konteks (klik kanan) untuk tugas yang gagal, dan pilih Command, Re-run. 4. Setelah itu, semua tugas akan selesai dengan sukses.	Administrator Pengontrol Universal

Buat Mulai Aplikasi dan Hentikan tugas Aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tindakan Mulai Aplikasi.	1. Arahkan ke Layanan, Tugas Modernisasi AWS Mainframe. 2. Di panel kanan, isi bidang yang diperlukan. • Nama: Modernisasi Mainframe Mulai Aplikasi • Agen: Pilih satu-satunya agen (AGNT0001) Di bawah Detail Modernisasi AWS Mainframe: • Tindakan: Mulai Aplikasi • AWS Credentials: Jika Anda memiliki peran IAM yang ditambahkan ke EC2 instance, Anda dapat membiarkan bidang ini kosong. Jika Anda akan menggunakan `AWSAccessKeyID` dan`AWSSecretKey` , pilih kredensi yang Anda buat sebelumnya. • Titik Akhir: Pastikan titik akhir memiliki Wilayah yang benar. Nilai default-nya `https://m2.us-east-1.amazonaws.com` . • Wilayah: Masuk ke Wilayah layanan	Administrator Pengontrol Universal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Modernisasi AWS Mainframe. Nilai default-nya `us-east-1` . • Aplikasi: Pilih ikon di sebelah bidang (), dan pilih Kirim di Refresh Application Choices. Ini akan terhubung ke layanan AWS Mainframe Modernization dan mengembalikan daftar aplikasi. Sekarang Anda dapat memilih aplikasi dari daftar dropdown. Pilih aplikasi yang ingin Anda jalankan pekerjaan batch. • Tunggu Sukses atau Kegagalan: Jika opsi ini dipilih, tugas akan menunggu hingga status pekerjaan batch berhasil atau gagal. • Interval Polling: Ini adalah jumlah waktu antara setiap polling. • Ambil Log Eksekusi: Jika dipilih, log akan diambil secara otomatis ketika pekerjaan batch telah selesai. • Format Log: Ini adalah format log yang akan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dicetak. Ini bisa berupa format Teks atau JSON. 3. Simpan nilai default di bidang lainnya, dan simpan tugas. 4. Sekarang salin tugas ini dan buat tugas untuk Stop Application. Ubah nama menjadi Mainframe Moderinization Stop Application, dan ubah action menjadi Stop Application.	

Membuat tugas Batalkan Eksekusi Batch

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tindakan Batch Batch.	1. Arahkan ke Layanan, Tugas Modernisasi AWS Mainframe. 2. Di panel kanan, isi bidang yang diperlukan. • Nama: Modernisasi Mainframe Batalkan Eksekusi Batch • Agen: Pilih satu-satunya agen (AGNT0001) Di bawah Detail Modernisasi AWS Mainframe: • Tindakan: Batalkan Eksekusi Batch	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• AWS Credentials: Jika Anda memiliki peran IAM yang ditambahkan ke EC2 instance, Anda dapat membiarkan bidang ini kosong. Jika Anda akan menggunakan `AWSAccessKeyID` dan`AWSSecretKey` , pilih kredensi yang Anda buat sebelumnya. • Titik Akhir: Pastikan titik akhir memiliki Wilayah yang benar. Nilai default-nya `https://m2.us-east-1.amazonaws.com` . • Wilayah: Masuk ke Wilayah layanan Modernisasi AWS Mainframe. Nilai default-nya `us-east-1` . • Aplikasi: Pilih ikon di sebelah bidang (), dan pilih Kirim di Refresh Application Choices. Ini akan terhubung ke layanan AWS Mainframe Modernization dan mengembalikan daftar aplikasi. Sekarang Anda dapat memilih aplikasi dari daftar dropdown. Pilih aplikasi yang ingin	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda jalankan pekerjaan batch. • Tunggu Sukses atau Kegagalan: Jika opsi ini dipilih, tugas akan menunggu hingga status pekerjaan batch berhasil atau gagal. • Interval Polling: Ini adalah jumlah waktu antara setiap polling. • Ambil Log Eksekusi: Jika dipilih, log akan diambil secara otomatis ketika pekerjaan batch telah selesai. • Format Log: Ini adalah format log yang akan dicetak. Ini bisa berupa format Teks atau JSON. 3. Simpan nilai default di bidang lainnya, dan simpan tugas.	

Sumber daya terkait

- Pengontrol Universal
- Agen Universal
- Pengaturan LDAP
- SALL Single Sign-On
- Alat Konversi Xpress

Informasi tambahan

Ikon di Editor Alur Kerja

Semua tugas terhubung

Status alur kerja

Memigrasi dan mereplikasi file VSAM ke Amazon RDS atau Amazon MSK menggunakan Connect from Excently

Prachi Khanna dan Boopathy GOPALSAMY, Amazon Web Services

Ringkasan

Pola ini menunjukkan kepada Anda cara memigrasi dan mereplikasi file Metode Akses Penyimpanan Virtual (VSAM) dari mainframe ke lingkungan target di AWS Cloud dengan menggunakan Connect from Accurly. Lingkungan target yang tercakup dalam pola ini termasuk Amazon Relational Database Service (Amazon RDS) dan Amazon Managed Streaming for Apache Kafka (Amazon MSK). Connect menggunakan change data capture (CDC) untuk terus memantau pembaruan ke file VSAM sumber Anda dan kemudian mentransfer pembaruan ini ke satu atau beberapa lingkungan target AWS Anda. Anda dapat menggunakan pola ini untuk memenuhi tujuan modernisasi aplikasi atau analisis data Anda. Misalnya, Anda dapat menggunakan Connect untuk memigrasikan file aplikasi VSAM Anda ke AWS Cloud dengan latensi rendah, atau memigrasikan data VSAM Anda ke gudang data AWS atau data lake untuk analitik yang dapat mentolerir latensi sinkronisasi yang lebih tinggi dari yang diperlukan untuk modernisasi aplikasi.

Prasyarat dan batasan

Prasyarat

- IBM z/OS V2R1 atau yang lebih baru
- Server Transaksi CICS untuk z/OS (CICS TS) V5.1 atau yang lebih baru (pengambilan data CICS/VSAM)
- IBM MQ 8.0 atau yang lebih baru
- Kepatuhan dengan persyaratan keamanan z/OS (misalnya, otorisasi APF untuk pustaka pemuatan) SQData
- Log pemulihan VSAM dihidupkan
- (Opsional) Versi Pemulihan CICS VSAM (CICS VR) untuk secara otomatis menangkap log CDC
- Akun AWS yang aktif
- Amazon Virtual Private Cloud (VPC) dengan subnet yang dapat dijangkau oleh platform lama Anda
- Lisensi VSAM Connect dari Tepatnya

Batasan

- Connect tidak mendukung pembuatan tabel target otomatis berdasarkan skema atau copybook sumber VSAM. Anda harus menentukan struktur tabel target untuk pertama kalinya.
- Untuk target non-streaming seperti Amazon RDS, Anda harus menentukan sumber konversi ke pemetaan target dalam skrip konfigurasi Apply Engine.
- Fungsi pencatatan, pemantauan, dan peringatan diimplementasikan melalui APIs dan memerlukan komponen eksternal (seperti Amazon CloudWatch) untuk beroperasi penuh.

Versi produk

- SQData 40134 untuk z/OS
- SQData 4.0.43 untuk Amazon Linux Amazon Machine Image (AMI) di Amazon Elastic Compute Cloud (Amazon) EC2

Arsitektur

Tumpukan teknologi sumber

- Bahasa Kontrol Pekerjaan (JCL)
- z/OS Unix shell dan Fasilitas Produktivitas Sistem Interaktif (ISPF)
- Utilitas VSAM (IDCAMS)

Tumpukan teknologi target

- Amazon EC2
- Amazon MSK
- Amazon RDS
- Amazon VPC

Arsitektur target

Memigrasi file VSAM ke Amazon RDS

Diagram berikut menunjukkan cara memigrasikan file VSAM ke database relasional, seperti Amazon RDS, secara real time atau mendekati waktu nyata dengan menggunakan CDC agent/publisher di lingkungan sumber (mainframe lokal) dan Apply Engine di lingkungan target (AWS Cloud).

Diagram menunjukkan alur kerja batch berikut:

1. Connect menangkap perubahan ke file dengan membandingkan file VSAM dari file cadangan untuk mengidentifikasi perubahan dan kemudian mengirimkan perubahan ke logstream.
2. Penerbit mengkonsumsi data dari logstream sistem.
3. Publisher mengkomunikasikan perubahan data yang ditangkap ke mesin target melalui TCP/IP. Controller Daemon mengotentikasi komunikasi antara sumber dan lingkungan target.
4. Apply Engine di lingkungan target menerima perubahan dari agen Publisher dan menerapkannya ke database relasional atau non-relasional.

Diagram menunjukkan alur kerja online berikut:

1. Connect menangkap perubahan dalam file online dengan menggunakan replika log dan kemudian mengalirkan perubahan yang ditangkap ke logstream.
2. Penerbit mengkonsumsi data dari logstream sistem.
3. Publisher mengkomunikasikan perubahan data yang ditangkap ke mesin target melalui TCP/IP. Controller Daemon mengotentikasi komunikasi antara sumber dan lingkungan target.
4. Apply Engine di lingkungan target menerima perubahan dari agen Publisher dan kemudian menerapkannya ke database relasional atau non-relasional.

Migrasi file VSAM ke Amazon MSK

Diagram berikut menunjukkan cara mengalirkan struktur data VSAM dari mainframe ke Amazon MSK dalam mode kinerja tinggi dan secara otomatis menghasilkan konversi skema JSON atau AVRO yang terintegrasi dengan Amazon MSK.

Diagram menunjukkan alur kerja batch berikut:

1. Connect menangkap perubahan ke file dengan menggunakan CICS VR atau dengan membandingkan file VSAM dari file cadangan untuk mengidentifikasi perubahan. Perubahan yang diambil dikirim ke logstream.
2. Penerbit mengkonsumsi data dari logstream sistem.
3. Publisher mengkomunikasikan perubahan data yang ditangkap ke mesin target melalui TCP/IP. Controller Daemon mengotentikasi komunikasi antara sumber dan lingkungan target.

4. Mesin Replicator yang beroperasi dalam mode pemrosesan paralel membagi data ke unit cache kerja.
5. Thread pekerja menangkap data dari cache.
6. Data dipublikasikan ke topik MSK Amazon dari utas pekerja.
7. Pengguna menerapkan perubahan dari Amazon MSK ke target seperti Amazon DynamoDB, Amazon Simple Storage Service (Amazon S3) OpenSearch , atau Amazon Service dengan menggunakan konektor.

Diagram menunjukkan alur kerja online berikut:

1. Perubahan dalam file online ditangkap dengan menggunakan replika log. Perubahan yang ditangkap dialirkan ke logstream.
2. Penerbit mengkonsumsi data dari logstream sistem.
3. Publisher mengkomunikasikan perubahan data yang ditangkap ke mesin target melalui TCP/IP. Controller Daemon mengotentikasi komunikasi antara sumber dan lingkungan target.
4. Mesin Replicator yang beroperasi dalam mode pemrosesan paralel membagi data ke unit cache kerja.
5. Thread pekerja menangkap data dari cache.
6. Data dipublikasikan ke topik MSK Amazon dari utas pekerja.
7. Pengguna menerapkan perubahan dari Amazon MSK ke target seperti DynamoDB, Amazon S3, atau Layanan dengan menggunakan konektor OpenSearch .

Alat

- Amazon Managed Streaming for Apache Kafka (Amazon MSK) adalah layanan yang dikelola sepenuhnya yang membantu Anda membangun dan menjalankan aplikasi yang menggunakan Apache Kafka untuk memproses data streaming.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.

Epik

Siapkan lingkungan sumber (mainframe)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Connect CDC 4.1.	1. Hubungi tim Support yang tepat untuk mendapatkan lisensi dan paket instalasi. 2. Gunakan contoh JCLs untuk menginstal Connect CDC 4.1. Untuk petunjuk, lihat Instal Connect CDC (SQData) menggunakan JCL dalam dokumentasi Tepatnya. 3. Jalankan `SETPROG APF` perintah untuk mengotori sasi pustaka beban Connect SQData.v4 NNN.LoadLib.	Pengembang/Admin Mainframe IBM
Siapkan direktori ZFs.	Untuk menyiapkan direktori ZFs, ikuti instruksi dari direktori variabel ZFs di dokumentasi Tepat. **Note** Konfigurasi Daemon dan Capture/Publisher agen pengontro l disimpan dalam sistem file z/OS UNIX Systems Services (disebut sebagai ZFs). Agen Controlle	Pengembang/Admin Mainframe IBM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	r Daemon, Capture, Storage, dan Publisher memerlukan struktur direktori ZFs yang telah ditentukan untuk menyimpan sejumlah kecil file.	
Konfigurasikan TCP/IP port.	Untuk mengkonfigurasi TCP/IP port, ikuti instruksi dari port TCP/IP di dokumentasi Tepat. **Note** Daemon Controller membutuhkan TCP/IP port pada sistem sumber. Port direferensikan oleh mesin pada sistem target (di mana data perubahan yang ditangkap diproses).	Pengembang/Admin Mainframe IBM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat z/OS logstream.	Untuk membuat logstream z/OS, ikuti instruksi dari Buat LogStreams z/OS sistem di dokumentasi Tepatnya. **Note** Connect menggunakan logstream untuk menangkap dan mengalirkan data antara lingkungan sumber dan lingkungan target selama migrasi. Untuk contoh JCL yang membuat z/OS LogStream, lihat Membuat z/OS sistem LogStreams dalam dokumentasi Tepatnya.	Pengembang Mainframe IBM
Identifikasi dan otorisasi IDs untuk pengguna ZF dan mulai tugas.	Gunakan RACF untuk memberikan akses ke sistem file OMVS ZFS. Untuk contoh JCL, lihat Mengidentifikasi dan mengotorisasi pengguna ZFs dan memulai tugas IDs dalam dokumentasi Tepatnya.	Pengembang/Admin Mainframe IBM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan kunci z/OS publik/pribadi dan file kunci resmi.	Jalankan JCL untuk menghasilkan key pair. Sebagai contoh, lihat Contoh pasangan kunci di bagian Informasi tambahan dari pola ini. Untuk petunjuk, lihat Menghasilkan kunci z/OS publik dan pribadi serta file kunci resmi dalam dokumentasi Tepatnya.	Pengembang/Admin Mainframe IBM

Aktifkan Replikasi Log CICS VSAM dan lampirkan ke logstream.	Jalankan skrip JCL berikut:
	
	`//STEP1 EXEC PGM=IDCAMS`
	`//SYSPRINT DD SYSOUT=*`
	`//SYSIN DD *`
	`ALTER SQDATA.CICS.FILEA -`
	`LOGSTREAMID(SQDATA.VSAMCDC.LOG1) -`
	`LOGREPLICATE`
		Pengembang/Admin Mainframe IBM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan Log Pemulihan File VSAM melalui FCT.	Ubah File Control Table (FCT) untuk mencerminkan perubahan parameter berikut: `Configure FCT Parms CEDA ALT FILE(name) GROUP(groupname) DSNAME(data set name) RECOVERY(NONE|BACKOUTONLY|ALL) FWDRECOVLOG(NO|1-99) BACKUPTYPE(STATIC|DYNAMIC) RECOVERY PARAMETERS RECOVery : None | Backoutonly | All Fwdrecovlog : No | 1-99 BAckuptype : Static | Dynamic`	Pengembang/Admin Mainframe IBM
Siapkan CDCz Log untuk agen Penerbit.	1. Buat file CAB Penerbit CDCz Log. 2. Enkripsi data yang dipublikasikan. 3. Siapkan CDCz Log Publisher Runtime JCL.	Pengembang/Admin Mainframe IBM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan Daemon Pengontrol.	1. Buka panel ISPF dan jalankan perintah berikut untuk membuka menu Tepat: `EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC$STA)' 'SQDATA.V4nnnnn'` 2. Untuk mengatur Daemon Pengontrol, pilih opsi 2 dari menu.	Pengembang/Admin Mainframe IBM
Aktifkan penerbit.	1. Buka panel ISPF dan jalankan perintah berikut untuk membuka menu Tepat: `EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC$STA)' 'SQDATA.V4nnnnn'` 2. Untuk mengatur penerbit, pilih opsi 3 dari menu dan saya untuk menyisipkan.	Pengembang/Admin Mainframe IBM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan logstream.	1. Buka panel ISPF dan jalankan perintah berikut untuk membuka menu Tepat: `EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC$STA)' 'SQDATA.V4nnnnn'` 2. Untuk mengatur logstream , pilih opsi 4 dari menu dan saya untuk menyisipkan. Kemudian, masukkan nama logstream yang dibuat pada langkah-langkah sebelumnya.	Pengembang/Admin Mainframe IBM

Siapkan lingkungan target (AWS)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Tepat pada sebuah EC2 instance.	Untuk menginstal Connect from Tepat di Amazon Linux AMI untuk Amazon EC2, ikuti petunjuk dari Install Connect CDC (SQData) pada UNIX dalam dokumentasi Tepat.	AWS Umum
Buka TCP/IP port.	Untuk memodifikasi grup keamanan untuk menyertakan port Daemon Pengontrol untuk akses masuk dan keluar, ikuti instruksi dari TCP/IP dalam dokumentasi Tepatnya.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat direktori file.	Untuk membuat direktori file, ikuti petunjuk dari Siapkan lingkungan penerapan target dalam dokumentasi Tepat.	AWS Umum

Buat file konfigurasi Apply Engine.	Buat file konfigurasi Apply Engine di direktori kerja Apply Engine. Contoh file konfigurasi berikut menunjukkan Apache Kafka sebagai target:
	`builtin.features=SASL_SCRAM`
	`security.protocol=SASL_SSL`
	`sasl.mechanism=SCRAM-SHA-512`
	`sasl.username=`
	`sasl.password=`
	`metadata.broker.list=`
	Note
	Untuk informasi selengkapnya, lihat Keamanan dalam dokumentasi Apache Kafka.
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat skrip untuk pemrosesan Apply Engine.	Buat skrip untuk Apply Engine untuk memproses data sumber dan mereplikasi data sumber ke target. Untuk informasi selengkapnya, lihat Membuat skrip mesin terapkan di dokumentasi Tepatnya.	AWS Umum
Jalankan skrip.	Gunakan `SQDENG` perintah `SQDPARSE` dan untuk menjalankan skrip. Untuk informasi selengkapnya, lihat Mengurai skrip untuk ZoS di dokumentasi Tepatnya.	AWS Umum

Validasi lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi daftar file VSAM dan tabel target untuk pemrosesan CDC.	1. Validasi file VSAM, termasuk log replikasi, log pemulihan, parameter FCT, dan logstream. 2. Validasi tabel database target, termasuk apakah tabel dibuat sesuai definisi skema yang diperlukan, akses tabel, dan kriteria lainnya.	AWS Umum, Mainframe
Verifikasi bahwa SQData produk Connect CDC ditautkan.	Jalankan pekerjaan pengujian dan verifikasi bahwa kode pengembalian dari pekerjaan ini adalah 0 (Berhasil).	AWS Umum, Mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Connect CDC SQData Apply Engine pesan status harus menampilkan pesan koneksi aktif.	

Jalankan dan validasi kasus uji (Batch)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan pekerjaan batch di mainframe.	Jalankan pekerjaan aplikasi batch menggunakan JCL yang dimodifikasi. Sertakan langkah-langkah dalam JCL yang dimodifikasi yang melakukan hal berikut: 1. Ambil cadangan file data. 2. Bandingkan file cadangan dengan file data yang dimodifikasi, buat file delta, lalu catat jumlah catatan delta dari pesan. 3. Dorong file delta ke z/OS logstream. 4. Jalankan JCL. Untuk contoh JCL, lihat Mempersiapkan file compare capture JCL dalam dokumentasi Tepatnya.	AWS Umum, Mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa logstream.	Periksa logstream untuk mengonfirmasi bahwa Anda dapat melihat data perubahan untuk pekerjaan batch mainframe yang telah selesai.	AWS Umum, Mainframe
Validasi jumlah untuk perubahan delta sumber dan tabel target.	Untuk mengonfirmasi bahwa catatan dihitung, lakukan hal berikut: 1. Kumpulkan jumlah delta sumber dari pesan JCL batch. 2. Pantau Apply Engine untuk mencatat jumlah level dari jumlah catatan yang dimasukkan, diperbarui, atau dihapus dalam file VSAM. 3. Kueri tabel target untuk jumlah catatan. 4. Bandingkan dan hitung semua jumlah rekor yang berbeda.	AWS Umum, Mainframe

Jalankan dan validasi kasus uji (Online)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan transaksi online di wilayah CICS.	1. Jalankan transaksi online untuk memvalidasi kasus uji.	Pengembang Mainframe IBM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Validasi kode eksekusi transaksi (RC = 0 - Sukses).	
Periksa logstream.	Konfirmasikan bahwa logstream diisi dengan perubahan tingkat catatan tertentu.	Pengembang AWS Mainframe
Validasi hitungan dalam database target.	Pantau Apply Engine untuk mencatat jumlah level.	Tepatnya, Linux
Validasi jumlah catatan dan catatan data dalam database target.	Kueri database target untuk memvalidasi jumlah catatan dan catatan data.	AWS Umum

Sumber daya terkait

- VSAM z/OS (Dokumentasi tepatnya)
- Terapkan mesin (Tepatnya dokumentasi)
- Mesin replikator (Dokumentasi tepatnya)
- Aliran log (dokumentasi IBM)

Informasi tambahan

Contoh file konfigurasi

Ini adalah contoh file konfigurasi untuk logstream di mana lingkungan sumber adalah mainframe dan lingkungan target adalah Amazon MSK:

```
 -- JOBNAME -- PASS THE SUBSCRIBER NAME
 -- REPORT  progress report will be produced after "n" (number) of Source records
processed.

 JOBNAME VSMTOKFK;
```

```
 --REPORT EVERY 100;
 -- Change Op has been ‘I’ for insert, ‘D’ for delete , and ‘R’ for Replace. For RDS
it is 'U' for update
 -- Character Encoding on z/OS is Code Page 1047, on Linux and UNIX it is Code Page
819 and on Windows, Code Page 1252
 OPTIONS
 CDCOP('I', 'U', 'D'),
 PSEUDO NULL = NO,
 USE AVRO COMPATIBLE NAMES,
 APPLICATION ENCODING SCHEME = 1208;

 --       SOURCE DESCRIPTIONS

 BEGIN GROUP VSAM_SRC;
 DESCRIPTION COBOL ../copybk/ACCOUNT AS account_file;
 END GROUP;

 --       TARGET DESCRIPTIONS

 BEGIN GROUP VSAM_TGT;
 DESCRIPTION COBOL ../copybk/ACCOUNT AS account_file;
 END GROUP;

 --       SOURCE DATASTORE (IP & Publisher name)

 DATASTORE cdc://10.81.148.4:2626/vsmcdct/VSMTOKFK
 OF VSAMCDC
 AS CDCIN
 DESCRIBED BY GROUP VSAM_SRC ACCEPT ALL;

 --       TARGET DATASTORE(s) - Kafka and topic name

 DATASTORE 'kafka:///MSKTutorialTopic/key'
 OF JSON
 AS CDCOUT
 DESCRIBED BY GROUP VSAM_TGT FOR INSERT;

 --       MAIN SECTION

 PROCESS INTO
 CDCOUT
 SELECT
 {
 SETURL(CDCOUT, 'kafka:///MSKTutorialTopic/key')
```

```
  REMAP(CDCIN, account_file, GET_RAW_RECORD(CDCIN, AFTER), GET_RAW_RECORD(CDCIN,
 BEFORE))
  REPLICATE(CDCOUT, account_file)
  }
  FROM CDCIN;
```

Contoh pasangan kunci

Ini adalah contoh bagaimana menjalankan JCL untuk menghasilkan key pair:

```
//SQDUTIL EXEC PGM=SQDUTIL //SQDPUBL DD DSN=&USER..NACL.PUBLIC, //
 DCB=(RECFM=FB,LRECL=80,BLKSIZE=21200), // DISP=(,CATLG,DELETE),UNIT=SYSDA, //
 SPACE=(TRK,(1,1)) //SQDPKEY DD DSN=&USER..NACL.PRIVATE, //
 DCB=(RECFM=FB,LRECL=80,BLKSIZE=21200), // DISP=(,CATLG,DELETE),UNIT=SYSDA, //
 SPACE=(TRK,(1,1)) //SQDPARMS DD  keygen //SYSPRINT DD SYSOUT= //SYSOUT DD SYSOUT=* //
SQDLOG DD SYSOUT=* //*SQDLOG8 DD DUMMY
```

Modernisasi aplikasi CardDemo mainframe dengan menggunakan AWS Transform

Santosh Kumar Singh dan Cheryl du Preez, Amazon Web Services

Ringkasan

AWS Transformdirancang untuk mempercepat modernisasi aplikasi mainframe. Ini menggunakan AI generatif untuk merampingkan proses modernisasi mainframe. Ini mengotomatiskan tugas-tugas kompleks, seperti: analisis kode warisan, dokumentasi mainframe, ekstraksi aturan bisnis, dekomposisi aplikasi monolitik ke dalam domain bisnis, dan refactoring kode. Ini mempercepat proyek modernisasi dengan mengotomatiskan tugas-tugas kompleks, seperti analisis aplikasi dan perencanaan urutan migrasi. Saat menguraikan aplikasi monolitik, urutkan transformasi aplikasi mainframe AWS Transform secara cerdas, yang membantu Anda mengubah fungsi bisnis secara paralel. AWS Transform dapat mempercepat pengambilan keputusan dan meningkatkan kelincahan operasional dan efisiensi migrasi.

Pola ini menawarkan step-by-step instruksi untuk membantu Anda menguji kemampuan modernisasi mainframe AWS Transform dengan menggunakan CardDemo, yang merupakan contoh aplikasi mainframe open source.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS IAM Identity Center, diaktifkan
- Izin yang memungkinkan administrator mengaktifkan AWS Transform
- Izin yang memungkinkan administrator menerima permintaan koneksi Amazon Simple Storage Service (Amazon S3) untuk aplikasi web AWS Transform

Batasan

- AWS Transform hanya tersedia di beberapa Wilayah AWS. Untuk daftar lengkap Wilayah yang didukung, lihat Wilayah yang Didukung untuk AWS Transform.
- AWS Transform mendukung analisis kode, pembuatan dokumen, ekstraksi aturan bisnis, dekomposisi, dan refactoring dari Common Business-Oriented Language (COBOL) ke Jawa. Untuk

informasi selengkapnya, lihat Kemampuan dan fitur utama serta Jenis file yang didukung untuk transformasi aplikasi mainframe.

- Ada kuota layanan untuk kapabilitas transformasi mainframe di. AWS Transform Untuk informasi lebih lanjut, lihat Kuota untuk AWS Transform.
- Untuk berkolaborasi di ruang kerja bersama, semua pengguna harus menjadi pengguna terdaftar dari contoh yang sama AWS IAM Identity Center yang terkait dengan instance AWS Transform aplikasi web Anda.
- Bucket Amazon S3 dan AWS Transform harus berada di tempat yang sama Akun AWS dan Wilayah.

Arsitektur

Diagram berikut menunjukkan arsitektur yang Anda atur dalam pola ini.

Diagram menunjukkan alur kerja berikut:

1. AWS Transform menggunakan konektor untuk mengakses aplikasi CardDemo mainframe, yang disimpan dalam bucket Amazon S3.
2. AWS Transform digunakan AWS IAM Identity Center untuk mengelola akses pengguna dan otentikasi. Sistem ini menerapkan beberapa lapisan kontrol keamanan untuk otentikasi, otorisasi, enkripsi, dan manajemen akses untuk membantu melindungi kode dan artefak selama pemrosesan. Pengguna berinteraksi dengan AWS Transform agen melalui antarmuka obrolan. Anda dapat memberikan instruksi kepada agen AI untuk tugas-tugas tertentu dalam bahasa Inggris. Untuk informasi lebih lanjut, lihat Human in the loop (HITL) dalam AWS Transform dokumentasi.
3. Agen AI menafsirkan instruksi pengguna, membuat rencana kerja, membagi pekerjaan menjadi tugas yang dapat dieksekusi, dan bertindak secara mandiri. Pengguna dapat meninjau dan menyetujui transformasi. Tugas transformasi meliputi:
   - Analisis kode — AWS Transform menganalisis kode di setiap file untuk detail seperti nama file, jenis file, baris kode, dan jalurnya. Agen menganalisis kode sumber, menjalankan klasifikasi, membuat pemetaan ketergantungan, dan mengidentifikasi artefak yang hilang. Ini juga mengidentifikasi komponen duplikat.
   - Pembuatan dokumen — AWS Transform menghasilkan dokumentasi untuk aplikasi mainframe. Dengan menganalisis kode, secara otomatis dapat membuat dokumentasi terperinci dari

program aplikasi, termasuk deskripsi logika bisnis, alur, integrasi, dan dependensi yang ada dalam sistem lama Anda.

- Ekstraksi logika bisnis — AWS Transform menganalisis program COBOL untuk mendokumentasikan logika bisnis inti mereka, untuk membantu Anda memahami logika bisnis mendasar.
- Dekomposisi kode — AWS Transform menguraikan kode menjadi domain yang memperhitungkan dependensi antara program dan komponen. Mengelompokkan file dan program terkait dalam domain yang sama meningkatkan organisasi dan membantu melestarikan struktur logis aplikasi saat memecahnya menjadi komponen yang lebih kecil.
- Perencanaan gelombang migrasi — Berdasarkan domain yang Anda buat selama fase dekomposisi, buat rencana gelombang AWS Transform migrasi dengan urutan modernisasi yang direkomendasikan.
- Code refactoring — AWS Transform memfaktorkan ulang kode di semua atau file domain yang dipilih ke dalam kode Java. Tujuan dari langkah ini adalah untuk melestarikan logika bisnis kritis aplikasi sambil memfaktorkannya ke aplikasi Java yang dimodernisasi dan dioptimalkan di cloud.

4. AWS Transform menyimpan kode refactored, dokumen yang dihasilkan, artefak terkait, dan pustaka runtime di bucket Amazon S3 Anda. Anda dapat melakukan tindakan berikut:
   - Akses folder runtime di bucket Amazon S3 Anda.
   - Buat dan terapkan aplikasi dengan mengikuti Build dan deploy aplikasi modern pasca-refactoring Anda dalam dokumentasi. AWS Transform
   - Melalui antarmuka obrolan, minta dan unduh sampel AWS CloudFormation, AWS Cloud Development Kit (AWS CDK), atau template Hashicorp Terraform. Template ini dapat membantu Anda menyebarkan AWS sumber daya yang diperlukan untuk mendukung aplikasi refactored.
   - Gunakan Reforge untuk meningkatkan kualitas kode refactored dengan menggunakan model bahasa besar (). LLMs Mesin refactoring mempertahankan kesetaraan fungsional COBOL sambil mengubahnya menjadi kode Java. Reforge adalah langkah opsional yang tersedia setelah transformasi. Langkah ini digunakan LLMs untuk merestrukturisasi kode agar sangat mirip dengan Java asli, yang dapat meningkatkan keterbacaan dan pemeliharaan. Reforge juga menambahkan komentar yang dapat dibaca manusia untuk membantu Anda memahami kode, dan menerapkan pola pengkodean modern dan praktik terbaik.

Alat

Layanan AWS

- AWS Transformmenggunakan AI agen untuk membantu Anda mempercepat modernisasi beban kerja lama, seperti .NET, mainframe, dan beban kerja. VMware
- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke aplikasi Anda Akun AWS dan cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Anda dapat menggunakan aplikasi AWS CardDemomainframe open source sebagai contoh aplikasi untuk memulai modernisasi mainframe.

Praktik terbaik

- Mulai dari yang kecil — Mulailah dengan kode kecil yang kurang kompleks (15.000-20.000 baris kode) untuk mendapatkan pemahaman tentang bagaimana AWS Transform menganalisis dan mengubah aplikasi mainframe.
- Kombinasikan dengan keahlian manusia — Gunakan AWS Transform sebagai akselerator sambil menerapkan keahlian manusia untuk hasil yang optimal.
- Tinjau dan uji secara menyeluruh — Selalu tinjau kode yang diubah dengan hati-hati dan jalankan tes komprehensif untuk memvalidasi kesetaraan fungsional setelah transformasi.
- Berikan umpan balik — Untuk memberikan umpan balik dan saran untuk perbaikan, gunakan tombol Kirim umpan balik di Konsol Manajemen AWS atau buat kasus dengan AWS Dukungan. Untuk informasi selengkapnya, lihat Membuat kasus dukungan. Masukan Anda sangat berharga untuk peningkatan layanan dan pengembangan masa depan.

Epik

Siapkan aplikasi mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket.	Buat bucket Amazon S3 di tempat yang sama Akun AWS dan Wilayah yang AWS Transform diaktifkan.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda menggunakan bucket ini untuk menyimpan kode aplikasi mainframe, dan AWS Transform menggunakan bucket ini untuk menyimpan dokumen yang dihasilkan, kode refactored, dan file lain yang terkait dengan transform asi. Untuk petunjuknya, lihat Membuat bucket di dokumenta si Amazon S3.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan contoh aplikasi mainframe.	1. Masukkan perintah berikut untuk mengkloning CardDemorepositori ke workstation lokal Anda: `git clone https://github.com/aws-samples/aws-mainframe-modernization-carddemo.git` 2. Buat folder baru bernama`carddemo`. 3. Salin `app` folder, yang berisi kode sumber mainframe, dari repositori kloning ke dalam folder. `carddemo` 4. Kompres `cardemo` folder menjadi file ZIP. 5. Unggah file ZIP ke bucket Amazon S3 yang Anda buat. Untuk petunjuk, lihat Mengunggah objek dalam dokumentasi Amazon S3.	Pengembang aplikasi, DevOps insinyur

Konfigurasikan Pusat Identitas IAM dan AWS Transform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan pengguna ke Pusat Identitas IAM.	Tambahkan calon pengguna Anda ke IAM Identity Center. Ikuti petunjuk di Menambahkan pengguna di Pusat	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Identitas IAM dalam AWS Transform dokumentasi.	
Aktifkan AWS Transform dan tambahkan pengguna.	1. Aktifkan AWS Transform . Ikuti petunjuk di Aktifkan AWS Transform. 2. Tambahkan pengguna ke AWS Transform. Ikuti petunjuk di Menambahkan pengguna ke AWS Transform.	Administrator AWS
Konfigurasikan akses pengguna ke aplikasi AWS Transform web.	Setiap pengguna harus menerima undangan untuk mengakses aplikasi AWS Transform web. Ikuti instruksi dalam Menerima undangan dalam AWS Transform dokumentasi.	Pengembang aplikasi, Pemilik aplikasi
Masuk ke aplikasi AWS Transform web.	Ikuti instruksi di Masuk ke AWS Transform.	Pengembang aplikasi, Pemilik aplikasi
Siapkan ruang kerja.	Siapkan ruang kerja tempat pengguna dapat berkolaborasi dalam AWS Transform aplikasi web. Ikuti petunjuk dalam Menyiapkan ruang kerja Anda di AWS Transform dokumentasi.	Administrator AWS

Ubah aplikasi mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pekerjaan transformasi.	Buat pekerjaan transformasi untuk memodernisasi aplikasi CardDemo mainframe. Untuk petunjuk, lihat Membuat dan memulai pekerjaan di AWS Transform dokumentasi. Ketika Anda diminta untuk menetapkan tujuan dalam antarmuka AWS Transform obrolan, pilih Lakukan modernisasi mainframe (IBM z/OS ke AWS) dan kemudian pilih Analyze code, Generate technical documentation, Business logic, Decompose code, Plan Migration sequence dan Transform code to Java.	Pengembang aplikasi, Pemilik aplikasi
Siapkan konektornya.	Buat konektor ke bucket Amazon S3 yang berisi aplikasi CardDemo mainframe . Konektor ini memungkin kan AWS Transform untuk mengakses sumber daya dalam bucket dan melakukan fungsi transformasi berturut-turut. Untuk petunjuk, lihat Mengatur konektor dalam AWS Transform dokumentasi.	Administrator AWS
Lakukan analisis kode.	1. Pada halaman Tentukan lokasi aset, masukkan path	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bucket Amazon S3 untuk file `carddemo` ZIP yang Anda unggah. 2. Pilih Setujui dan kirim ke AWS Transform. Ini mulai menganalisis kode. 3. Pantau status pada tab Worklog. 4. Saat analisis selesai, di bawah Analisis kode di panel navigasi kiri, pilih Lihat hasil analisis kode. 5. (Opsional) Pilih Unduh untuk mengunduh daftar aset lengkap, kode sumber yang hilang, dan file ketergantungan. Untuk informasi selengkapnya, lihat Analisis kode dalam AWS Transform dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan dokumentasi teknis.	1. Di panel navigasi kiri, di bawah Hasilkan dokumentasi teknis, pilih Pilih file dan konfigurasikan pengaturan. 2. Perluas COBOL atau JCL, lalu pilih satu atau beberapa file. 3. Pilih tingkat detail dokumentasi: • Ringkasan — Memberikan ikhtisar tingkat tinggi dari setiap file dalam ruang lingkup. Juga, berikan ringkasan satu baris dari setiap file. • Spesifikasi fungsional terperinci - Memberikan detail komprehensif untuk setiap file dalam lingkup transformasi aplikasi mainframe. Beberapa detail termasuk logika dan aliran, aturan bisnis yang diidentifikasi, aliran data, dependensi, pemrosesan input dan output, dan berbagai detail transaksi. 4. Pilih Kirim ke AWS Transform. 5. Pantau kemajuan di tab Worklog.	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Waktu pembuatan dokumen bervariasi dari menit ke jam, tergantung pada jumlah file dan baris kode. 6. Setelah selesai, pilih Tinjau hasil dokumentasi. 7. Akses dokumen yang dihasilkan melalui aplikasi web atau unduh. Dokumentasi yang dihasilkan juga disimpan di bucket Amazon S3. 8. Jelajahi dokumentasi yang dihasilkan dengan mengajukan pertanyaan spesifik di antarmuka obrolan. Misalnya, Anda dapat mengatakan "Katakan padaku fitur CBACT01 C." Untuk informasi selengkapnya, lihat Menghasilkan dokumentasi teknis dalam AWS Transform dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak logika bisnis.	1. Di panel navigasi kiri, perluas Ekstrak logika bisnis, lalu pilih Pilih file untuk ekstraksi logika bisnis. 2. Perluas COBOL atau JCL, lalu pilih satu atau beberapa file. 3. Pilih Kirim ke AWS Transform. 4. Pantau kemajuan di tab Worklog. **Note** Waktu ekstraksi logika bisnis bervariasi dari menit ke jam, tergantung pada jumlah file dan baris kode. 5. Ketika ekstraksi selesai, pilih Tinjau hasil ekstraksi logika bisnis. 6. Pilih file COBOL atau JCL, lalu pilih Lihat. 7. Ubah ruang lingkup dalam Diagram alir dan aturan fungsional untuk meninjau berbagai tingkat diagram alir, termasuk kode pendukung.	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. (Opsional) Tinjau logika bisnis yang dihasilkan, yang disimpan dalam format JSON di bucket Amazon S3 Anda. Untuk informasi selengkap nya, lihat Mengekstrak logika bisnis dalam AWS Transform dokumentasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menguraikan kode.	1. Di panel navigasi kiri, perluas kode Dekomposisi, lalu pilih Dekomposisi menjadi domain. 2. Dalam daftar Tindakan, pilih Buat domain. 3. Masukkan nama untuk domain baru, seperti `Bill payment`. 4. (Opsional) Berikan deskripsi. 5. Di bilah pencarian Cari file, cari `CB00`, lalu pilih file. 6. Pilih Tandai sebagai benih. 7. Verifikasi bahwa bendera benih berubah dari Tidak menjadi Ya. 8. Pilih Buat. 9. Pilih Simpan. 10. Pilih Dekomposisi. 11. Tinjau output dekomposisi dengan memilih nama domain. 12. Ketika dekomposisi selesai, pilih Kirim ke AWS Transform. Untuk informasi lebih lanjut tentang dekomposisi dan benih, lihat Dekomposisi dalam dokumentasi. AWS Transform	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rencanakan gelombang migrasi.	Rencanakan gelombang migrasi untuk CardDemo aplikasi. Ikuti petunjuk dalam perencanaan gelombang Migrasi dalam AWS Transform dokumentasi untuk meninjau dan mengedit rencana gelombang.	Pengembang aplikasi, Pemilik aplikasi
Memfaktorkan ulang kode.	1. Refactor aplikasi CardDemo mainframe ke dalam kode Java di semua atau file domain yang dipilih. Ikuti instruksi dalam kode Refactor dalam AWS Transform dokumentasi. 2. Setelah proses refactoring selesai, bangun dan terapkan aplikasi Java yang dimodernisasi. Ikuti petunjuk di Build dan terapkan aplikasi modern pasca-refactoring dalam dokumentasi. AWS Transform	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Gunakan Reforge untuk meningkatkan kode Java.	1. Buat pekerjaan baru, dan kemudian masukkan tujuan`Reforge the code`. 2. Masukkan proyek yang telah dimodernisasi. AWS Transform Proyek ini harus ada di bucket Amazon S3 Anda. 3. Masukkan daftar kelas Java yang menentukan kelas layanan mana yang akan direformasi. 4. Tinjau output yang dihasilkan di bucket Amazon S3. Untuk informasi selengkapnya, lihat Reforge di dokumentasi. AWS Transform	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Merampingkan penyebaran.	AWS Transform dapat menyediakan infrastruktur sebagai templat kode (IAc) untuk CloudFormation, AWS CDK, atau Terraform . Template ini membantu Anda menerapkan komponen inti, termasuk komputasi, database, penyimpanan, dan sumber daya keamanan. 1. Di antarmuka AWS Transform obrolan, minta templat dengan memasukkan`Share the mainframe refactor <service> templates` , di `<service>` mana salah satu layanan IAc yang didukung. 2. Menyebarkan template dengan menggunakan layanan pilihan Anda. Misalnya, lihat sumber daya berikut: • Buat tumpukan dari CloudFormation konsol (CloudFormation dokumentasi) • Menyebarkan AWS CDK aplikasi (AWS CDK dokumentasi) • Membangun infrastruktur (dokumentasi Terraform)	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya, lihat Kemampuan penerapan dalam AWS Transform dokumentasi.	

Pemecahan masalah

Isu	Solusi
Anda tidak dapat melihat kode sumber atau dokumen yang dihasilkan dalam aplikasi AWS Transform web.	Tambahkan kebijakan ke izin CORS untuk bucket Amazon S3 untuk AWS Transform mengizinkan sebagai asal. Untuk informasi selengkapnya, lihat izin CORS bucket S3 dalam dokumentasi. AWS Transform

Sumber daya terkait

AWS dokumentasi

- Transformasi aplikasi mainframe (AWS Transform dokumentasi)

AWS Sumber daya lainnya

- Percepat Perjalanan Modernisasi Mainframe Anda menggunakan Agen AI dengan AWS Transform(posting blog)AWS
- AWS Transform FAQs
- AWS IAM Identity Center FAQs

Video dan tutorial

- Pengantar Pengembang Amazon Q: Transform (Pembuat AWS Keterampilan)
- AWS re:invent 2024 - Modernisasi aplikasi mainframe lebih cepat menggunakan Amazon Q Developer () YouTube

- AWS re:invent 2024 - Mengotomatiskan migrasi dan modernisasi untuk mempercepat transformasi () YouTube
- AWS re:invent 2024 - Toyota mendorong inovasi & meningkatkan efisiensi operasional dengan gen AI () YouTube

> **Note**
>
> AWS Transform Sebelumnya dikenal sebagai Amazon Q Developer transform untuk mainframe.

Memodernisasi dan menyebarkan aplikasi mainframe menggunakan dan Terraform AWS Transform

Mason Cahill, Polaris Jhandi, Prachi Khanna, Ramani Sivasubrama, dan Santosh Kumar Singh, Amazon Web Services

Ringkasan

AWS Transformdapat mempercepat modernisasi skala besar .NET, mainframe, dan beban kerja. VMware Ini menyebarkan agen AI khusus yang mengotomatiskan tugas-tugas kompleks seperti penilaian, analisis kode, refactoring, dekomposisi, pemetaan ketergantungan, validasi, dan perencanaan transformasi. Pola ini menunjukkan cara menggunakan AWS Transform untuk memodernisasi aplikasi mainframe dan kemudian menerapkannya ke AWS infrastruktur dengan menggunakan Hashicorp Terraform. step-by-stepInstruksi ini membantu Anda mengubah CardDemo, yang merupakan contoh aplikasi mainframe open source, dari COBOL ke aplikasi Java modern.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Izin administratif untuk membuat AWS sumber daya dan menyebarkan aplikasi
- Terraform versi 1.5.7 atau lebih tinggi, dikonfigurasi
- AWS Penyedia untuk Terraform, dikonfigurasi
- AWS IAM Identity Center, diaktifkan
- AWS Transform, diaktifkan
- Pengguna, yang terhubung ke AWS Transform ruang kerja dengan peran kontributor yang dapat menjalankan pekerjaan transformasi

Batasan

- AWS Transform hanya tersedia di beberapa Wilayah AWS. Untuk daftar lengkap Wilayah yang didukung, lihat Wilayah yang Didukung untuk AWS Transform.
- Ada kuota layanan untuk kapabilitas transformasi mainframe di. AWS Transform Untuk informasi lebih lanjut, lihat Kuota untuk AWS Transform.

- Untuk berkolaborasi di ruang kerja bersama, semua pengguna harus menjadi pengguna terdaftar dari contoh yang sama AWS IAM Identity Center yang terkait dengan instance AWS Transform aplikasi web Anda.
- Bucket Amazon Simple Storage Service (Amazon S3) AWS Transform dan harus berada di tempat yang sama dan Wilayah. Akun AWS

Arsitektur

Diagram berikut menunjukkan end-to-end modernisasi aplikasi warisan dan penyebaran ke. AWS Cloud Kredensi aplikasi dan database disimpan AWS Secrets Manager, dan Amazon CloudWatch menyediakan kemampuan pemantauan dan pencatatan.

Diagram menunjukkan alur kerja berikut:

1. Melalui AWS IAM Identity Center, pengguna mengautentikasi dan mengakses AWS Transform di. Akun AWS
2. Pengguna mengunggah kode mainframe COBOL ke bucket Amazon S3 dan memulai transformasi di. AWS Transform
3. AWS Transform memodernisasi kode COBOL menjadi kode Java cloud-native dan menyimpan kode modern di bucket Amazon S3.
4. Terraform membuat AWS infrastruktur untuk menerapkan aplikasi modern, termasuk Application Load Balancer, instans Amazon Elastic Compute Cloud (Amazon EC2), dan database Amazon Relational Database Service (Amazon RDS). Terraform menerapkan kode modern ke instans Amazon EC2.
5. File VSAM diunggah ke Amazon EC2 dan dimigrasikan dari Amazon EC2 ke database Amazon RDS.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah. Dalam pola ini, instance cluster failover SQL Server diinstal pada instans Amazon EC2.

- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke aplikasi Anda Akun AWS dan cloud.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di. AWS Cloud
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Transformmenggunakan AI agen untuk membantu Anda mempercepat modernisasi beban kerja lama, seperti .NET, mainframe, dan beban kerja. VMware

Alat-alat lainnya

- Apache Maven adalah manajemen proyek perangkat lunak open source dan membangun alat otomatisasi untuk proyek Java.
- Apache Tomcat adalah wadah Servlet open source dan server web untuk kode Java.
- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- Spring Boot adalah kerangka kerja open source yang dibangun di atas Spring Framework di Jawa.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Mainframe Transformation E2E. Pola ini menggunakan aplikasi AWS CardDemomainframe open source sebagai contoh aplikasi.

Praktik terbaik

- Tetapkan kepemilikan penuh kode dan sumber daya yang ditargetkan untuk migrasi.
- Kembangkan dan uji bukti konsep sebelum melakukan penskalaan ke migrasi penuh.
- Menjamin komitmen dari semua pemangku kepentingan.
- Membangun saluran komunikasi yang jelas.
- Menentukan dan mendokumentasikan persyaratan produk minimum yang layak (MVP).
- Tetapkan kriteria keberhasilan yang jelas.

Epik

Siapkan dan unggah kode aplikasi mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket.	Buat bucket Amazon S3 di tempat yang sama Akun AWS dan Wilayah yang AWS Transform diaktifkan. Anda menggunakan bucket ini untuk menyimpan kode aplikasi mainframe, data, dan skrip tambahan yang diperluka n untuk membangun dan menjalankan aplikasi. AWS Transform menggunakan bucket ini untuk menyimpan kode refactored dan file lain yang terkait dengan transform asi. Untuk petunjuknya, lihat Membuat bucket di dokumenta si Amazon S3.	AWS umum, administrator AWS
Setel izin CORS untuk bucket.	Saat menyiapkan bucket untuk AWS Transform akses, Anda perlu mengonfigurasi berbagi sumber daya lintas asal (CORS) untuk bucket. Jika ini tidak diatur dengan benar, Anda mungkin tidak dapat menggunakan fungsi tampilan sebaris atau perbandin gan file. AWS Transform Untuk petunjuk tentang cara mengonfigurasi CORS untuk bucket, lihat Menggunakan	AWS umum, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	berbagi sumber daya lintas asal di bucket Amazon S3. Untuk kebijakan ini, lihat izin CORS bucket S3 di dokumentasi. AWS Transform	
Siapkan contoh kode aplikasi mainframe.	1. Masukkan perintah berikut untuk mengkloning CardDemorepositori ke workstation lokal Anda: `git clone https://github.com/aws-samples/aws-mainframe-modernization-carddemo.git` 2. Kompres `aws-mainframe-modernization-carddemo` folder menjadi file ZIP. 3. Unggah file ZIP ke bucket Amazon S3 yang Anda buat. Untuk petunjuk, lihat Mengunggah objek dalam dokumentasi Amazon S3.	AWS Umum, Pengembang aplikasi

Ubah aplikasi mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS Transform pekerjaan.	1. Akses aplikasi AWS Transform web dengan masuk dengan kredensil Anda.	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Buat ruang kerja baru dengan mengikuti petunjuk di Menyiapkan ruang kerja Anda di dokumentasi. AWS Transform 3. Di halaman landing ruang kerja Anda, pilih Minta AWS Transform untuk membuat pekerjaan. 4. Selanjutnya, pilih Modernisasi Mainframe sebagai jenis pekerjaan. 5. Di jendela obrolan, masukkan Transform code ke Java. 6. Tinjau jenis pekerjaan , nama, dan tujuan yang disarankan. Untuk mengonfirmasi, masukkan Ya. 7. Pilih Buat tugas.	
Siapkan konektor.	1. Siapkan konektor dengan bucket Amazon S3 yang Anda buat. Untuk petunjuk, lihat Mengatur konektor dalam AWS Transform dokumentasi. Saat diminta, masukkan jalur untuk file `aws-mainframe-modernization-carddemo.zip` di bucket Amazon S3. 2. Tunggu analisis selesai.	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah kode.	1. Tinjau hasil analisis kode sesuai dengan instruksi dalam Analisis kode dalam AWS Transform dokumentasi. 2. Refactor kode mainframe sesuai dengan instruksi dalam kode Refactor dalam dokumentasi. AWS Transform Untuk contoh aplikasi CardDemo mainframe, Anda dapat menerima pengaturan default. 3. Tunggu hingga refactoring selesai. 4. Pilih Lihat hasil untuk melihat jalur kode refactored di bucket Amazon S3. Catat jalur file ini. Anda akan membutuhkannya nanti.	Pengembang aplikasi, Pemilik aplikasi

Menyebarkan infrastruktur melalui Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui template.	1. Masukkan perintah berikut untuk mengkloning repositori Mainframe Transformation E2E ke workstation lokal Anda:	AWS umum, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`git clone https://github.com/aws-samples/sample-mainframe-transformation-e2e.git`
	2. Masukkan perintah berikut untuk mengambil alamat IP publik Anda saat ini:
	`curl checkip.amazonaws.com`
	3. Masukkan perintah berikut untuk menavigasi ke direktori infra:
	`cd mainframe-transformation-e2e/infra`
	4. Buka file `variables.tf` .
	5. Ganti `YOUR_IP_ADDRESS_HERE` dengan alamat IP Anda.
	6. Jika Anda memiliki zona yang dihosting publik, lakukan hal berikut:
	a. Ganti `hosted_zone_name` dengan nama zona host Anda.
	b. Atur `hosted_zone_enabled` ke `true`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Jika Anda tidak memiliki zona yang dihosting publik, lakukan hal berikut:
	a. Masukkan perintah berikut untuk menghasilkan sertifikat yang ditandatangani sendiri:
	`openssl genrsa 2048 > my-private-key.pem`
	`openssl req -new -x509 -nodes -sha256 -days 365 -key my-private-key.pem -outform PEM -out my-certificate.pem`
	b. Masukkan perintah berikut untuk mengimpor sertifikat ke AWS Certificate Manager (ACM):
	`aws acm import-certificate \`
	`--certificate fileb://my-certificate.pem \`
	`--private-key fileb://my-private-key.pem`
	Outpout dari perintah ini termasuk Amazon Resource Name (ARN)	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dari sertifikat yang diimpor. c. Ganti `self_signed_cert_arn` dengan ARN sertifikat Anda. d. Atur `hosted_zone_enabled` ke `false`. 7. Ubah `aws_region` ke Wilayah target. Nilai default-nya `us-east-1` . 8. Simpan dan tutup file `variables.tf` . **Note** Untuk lingkungan produksi atau seperti produksi, konfigurasikan komponen keamanan tambahan. Misalnya, aktifkan AWS WAF perlindungan untuk Application Load Balancer Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan infrastruktur.	1. Masukkan perintah berikut untuk menginisialisasi Terraform:
	`terraform init`
	2. Masukkan perintah berikut untuk menghasilkan rencana eksekusi:
	`terraform plan`
	3. Tinjau rencana, dan validasi sumber daya dan komponen infrastruktur yang akan dibuat.
	4. Masukkan perintah berikut untuk menyebarkan infrastruktur:
	`terraform apply`
	5. Saat diminta, masukkan ya untuk mengonfirmasi penerapan.
	6. Tunggu hingga penerapan selesai.
		Terraform

Instal dan konfigurasikan Apache Tomcat di instans Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal perangkat lunak yang diperlukan.	1. Connect ke instans Amazon EC2 Anda dengan menggunakan. AWS	Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Systems Manager Session Manager
	2. Masukkan perintah berikut untuk beralih ke pengguna root:
	`sudo su -`
	3. Masukkan perintah berikut untuk menavigasi ke `scripts` direktori:
	`cd /opt/scripts`
	4. Tinjau skrip install_software.sh. Script ini menginstal Java 17, Apache Maven, dan Apache Tomcat 10.0.23.
	5. Perbarui skrip sesuai kebutuhan untuk kasus penggunaan Anda.
	6. Masukkan perintah berikut untuk membuat skrip dapat dieksekusi:
	`chmod +x install_software.sh`
	7. Masukkan perintah berikut untuk menjalankan skrip:
	`./install_software.sh`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi instalasi perangkat lunak.	1. Masukkan perintah berikut untuk memulai server Tomcat:
	`/opt/tomcat/apache-tomcat-10.0.23/bin/startup.sh`
	2. Masukkan perintah berikut untuk memverifikasi respons server web:
	`curl http://localhost:8080`
	Output harus mengkonfirmasi bahwa Tomcat melayani halaman HTML.
		Pengembang aplikasi, insinyur Migrasi

Kompilasi dan paket kode aplikasi modern

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh dan ekstrak kode yang dihasilkan.	1. Masukkan perintah berikut untuk membuat skrip download_and_extract.sh dapat dieksekusi. Skrip ini mengunduh kode refactored dan pustaka runtime Gapwalk yang disimpan di bucket Amazon S3: `chmod +x /opt/scripts/download_and_extract.sh`	Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Masukkan perintah berikut untuk menjalankan skrip. Ganti `<file_path>` dengan path ke file generated.zip di bucket Amazon S3 Anda:
	`./download_and_extract.sh <file_path>`
	Note
	Jalur file biasanya `s3://<bucket-name>/transform_output/<aws_transform_job_id>/code transformation/generated.zip`.
	3. Masukkan perintah berikut untuk menavigasi ke `shared` folder:
	`cd /opt/runtime/velocity/shared`
	4. Masukkan perintah berikut untuk menyalin deploy-velocity-runtimeskrip.sh:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cp /opt/scripts/deploy-velocity-runtime.sh .`
	5. Masukkan perintah berikut untuk membuat skrip yang disalin dapat dieksekusi:
	`chmod +x deploy-velocity-runtime.sh`
	6. Masukkan perintah berikut untuk menjalankan skrip. Script ini menyalin semua dependensi Web Application Archive (WAR) yang diperlukan yang ada dalam file Project Object Model (POM) ke dalam folder repositori:
	`./deploy-velocity-runtime.sh`
	7. Verifikasi eksekusi yang berhasil dengan memeriksa bahwa tidak ada kesalahan dan bahwa dependensi WAR yang diperlukan diinstal di repositori Maven lokal Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bangun aplikasi yang dimodernisasi.	1. Masukkan perintah berikut untuk menavigasi ke direktori `app-pom` proyek:
	`cd /opt/codebase/app-pom/`
	2. Masukkan perintah berikut untuk menginstal Maven:
	`mvn clean install`
	3. Tunggu instalasi dan build selesai.
	Note
	Saat menjalankan perintah ini untuk CardDemo aplikasi, Anda mungkin menemukan pesan peringatan untuk `app-web` proyek. Anda dapat dengan aman mengabaikan peringatan ini.
	4. Setelah selesai membangun berhasil, konfirmasikan keberadaan `app-service/target/app-service-1.0.0.war danapp-web/target/app-web-1.0.0.war .`
		Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important Jangan restart server Tomcat pada tahap ini. Ini akan mengakibatkan kesalahan karena tidak adanya database yang diperlukan. Anda harus mengatur database sebelum Anda dapat me-restart server.	

Migrasi database

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat database dan skema JICS.	1. Masukkan perintah berikut untuk mengganti nama `csd commands` folder menjadi`csd_commands` . Ini menghapus spasi dari nama folder: `mv /opt/codebase/extra/csd\ commands/ /opt/codebase/extra/csd_commands` 2. Masukkan perintah berikut untuk menavigasi ke `scripts` direktori:	Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cd /opt/scripts`
	3. Masukkan perintah berikut untuk membuat skrip migrasi database dapat dieksekusi:
	`chmod +x database_migration_setup.sh`
	4. Masukkan perintah berikut untuk mengkonfigurasi variabel berikut sebagai parameter:
	`RDS_ENDPOINT=<database_endpoint> SECRET_NAME=<secret_name> JICS_SQL_SCRIPT_DIR=/opt/runtime/velocity/jics/sql/jics.sql INIT_JICS_SQL_SCRIPT_DIR=/opt/codebase/extra/csd_commands/sql/aws-mainframe-modernization-carddemo-main/app/csd/initJics.sql`
	Di mana:
	• `<database_endpoint>` adalah titik akhir dari database Amazon RDS yang Anda terapkan melalui Terraform.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• `<secret_name>` adalah nama AWS Secrets Manager rahasia yang Anda sebarkan melalui Terraform.
	5. Masukkan perintah berikut untuk menjalankan skrip migrasi database:
	`./database_migration_setup.sh $RDS_ENDPOINT $SECRET_NAME $JICS_SQL_SCRIPT_DIR $INIT_JICS_SQL_SCRIPT_DIR`
	6. Masukkan perintah berikut untuk menyambung ke database dari instans Amazon EC2 Anda:
	`psql -h <Your Amazon RDS Endpoint> -U foo -p 5432 postgres`
	7. Saat diminta, masukkan kredensil database Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Validasi pembuatan database.	1. Masukkan perintah berikut untuk melihat semua database:
	`\l`
	2. Masukkan perintah berikut untuk beralih ke `jics` database:
	`\c jics`
	3. Masukkan perintah berikut untuk meninjau daftar tabel yang dibuat:
	`\dt`
		Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Migrasikan data ke database JICS.	1. Masukkan perintah berikut untuk membuat skrip execute_listcat_sql.sh dapat dieksekusi:
	`chmod +x execute_listcat_sql.sh`
	2. Masukkan perintah berikut untuk mengkonfigurasi `PATH_TO_LISTCAT_SQL_FILES` variabel, yang merupakan direktori yang berisi file SQL LISTCAT Anda:
	`PATH_TO_LISTCAT_SQL_FILES=/opt/codebase/extra/listcat/sql/cluster/aws-mainframe-modernization-carddemo-main/app/catlg`
	3. Pastikan bahwa`RDS_ENDPOINT` ,`SECRET_NAME` , dan `PATH_TO_LISTCAT_SQL_FILES` variabel diatur dengan benar sesuai dengan instruksi sebelumnya.
	4. Masukkan perintah berikut untuk menjalankan skrip execute_listcat_sql.sh:
	`./execute_listcat_sql.sh $RDS_ENDP`
		Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`OINT $SECRET_NAME $PATH_TO_LISTCAT_SQL_FILES` Skrip ini memperbarui properti file VSAM dalam database JICS dan menjalankan kueri yang diperlukan untuk memodifikasi database.	

Instal aplikasi yang dimodernisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal aplikasi modern pada instans Amazon EC2.	1. Masukkan perintah berikut untuk membuat skrip application_installer.sh dapat dieksekusi:
	
	`chmod +x /opt/scripts/application_installer.sh`
	
	2. Masukkan perintah berikut untuk mengkonfigurasi variabel berikut sebagai parameter:
	
	`RDS_ENDPOINT=<database_endpoint> SECRET_NAME=<secret_name> AIX_JSON_FILE_PATH=/opt/codebase/extra/csd_commands/json/aws-mainframe-moder`
		Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>nization-carddemo-main/jicsFileAix.json
	LISTCAT_JSON_FILES_DIR=/opt/codebase/extra/listcat/json/cluster/default/aws-mainframe-modernization-carddemo-main/app/catlg
	S3_PATH_FOR_EBCDIC_DATA_FILES=s3://<bucket_name>/transform-output/<job_id>/inputs/aws-mainframe-modernization-carddemo-main/app/data/EBCDIC</pre> Di mana:
	• <database_endpoint> adalah titik akhir dari database Amazon RDS yang Anda terapkan melalui Terraform.
	• <secret_name> adalah nama AWS Secrets Manager rahasia yang Anda sebarkan melalui Terraform.
	• <bucket_name> adalah nama bucket Amazon S3 yang berisi aplikasi modern.
	• <job_id>adalah ID AWS Transform pekerjaan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Masukkan perintah berikut untuk menjalankan skrip application_installer.sh:
	`./application_installer.sh $RDS_ENDPOINT $SECRET_NAME $AIX_JSON_FILE_PATH $LISTCAT_JSON_FILES_DIR $S3_PATH_FOR_EBCDIC_DATA_FILES`
	4. Di `/opt/tomcat/apache-tomcat-10.0.23/workingdir/config` folder, di application-utility-pgmfile.yl, ubah parameter pengkodean menjadi yang berikut:
	`encoding : CP1047`
	Note
	Saat Anda memfaktorkan ulang aplikasi secara AWS Transform otomatis dengan menggunakan AWS Blu Age, Anda mengonfigurasi aplikasi dan lingkungan runtime-nya	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	melalui file YAMM. Misalnya, Anda dapat mengonfig urasi logging di file application-main.yml. untuk aplikasi. Untuk informasi selengkap nya tentang properti yang tersedia, lihat Mengaktifkan properti untuk AWS Blu Age Runtime.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulai ulang server Tomcat.	1. Masukkan perintah berikut untuk menavigasi ke direktori kerja:
	`cd /opt/tomcat/apache-tomcat-10.0.23/workingdir`
	2. Masukkan perintah berikut untuk berhenti dan memulai server Tomcat:
	`/opt/tomcat/apache-tomcat-10.0.23/bin/shutdown.sh`
	`/opt/tomcat/apache-tomcat-10.0.23/bin/startup.sh`
	3. Masukkan perintah berikut untuk memantau log startup layanan Tomcat:
	`tail -f /opt/tomcat/apache-tomcat-10.0.23/logs/catalina.out`
		Pengembang aplikasi, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Migrasikan kumpulan data VSAM.	1. Buka konsol Amazon EC2. 2. Di panel navigasi, pilih Load balancer. 3. Pilih penyeimbang beban yang dibuat melalui Terraform. 4. Temukan nama Domain Name System (DNS) dari Application Load Balancer Anda, seperti. `application-load-balancer-<id>.<region>.elb.amazonaws.com` 5. Di browser Anda, navigasikan kehttp://<dns_name>/gapwalk-application/scripts/data-load , di <dns_name> mana nama DSN dari Application Load Balancer. Ini memulai skrip pemuatan data. 6. Tunggu skrip selesai. Setelah selesai, browser akan menampilkan SELESAI. 7. Pada instans Amazon EC2, buka terminal baru. 8. Masukkan perintah berikut untuk terhubung ke database Amazon RDS, ganti <database	Pengembang aplikasi, insinyur Migrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`_endpoint>` dengan nilai Anda:
	`psql -h <database _endpoint> -U foo -p 5432 postgres`
	9. Saat diminta, masukkan kredensil Anda untuk terhubung ke database.
	10. Masukkan perintah berikut untuk melihat semua database:
	`\l`
	11. Masukkan perintah berikut untuk beralih ke `bluesam` database:
	`\c bluesam`
	12. Masukkan perintah berikut untuk meninjau daftar tabel yang dibuat:
	`\dt`
	13. Masukkan perintah berikut untuk memvalidasi beban data:
	`SELECT * FROM public.aws_m2_card demo_usrsec_vsam_k sds;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Output harus menunjukkan 10 catatan yang dikembali kan.	
Perbarui parameter dalam skrip Groovy.	1. Masukkan perintah berikut untuk menavigasi ke direktori skrip: `cd /opt/tomcat/apache-tomcat-10.0.23/webapps/workingdir/scripts` 2. Di semua file Groovy yang berisi referensi ke file datar, perbarui konfigurasi file berikut: • Jalur file lokal - Jalur ke file datar di direktori lokal instans Amazon EC2 Anda • Jenis sistem file — Sistem file yang berisi file datar • Ukuran rekaman - Ukuran file datar Untuk referensi, lihat contoh skrip Dusrsecj.jcl.Groovy di repositori kode. 3. Simpan dan tutup file.	Pengembang aplikasi

Uji aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji aplikasi yang dimoderni sasi.	1. Akses aplikasi online melalui Application Load Balancer (`http://<your-load-balancer-dns>`) atau melalui zona host (`https://myhostedzone.dev/`). 2. Untuk ID transaksi, masukkan`CC00`. 3. Untuk nama pengguna, masukkan`USER0001`. 4. Untuk kata sandi, masukkan`PASSWORD`. Setelah berhasil login, menu utama ditampilkan.	Pengembang aplikasi, Insinyur uji
Verifikasi skrip batch.	1. Akses antarmuka skrip melalui Application Load Balancer `http://<your-load-balancer-dns>/gapwalk-application/scripts` () atau melalui zona `https://myhostedzone.dev/gapwalk-application/scripts` yang dihosting (). 2. Pilih skrip untuk dijalankan, seperti skrip Dusrsecj.jcl.Groovy.	Pengembang aplikasi, Insinyur uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Verifikasi bahwa skrip berjalan dengan sukses. Berikut ini adalah contoh output setelah eksekusi berhasil. `{ "exitCode": 0, "stepName": "STEP03", "program": "IDCAMS", "status": "Succeeded" }`	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersiaplah untuk menghapus infrastruktur.	1. Masukkan perintah berikut untuk menghapus perlindungan penghapusan dari instans Amazon RDS: `aws rds modify-db-instance \ --db-instance-identifier <your-db-instance-name> \ --no-deletion-protection \ --apply-immediately` 2. Masukkan perintah berikut untuk menghapus proteksi penghapusan dari Application Load Balancer:	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>aws elbv2 modify-load-balancer-attributes \
	  --load-balancer-arn <your-load-balancer-arn> \
	  --attributes Key=deletion_protection.enabled,Value=false</pre>
	3. Masukkan perintah berikut untuk menghapus konten bucket Amazon S3:
	<pre>ACCOUNT_NUMBER=$(aws sts get-caller-identity --query Account --output text)
	aws s3 rm s3://mf-carddemo-$ACCOUNT_NUMBER --recursive
	aws s3 rm s3://mf-carddemo-logs-$ACCOUNT_NUMBER --recursive</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Hapus infrastruktur.	**⚠ Important**
	Langkah-langkah ini akan menghapus sumber daya Anda secara permanen. Pastikan Anda telah mencadangkan data penting apa pun sebelum melanjutkan.
	
	1. Masukkan perintah berikut untuk menavigasi ke `infra` folder:
	`cd /mainframe-transformation-e2e/infra`
	2. Masukkan perintah berikut untuk menghapus infrastruktur:
	`terraform destroy --auto-approve`
		AWS Umum

Pemecahan masalah

Isu	Solusi
Otentikasi Terraform	1. Pastikan AWS kredensialnya dikonfigurasi dengan benar. 2. Verifikasi bahwa Anda telah memilih AWS profil yang benar.

Isu	Solusi
	3. Konfirmasikan bahwa Anda memiliki izin yang diperlukan.
Kesalahan terkait Tomcat	1. Periksa catalina.out `/opt/tomcat/apache-tomcat-10.0.23/logs` untuk pengecualian apa pun. 2. Masukkan perintah berikut untuk mengubah kepemilikan folder Tomcat ke pengguna Tomcat: `chown -R tomcat:tomcat /opt/tomcat/*`
Nama URL tidak dimuat	Pastikan bahwa grup keamanan Application Load Balancer memiliki alamat IP Anda dalam aturan masuk sebagai sumber.
Masalah otentikasi di log Tomcat	Konfirmasikan bahwa kata sandi rahasia database masuk AWS Secrets Manager dan kata sandi di server.xml cocok.

Sumber daya terkait

AWS Bimbingan Preskriptif

- Modernisasi aplikasi CardDemo mainframe dengan menggunakan AWS Transform

AWS dokumentasi layanan

- AWS Konsol Administrasi Blu Age Blusam
- Persyaratan penyiapan infrastruktur untuk AWS Blu Age Runtime (tidak dikelola)
- Orientasi AWS Blu Age Runtime
- Modernisasi aplikasi mainframe
- Siapkan konfigurasi untuk AWS Blu Age Runtime

AWS posting blog

- Percepat Perjalanan Modernisasi Mainframe Anda menggunakan Agen AI AWS Transform

Modernisasi manajemen output mainframe AWS dengan menggunakan Rocket Enterprise Server dan LRS X PageCenter

Shubham Roy, Amazon Web Services

Abraham Rondon, Fokus Mikro

Guy Tucker, Levi, Ray dan Shoup Inc

Ringkasan

Dengan memodernisasi manajemen output mainframe Anda, Anda dapat mencapai penghematan biaya, mengurangi utang teknis pemeliharaan sistem lama, dan meningkatkan ketahanan dan kelincahan melalui serta teknologi cloud-native Amazon Web DevOps Services (AWS). Pola ini menunjukkan kepada Anda cara memodernisasi beban kerja manajemen keluaran mainframe utama bisnis Anda di AWS Cloud. Pola ini menggunakan Rocket Enterprise Server sebagai runtime untuk aplikasi mainframe modern, dengan Levi, Ray & Shoup, Inc. (LRS) VPSX/MFI (Micro Focus Interface) sebagai server cetak dan LRS X sebagai server arsip. PageCenter LRS PageCenter X menyediakan solusi manajemen keluaran untuk melihat, mengindeks, mencari, mengarsipkan, dan mengamankan akses ke output bisnis.

Pola ini didasarkan pada pendekatan modernisasi mainframe replatform. Aplikasi mainframe dimigrasikan oleh AWS Mainframe Modernization di Amazon Elastic Compute Cloud (Amazon). EC2 Beban kerja manajemen output mainframe dimigrasikan ke Amazon EC2, dan database mainframe, seperti IBM Db2 for) bekerja z/OS, is migrated to Amazon Relational Database Service (Amazon RDS). The LRS Directory Integration Server (LRS/DIS dengan AWS Directory Service untuk Microsoft Active Directory untuk otentikasi dan otorisasi alur kerja manajemen output.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Beban kerja manajemen keluaran mainframe.
- Pengetahuan dasar tentang cara membangun kembali dan mengirimkan aplikasi mainframe yang berjalan di Rocket Enterprise Server. Untuk informasi selengkapnya, lihat lembar data Rocket Enterprise Server di dokumentasi Perangkat Lunak Roket.
- Pengetahuan dasar tentang solusi dan konsep pencetakan cloud LRS. Untuk informasi selengkapnya, lihat Modernisasi Keluaran dalam dokumentasi LRS.

- Perangkat lunak dan lisensi Rocket Enterprise Server. Untuk informasi lebih lanjut, hubungi Perangkat Lunak Roket.
- VPSX/MFI, LRS PageCenterX, LRS/Queue, and LRS/DISPerangkat lunak dan lisensi LRS. Untuk informasi lebih lanjut, hubungi LRS. Anda harus memberikan nama host dari EC2 contoh di mana produk LRS akan diinstal.

Catatan: Untuk informasi selengkapnya tentang pertimbangan konfigurasi untuk beban kerja manajemen keluaran mainframe, lihat Pertimbangan di bagian Informasi tambahan dari pola ini.

Versi produk

- Server Perusahaan Roket 10.0
- LRS VPSX/LKM
- LRS PageCenter X V1R3 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Sistem operasi — IBM z/OS
- Bahasa pemrograman — Bahasa berorientasi bisnis umum (COBOL), bahasa kontrol pekerjaan (JCL), dan Sistem Kontrol Informasi Pelanggan (CICS)
- Database — IBM Db2 untuk z/OS, database Sistem Manajemen Informasi IBM (IMS), dan Metode Akses Penyimpanan Virtual (VSAM)
- Keamanan — Fasilitas Kontrol Akses Sumber Daya (RACF), CA Top Secret untuk z/OS, dan Fasilitas Kontrol Akses 2 () ACF2
- Solusi cetak dan arsip - Produk z/OS keluaran dan pencetakan mainframe IBM (IBM Infoprint Server untuk z/OS, LRS, dan CA Deliver) dan solusi pengarsipan (CA Deliver, ASG Mobius, atau CA Bundle)

Arsitektur sumber

Diagram berikut menunjukkan arsitektur keadaan saat ini yang khas untuk beban kerja manajemen output mainframe.

Diagram menunjukkan alur kerja berikut:

1. Pengguna melakukan transaksi bisnis pada sistem keterlibatan (SoE) yang dibangun di atas aplikasi IBM CICS yang ditulis dalam COBOL.
2. SoE memanggil layanan mainframe, yang mencatat data transaksi bisnis dalam database system-of-records (SoR) seperti IBM Db2 untuk z/OS.
3. SoR mempertahankan data bisnis dari BUMN.
4. Penjadwal pekerjaan batch memulai pekerjaan batch untuk menghasilkan output cetak.
5. Pekerjaan batch mengekstrak data dari database. Ini memformat data sesuai dengan kebutuhan bisnis, dan kemudian menghasilkan output bisnis seperti laporan penagihan, kartu identitas, atau laporan pinjaman. Akhirnya, pekerjaan batch merutekan output ke manajemen output untuk format, publikasi, dan penyimpanan output berdasarkan persyaratan bisnis.
6. Manajemen output menerima output dari pekerjaan batch. Manajemen output mengindeks, mengatur, dan menerbitkan output ke tujuan tertentu dalam sistem manajemen output, seperti solusi LRS PageCenter X (seperti yang ditunjukkan dalam pola ini) atau CA View.
7. Pengguna dapat melihat, mencari, dan mengambil output.

Tumpukan teknologi target

- Sistem operasi - Windows Server berjalan di Amazon EC2
- Hitung - Amazon EC2
- Penyimpanan - Amazon Elastic Block Store (Amazon EBS) dan FSx Amazon untuk Windows File Server
- Bahasa pemrograman — COBOL, JCL, dan CICS
- Basis Data - Amazon RDS
- Keamanan — AWS Managed Microsoft AD
- Pencetakan dan pengarsipan — Solusi pencetakan LRS (VPSX) dan pengarsipan (PageCenterX) di AWS
- Lingkungan runtime mainframe - Rocket Enterprise Server

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk beban kerja manajemen keluaran mainframe yang diterapkan di AWS Cloud.

Diagram menunjukkan alur kerja berikut:

1. Penjadwal pekerjaan batch memulai pekerjaan batch untuk membuat output, seperti laporan penagihan, kartu identitas, atau laporan pinjaman.
2. Pekerjaan batch mainframe (direplatformed ke Amazon EC2) menggunakan runtime Rocket Enterprise Server untuk mengekstrak data dari database aplikasi, menerapkan logika bisnis ke data, dan memformat data. Kemudian mengirimkan data ke tujuan output dengan menggunakan modul keluar printer Rocket Software (dokumentasi OpenText Micro Focus).
3. Database aplikasi (SoR yang berjalan di Amazon RDS) menyimpan data untuk output cetak.
4. Solusi VPSX/MFI pencetakan LRS digunakan di Amazon EC2, dan data operasionalnya disimpan di Amazon EBS. LRS VPSX/MFI menggunakan agen TCP/IP-based LRS/Queue transmisi untuk mengumpulkan data output melalui Rocket Software JES Print Exit API.

   LRS VPSX/MFI melakukan preprocessing data, seperti terjemahan EBCDIC ke ASCII. Ini juga melakukan tugas yang lebih kompleks, termasuk mengubah aliran data eksklusif mainframe seperti IBM Advanced Function Presentation (AFP) dan Xerox Line Conditioned Data Stream (LCDS) menjadi aliran data tampilan dan pencetakan yang lebih umum seperti Printer Command Language (PCL) dan PDF.

   Selama jendela pemeliharaan LRS PageCenter X, VPSX/MFI LRS mempertahankan antrian output dan berfungsi sebagai cadangan untuk antrian output. LRS VPSX/MFI menghubungkan dan mengirimkan output ke LRS PageCenter X dengan menggunakan protokol. LRS/Queue LRS/Queue melakukan pertukaran kesiapan dan penyelesaian untuk pekerjaan untuk membantu memastikan bahwa transfer data terjadi.

   Catatan:

   Untuk informasi selengkapnya tentang data cetak yang diteruskan dari Rocket Software Print Exit ke LRS/Queue dan mekanisme batch mainframe yang VPSX/MFI didukung LRS, lihat Mencetak pengambilan data di bagian Informasi tambahan.

   LRS VPSX/MFI dapat melakukan pemeriksaan kesehatan di tingkat armada printer. Untuk informasi lebih lanjut, lihat Pemeriksaan kesehatan armada printer di bagian Informasi tambahan dari pola ini.

5. Solusi manajemen keluaran LRS PageCenter X digunakan di Amazon EC2, dan data operasionalnya disimpan di Amazon FSx untuk Windows File Server. LRS PageCenter X menyediakan sistem manajemen laporan pusat dari semua file yang diimpor ke LRS PageCenter X bersama dengan semua pengguna dapat mengakses file. Pengguna dapat melihat konten file tertentu atau melakukan pencarian di beberapa file untuk kriteria pencocokan.

   LRS/NetX Komponen ini adalah server aplikasi web multi-threaded yang menyediakan lingkungan runtime umum untuk aplikasi LRS PageCenter X dan aplikasi LRS lainnya. Komponen LRS/Web Connect diinstal pada server web Anda dan menyediakan konektor dari server web ke server aplikasi LRS/NetX web.
6. LRS PageCenter X menyediakan penyimpanan untuk objek sistem file. Data operasional LRS PageCenter X disimpan di Amazon FSx untuk Windows File Server.
7. Otentikasi dan otorisasi manajemen keluaran dilakukan oleh AWS Managed Microsoft AD dengan LRS/DIS.

> **Note**
>
> Solusi target biasanya tidak memerlukan perubahan aplikasi untuk mengakomodasi bahasa pemformatan mainframe, seperti IBM AFP atau Xerox LCD.

Arsitektur infrastruktur AWS

Diagram berikut menunjukkan arsitektur infrastruktur AWS yang sangat tersedia dan aman untuk beban kerja manajemen output mainframe.

Diagram menunjukkan alur kerja berikut:

1. Penjadwal batch memulai proses batch dan diterapkan di Amazon EC2 di beberapa Availability Zone for high availability (HA).

   > **Note**
   >
   > Pola ini tidak mencakup implementasi penjadwal batch. Untuk informasi selengkapnya tentang implementasi, lihat dokumentasi vendor perangkat lunak untuk penjadwal Anda.

2. Pekerjaan batch mainframe (ditulis dalam bahasa pemrograman seperti JCL atau COBOL) menggunakan logika bisnis inti untuk memproses dan menghasilkan output cetak, seperti laporan penagihan, kartu ID, dan laporan pinjaman. Pekerjaan batch diterapkan di Amazon EC2 di dua Availability Zone untuk HA. Ini menggunakan Rocket Software Print Exit API untuk merutekan output cetak ke LRS VPSX/MFI untuk preprocessing data.
3. Server VPSX/MFI cetak LRS digunakan EC2 di Amazon di dua Availability Zone untuk HA (active-standby redundant pair). Ini menggunakan Amazon EBS sebagai penyimpanan data operasional. Network Load Balancer melakukan pemeriksaan kesehatan pada instans VPSX/MFI EC2 LRS. Jika instans aktif dalam keadaan tidak sehat, penyeimbang beban merutekan lalu lintas ke instans siaga panas di Availability Zone lainnya. Permintaan cetak disimpan dalam Antrian Job LRS secara lokal di setiap instance. EC2 Jika terjadi kegagalan, instance yang gagal harus dimulai ulang sebelum layanan LRS dapat melanjutkan pemrosesan permintaan cetak.

> **Note**
>
> LRS juga VPSX/MFI dapat melakukan pemeriksaan kesehatan di tingkat armada printer. Untuk informasi lebih lanjut, lihat Pemeriksaan kesehatan armada printer di bagian Informasi tambahan dari pola ini.

4. Manajemen keluaran LRS PageCenter X diterapkan EC2 di Amazon di dua Availability Zone untuk HA (active-standby redundant pair). Ini menggunakan Amazon FSx untuk Windows File Server sebagai penyimpanan data operasional. Jika instans aktif dalam keadaan tidak sehat, penyeimbang beban melakukan pemeriksaan kesehatan pada instans LRS PageCenter X dan mengarahkan lalu lintas ke EC2 instans siaga di Availability Zone lainnya.
5. Network Load Balancer menyediakan nama DNS untuk mengintegrasikan Server LRS dengan LRS X VPSX/MFI . PageCenter

> **Note**
>
> LRS PageCenter X mendukung penyeimbang beban Layer 4.

6. LRS PageCenter X menggunakan Amazon FSx untuk Windows File Server sebagai penyimpanan data operasional yang digunakan di dua Availability Zone untuk HA. LRS PageCenter X hanya memahami file yang ada di berbagi file, bukan dalam database eksternal.
7. AWS Managed Microsoft AD digunakan LRS/DIS untuk melakukan autentikasi dan otorisasi alur kerja manajemen keluaran. Untuk informasi selengkapnya, lihat Mencetak otentikasi dan otorisasi keluaran di bagian Informasi tambahan.

Alat

Layanan AWS

- AWS Directory Service untuk Microsoft Active Directory memungkinkan beban kerja yang sadar direktori dan sumber daya AWS Anda untuk menggunakan Microsoft Active Directory di AWS Cloud.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon Elastic Compute Cloud (Amazon). EC2
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Elastic Load Balancing (ELB) mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh EC2 instans Amazon, kontainer, dan alamat IP di satu atau beberapa Availability Zone. Pola ini menggunakan Network Load Balancer.
- Amazon FSx menyediakan sistem file yang mendukung protokol konektivitas standar industri dan menawarkan ketersediaan dan replikasi tinggi di seluruh Wilayah AWS. Pola ini menggunakan Amazon FSx untuk Windows File Server.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.

Alat lainnya

- Perangkat lunak LRS PageCenter X menyediakan dokumen yang dapat diskalakan dan solusi manajemen konten laporan yang membantu pengguna memperoleh nilai maksimum dari informasi melalui pengindeksan otomatis, enkripsi, dan fitur pencarian lanjutan.
- LRS VPSX/MFI (Micro Focus Interface), dikembangkan bersama oleh LRS dan Rocket Software, menangkap output dari spool JES Perangkat Lunak Rocket dan dengan andal mengirimkannya ke tujuan cetak tertentu.
- LRS/Queue is a transmission agent that’s TCP/IP based. LRS VPSX/MFI uses LRS/Queueuntuk mengumpulkan atau menangkap data cetak melalui antarmuka pemrograman Rocket Software JES Print Exit.
- LRS Directory Integration Server (LRS/DIS) digunakan untuk otentikasi dan otorisasi selama alur kerja cetak.

- Rocket Enterprise Server adalah lingkungan penyebaran aplikasi untuk aplikasi mainframe. Ini menyediakan lingkungan runtime untuk aplikasi mainframe yang dimigrasikan atau dibuat dengan menggunakan versi Pengembang Rocket Enterprise apa pun.

Epik

Siapkan runtime Rocket dan gunakan aplikasi batch mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan runtime dan terapkan aplikasi demo.	Untuk menyiapkan Rocket Enterprise Server di Amazon EC2 dan menerapkan aplikasi BankDemo demonstrasi Perangkat Lunak Rocket, ikuti petunjuk dalam panduan pengguna Modernisasi AWS Mainframe. BankDemo Aplikasi ini adalah aplikasi batch mainframe yang membuat dan kemudian memulai output cetak.	Arsitek awan

Siapkan server cetak LRS di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance Amazon EC2 Windows.	Untuk meluncurkan instans Amazon EC2 Windows, ikuti petunjuk di Luncurkan EC2 instans Amazon di EC2 dokumentasi Amazon. Gunakan nama host yang sama yang Anda gunakan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk lisensi produk LRS Anda. Instans Anda harus memenuhi persyaratan perangkat keras dan perangkat lunak berikut untuk LRS VPSX/LKM: • CPU - Inti Ganda • RAM — 16 GB • Penggerak - 500 GB • EC2 Contoh minimum - `m5.xlarge` • OS — Windows • Perangkat Lunak — Layanan Informasi Internet (IIS) atau Apache **Note** Persyaratan perangkat keras dan perangkat lunak sebelumnya ditujukan untuk armada printer kecil (sekitar 500-1000). Untuk mendapatkan persyaratan lengkap, konsultasikan dengan kontak LRS dan AWS Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Saat Anda membuat instance Windows, konfirmasikan bahwa EC2 nama host adalah nama host yang sama yang digunakan untuk lisensi produk LRS. 2. Connect ke EC2 instans Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 3. Pada menu Start Windows, temukan dan buka Server Manager. 4. Di Server Manager, pilih Dasbor, Mulai Cepat, Tambahkan peran dan fitur, lalu pilih Peran server. 5. Dalam peran Server, pilih WebServer (IIS), dan kemudian pilih Pengembangan Aplikasi. 6. Dalam Pengembangan Aplikasi, pilih kotak centang CGI. 7. Untuk menginstal CGI, ikuti petunjuk di Windows Server Manger Add role and features wizard. 8. Buka port 5500 di firewall Windows EC2 instance	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk LRS/Queue komunikasi.	
Instal LRS VPSX/MFI pada EC2 instance.	1. Connect ke EC2 instans Anda. 2. Buka tautan ke halaman unduhan produk dari pesan email LRS yang seharusnya Anda terima. **Note** Produk LRS didistribusikan melalui transfer file elektronik (EFT). 3. Unduh LRS VPSX/MFI, dan unzip file (folder default:). `c:\LRS` 4. Untuk menginstal LRS VPSX/MFI, luncurkan Pemasang Produk LRS dari folder yang tidak di-zip. 5. Pada menu Select Features, pilih VPSX® Server, lalu pilih Next untuk memulai proses instalasi. Anda akan menerima pesan sukses ketika instalasi selesai.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS/antrian.	1. Connect ke EC2 instance Rocket Enterprise Server Anda. 2. Buka tautan ke halaman unduhan produk LRS dari pesan email LRS yang seharusnya Anda terima, unduh LRS/antrian, lalu unzip file. 3. Arahkan ke lokasi tempat Anda mengunduh file, lalu luncurkan Pemasang Produk LRS untuk menginstal LRS/antrian. 4. Ikuti petunjuk di Pemasang Produk LRS untuk menyelesaikan proses instalasi.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS/DIS.	LRS/DIS Produk ini sering disertakan dalam instalasi LRS VPSX. Namun, jika LRS/DIS tidak diinstal bersama dengan LRS VPSX, gunakan langkah-langkah berikut untuk menginstalnya: 1. Connect ke VPSX/MFI EC2 instans LRS Anda. 2. Buka tautan ke halaman unduhan produk LRS dari pesan email LRS yang seharusnya Anda terima, unduh LRS/DIS, lalu unzip file. 3. Arahkan ke lokasi tempat Anda mengunduh file, lalu luncurkan Pemasang Produk LRS. 4. Di LRS Product Installer, perluas LRS Misc Tools, pilih LRS DIS, lalu pilih Next. 5. Ikuti petunjuk lainnya di Pemasang Produk LRS untuk menyelesaikan proses instalasi.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup target.	Buat grup target dengan mengikuti petunjuk di Buat grup target untuk Network Load Balancer Anda. Saat Anda membuat grup target, daftarkan VPSX/MFI EC2 instance LRS sebagai target: 1. Pada halaman Tentukan detail grup, untuk Pilih Jenis Target, pilih Contoh. 2. Untuk Protokol, pilih TCP. 3. Untuk Port, pilih 5500. 4. Pada halaman Daftar target, di bagian Instance yang tersedia, pilih instans LRS VPSX/MFI EC2 .	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Network Load Balancer.	Untuk membuat Network Load Balancer, ikuti instruksi dalam dokumentasi Elastic Load Balancing. Network Load Balancer Anda merutekan lalu lintas dari Rocket Enterprise Server ke instans LRS VPSX/MFI EC2 . Saat Anda membuat Network Load Balancer, pilih nilai berikut pada halaman Listeners dan Routing: 1. Untuk Protokol, pilih TCP. 2. Untuk Port, pilih 5500. 3. Untuk tindakan Default, pilih Teruskan ke grup target yang Anda buat sebelumnya.	Arsitek awan

Integrasikan Rocket Enterprise Server dengan LRS/Queue dan LRS VPSX/MFI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Rocket Enterprise Server untuk LRS/Queue integrasi.	1. Connect ke EC2 instance Rocket Enterprise Server Anda dengan mengikuti petunjuk dalam EC2 dokumentasi Amazon. 2. Pada menu Start Windows, buka UI Administrasi Server Perusahaan Rocket.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di bilah menu, pilih NATIVE. 4. Di panel navigasi, pilih Server Direktori, lalu pilih BANKDEMO untuk wilayah Server Perusahaan Anda. 5. Dari Umum di panel navigasi kiri, gulir ke bawah ke bagian Tambahan untuk mengonfigurasi variabel lingkungan (`LRSQ_ADDRESS` ,`LRSQ_PORT` ,`LRSQ_COMMAND`) untuk menunjuk ke LRSQ. • Untuk LRSQ_ADDRESS, masukkan alamat IP atau nama DNS Network Load Balancer yang Anda buat sebelumnya. • Untuk LRSQ_PORT, masukkan VPSX LRSQ Listener Port (5500). • Untuk LRSQ_COMMAND, masukkan lokasi jalur LRSQ yang dapat dieksekusi. **Note** LRS saat ini mendukung batas karakter maksimum 50 untuk nama DNS. Jika nama	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	DNS Anda lebih dari 50 karakter, Anda dapat menggunakan alamat IP Network Load Balancer sebagai alternatif.	
Konfigurasikan Rocket Enterprise Server untuk VPSX/MFI integrasi LRS.	1. Salin `VPSX_MFI_R2` folder dari VPSX/MFI installer LRS ke lokasi Rocket Enterprise Server di. `C\BANKDEMO\print` 2. Connect ke EC2 instance Rocket Enterprise Server Anda dengan mengikuti petunjuk dalam EC2 dokumentasi Amazon. 3. Pada menu Start Windows, buka UI Administrasi Server Perusahaan Rocket. 4. Pada bilah menu, pilih NATIVE. 5. Di panel navigasi, pilih Directory Server, dan kemudian pilih BANKDEMO. 6. Di bawah BANKDEMO, pilih JES. 7. Di bawah Jalur Program JES, tambahkan `DLL(VPSX_MFI_R2)` jalur dari`C\BANKDEMO\print` .	Arsitek awan

Mengatur antrian cetak dan pengguna cetak

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kaitkan modul Keluar Cetak Perangkat Lunak Roket dengan Proses Eksekusi Server printer batch Rocket Enterprise Server.	1. Connect ke EC2 instance Rocket Enterprise Server Anda dengan mengikuti petunjuk dalam EC2 dokumentasi Amazon. 2. Pada menu Start Windows, buka UI Administrasi Server Perusahaan Rocket Focus. 3. Pada bilah menu, pilih NATIVE. 4. Di panel navigasi, pilih Directory Server, dan kemudian pilih BANKDEMO. 5. Di bawah BANKDEMO, pilih JES, dan gulir ke bawah ke Printer. 6. Di Printer, kaitkan modul Keluar Cetak Perangkat Lunak Roket (LRSPRTE6 untuk Batch) ke printer batch Rocket Enterprise Server Proses Eksekusi Server (SEP). Ini memungkinkan perutean keluaran cetak ke LRS VPSX/MFI. Untuk informasi selengkap nya tentang konfigurasi, lihat Menggunakan Keluar	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di dokumentasi Fokus OpenText Mikro.	
Buat antrian keluaran cetak di LRS VPSX/MFI dan integrasikan dengan LRS X. PageCenter	1. Connect ke VPSX/MFI EC2 instans LRS Anda. 2. Pada menu Mulai Windows, buka Antarmuka Web VPSX. 3. Di panel navigasi, pilih Printer. 4. Pilih Tambah, lalu pilih Tambah Printer. 5. Pada halaman Konfigurasi Printer, untuk Nama Printer, masukkan Lokal. 6. Untuk ID VPSX, masukkan. VPS1 7. Untuk CommType, pilih TCPIP/LRSQ. 8. Untuk Alamat Host/IP, masukkan alamat IP Network Load Balancer di depan instance LRS X. PageCenter EC2 9. Untuk port Remote, masukkan 5800. 10. Untuk antrian jarak jauh, masukkan nama folder dokumen LRS PageCente r X tempat output akan disimpan. 11. Pilih Tambahkan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna cetak di LRS VPSX/LKM.	1. Connect ke VPSX/MFI EC2 instans LRS Anda. 2. Pada menu Mulai Windows, buka Antarmuka Web VPSX. 3. Di panel navigasi, pilih Keamanan, lalu pilih Pengguna. 4. Di kolom Nama Pengguna, pilih admin, lalu pilih Salin. 5. Di jendela Pemelihar aan Profil Pengguna, untuk Nama Pengguna, masukkan nama pengguna (misalnya, PrintUser). 6. Untuk Deskripsi, masukkan deskripsi singkat (misalnya, Pengguna untuk tes cetak). 7. Pilih Perbarui. Ini menciptak an pengguna cetak (misalnya, PrintUser). 8. Di panel navigasi, di bawah Pengguna, pilih pengguna baru yang Anda buat. 9. Pada menu Command, pilih Security. 10. Pada halaman Aturan Keamanan, pilih semua opsi keamanan printer dan keamanan pekerjaan yang berlaku, lalu pilih Simpan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	11. Untuk menambahkan pengguna cetak baru Anda ke grup Administrator, pada panel navigasi, pilih Keamanan, lalu pilih Konfigurasi. 12. Di jendela konfigurasi Keamanan, tambahkan pengguna cetak baru Anda ke kolom Administrator.	

Siapkan server LRS PageCenter X di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance Amazon EC2 Windows.	Luncurkan instans Amazon EC2 Windows dengan mengikuti petunjuk dari Langkah 1: Luncurkan instance di EC2 dokumentasi Amazon. Gunakan nama host yang sama yang Anda gunakan untuk lisensi produk LRS Anda. Instans Anda harus memenuhi persyaratan perangkat keras dan perangkat lunak berikut untuk LRS PageCenter X: • CPU - Inti Ganda • RAM — 16 GB • Penggerak - 500 GB	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• EC2 Contoh minimum - `m5.xlarge` • OS — Windows • Perangkat lunak —IIS atau Apache **Note** Persyaratan perangkat keras dan perangkat lunak sebelumnya a ditujukan untuk armada printer kecil (sekitar 500-1000). Untuk mendapatkan persyaratan lengkap, konsultasikan dengan kontak LRS dan AWS Anda. 1. Saat Anda membuat instance Windows, konfirmasikan bahwa EC2 nama host adalah nama host yang sama yang digunakan untuk lisensi produk LRS. 2. Connect ke EC2 instans Anda dengan mengikuti petunjuk dalam EC2 dokumentasi Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pada menu Start Windows, temukan dan buka Server Manager. 4. Di Server Manager, pilih Dasbor, Mulai Cepat, Tambahkan peran dan fitur, lalu pilih Peran server. 5. Dalam peran Server, pilih WebServer (IIS), dan kemudian pilih Pengembangan Aplikasi. 6. Dalam Pengembangan Aplikasi, pilih kotak centang CGI. 7. Untuk menginstal CGI, ikuti petunjuk di Windows Server Manger Add role and features wizard. 8. Buka port 5800 untuk TCP/IP lalu lintas masuk di firewall Windows instance. EC2 LRS VPSX menggunakan TCPIP/LRSQ protokol pada port 5800 untuk berkomunikasi dengan LRS X. PageCenter	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS PageCenter X pada EC2 instance.	1. Connect ke EC2 instans Anda. 2. Buka tautan ke halaman unduhan produk dari pesan email LRS yang seharusnya Anda terima. **Note** Produk LRS didistribusikan melalui transfer file elektronik (EFT). 3. Unduh LRS PageCente r X, dan unzip file (folder default:`c:\LRS`). 4. Untuk menginstal LRS PageCenter X, luncurkan Pemasang Produk LRS dari folder yang tidak di-zip. 5. Pada menu Select Features, pilih PageCente rX, lalu pilih Next untuk memulai proses instalasi. Anda akan menerima pesan sukses ketika instalasi selesai.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS/DIS.	LRS/DIS Produk ini sering disertakan dalam instalasi LRS VPSX. Namun, jika LRS/DIS tidak diinstal bersama dengan LRS VPSX, gunakan langkah-langkah berikut untuk menginstalnya: 1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Buka tautan ke halaman unduhan produk LRS dari email LRS yang seharusnya Anda terima, unduh LRS/DIS, lalu unzip file. 3. Arahkan ke lokasi tempat Anda mengunduh file, lalu luncurkan Pemasang Produk LRS. 4. Di LRS Product Installer, perluas LRS Misc Tools, pilih LRS DIS, lalu pilih Next. 5. Ikuti petunjuk lainnya di Pemasang Produk LRS untuk menyelesaikan proses instalasi.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup target.	Buat grup target dengan mengikuti petunjuk di Buat grup target untuk Network Load Balancer Anda. Saat Anda membuat grup target, daftarkan EC2 instance LRS PageCenter X sebagai target: 1. Pada halaman Tentukan detail grup, untuk Pilih Jenis Target, pilih Contoh. 2. Untuk Protokol, pilih TCP. 3. Untuk Port, pilih 5800. 4. Pada halaman Daftar target, di bagian Instance yang tersedia, pilih instance LRS PageCenter X EC2 .	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Network Load Balancer.	Untuk membuat Network Load Balancer, ikuti instruksi dalam dokumentasi Elastic Load Balancing. Network Load Balancer Anda merutekan lalu lintas dari LRS VPSX/MFI ke instans LRS X. PageCenter EC2 Saat Anda membuat Network Load Balancer, pilih nilai berikut pada halaman Listeners dan Routing: 1. Untuk Protokol, pilih TCP. 2. Untuk Port, pilih 5800. 3. Untuk tindakan Default, pilih Teruskan ke grup target yang Anda buat sebelumnya.	Arsitek awan

Siapkan fitur manajemen keluaran di LRS X PageCenter

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan fungsi Impor di LRS X PageCenter.	Anda dapat menggunakan fungsi Impor LRS PageCenter X untuk mengenali output yang mendarat di LRS PageCenter X dengan kriteria seperti nama Job atau ID Formulir. Anda kemudian dapat merutekan output ke	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	folder tertentu di PageCenter LRS X. Untuk mengaktifkan fungsi Impor, lakukan hal berikut: 1. Connect ke EC2 instans LRS PageCenter X Anda dengan mengikuti petunjuk dalam EC2 dokumentasi Amazon. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin. 4. Pada halaman Konfigura si, pilih Advanced, Impor parameter. 5. Di bagian Impor parameter , pilih kotak centang Impor Lanjutan. 6. Untuk melakukan perubahan, pilih Perbarui.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kebijakan penyimpanan dokumen.	LRS PageCenter X menggunakan kebijakan penyimpanan dokumen untuk memutuskan berapa lama menyimpan dokumen di PageCenter LRS X. Untuk mengonfigurasi kebijakan penyimpanan dokumen, lakukan hal berikut: 1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin. 4. Pada halaman Admin, pilih List/General Admin Grup Arsip, lalu pilih Kebijakan penyimpanan. 5. Di bagian Kebijakan penyimpanan, pilih Tambah untuk membuat kebijakan penyimpanan. 6. Pada halaman Informasi Kebijakan Penyimpanan, masukkan nama kebijakan penyimpanan, deskripsi, dan periode penyimpanan dokumen. 7. Untuk menyimpan perubahan dan membuat kebijakan, pilih Ok.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan untuk merutekan dokumen output ke folder tertentu di LRS X PageCenter.	Di LRS PageCenter X, Tujuan menentukan jalur folder di mana output akan dikirim ketika tujuan ini dipanggil oleh Definisi Laporan. Untuk contoh ini, buat folder berdasark an folder ID Formulir dalam definisi laporan, dan simpan output ke folder itu. 1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin, Advance Import, Destination. 4. Di bagian Tujuan, pilih Tambah untuk membuka formulir Pemeliharaan Tujuan. 5. Pada formulir Destination Maintenance, masukkan nilai-nilai berikut: • Nama tujuan - Formulir • Deskripsi - Deskripsi tujuan, seperti struktur folder berbasis Form • Jenis tujuan - Folder • Parameter Folder - Impor jalur folder (jalur folder yang akan dibuat di PageCente	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	r X saat dokumen tiba; misalnya, jalur **`/Test/&FORM/&IMPORTDATE/&IMPORTTIME`** akan membuat **`Test`** folder dasar, subfolder berdasarkan nama Form-Id, subfolder berdasarkan tanggal Impor **`STD`**, dan kemudian subfolder berdasarkan waktu Impor) • Nama dokumen — Nama dinamis yang ditetapkan ke dokumen saat disimpan di folder. 6. Di daftar tarik-turun, pilih kebijakan penyimpanan. Misalnya, pilih Tahun1 untuk menyimpan dokumen selama 1 tahun. 7. Untuk menyimpan perubahan, pilih Ok.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat definisi laporan.	1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin, Advance Import, Report Definition, dan kemudian pilih Add. 4. Pada halaman Pemelihar aan definisi laporan, pada tab Umum, masukkan nama Definisi Laporan. 5. Pada tab Umum, di bawah Bidang, Anda dapat menentukan kriteria pemilihan seperti Nama Pekerjaan, Formulir, Kelas, dan Penulis. Misalnya, Anda bisa memasukkan Nama Job dari MFIDEMO. Nilai Job Name akan menjadi nama pekerjaan batch yang akan menghasil kan output cetak. 6. Pada tab Tujuan, di bawah Tujuan yang Tersedia, pilih tujuan yang dibuat sebelumnya (Formulir). 7. Pilih Tambah untuk menambahkan tujuan Formulir sebagai tujuan yang Ditugaskan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Contoh ini mencakup definisi laporan di mana output yang dihasilkan oleh MFIDEMO dan dirutekan ke LRS PageCenter X disimpan dalam struktur folder yang ditentukan dalam definisi tujuan.	

Mengatur otentikasi dan otorisasi untuk manajemen output

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat domain AWS Managed Microsoft AD dengan pengguna dan grup.	1. Untuk membuat direktori di AWS Managed Microsoft AD, ikuti petunjuk di Membuat direktori AWS Managed Microsoft AD Anda. 2. Untuk menerapkan EC2 instans (Active Directory manager) dan menginstal alat Active Directory untuk mengelola AWS Managed Microsoft AD Anda, ikuti petunjuk di Langkah 3: Menerapkan EC2 instans	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk mengelola AWS Managed Microsoft AD Anda. 3. Untuk menyambung ke EC2 instans Anda, ikuti petunjuk dalam EC2 dokumentasi Amazon. **Note** Saat Anda terhubung ke EC2 instance, di jendela Windows Security, masukkan kredensi administrator untuk direktori yang Anda buat di langkah 1. 4. Pada menu Start Windows, di bawah Windows Administrative Tools, pilih Active Directory Users and Computers. 5. Untuk membuat pengguna cetak di domain Active Directory, ikuti petunjuk di Buat pengguna.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bergabunglah dengan EC2 instans ke domain AWS Managed Microsoft AD.	Bergabunglah dengan EC2 instans LRS VPSX/MFI dan LRS PageCenter X ke domain AWS Managed Microsoft AD Anda secara otomatis (dokumentasi AWS Knowledge Center) atau secara manual (dokumentasi AWS Directory Service).	Arsitek awan
Konfigurasikan dan integrasikan LRS/DIS dengan AWS Managed Microsoft AD untuk EC2 instans LRS PageCenter X.	1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin. 4. Pada halaman Konfigurasi, di bagian Parameter Keamanan, untuk Jenis Keamanan, pilih LRS/DIS. 5. Masukkan preferensi Anda untuk sisa opsi di bagian Parameter Keamanan. 6. Pada menu Start Windows, buka folder PageCenterX, pilih Server Start, dan kemudian pilih Server Stop. 7. Masuk ke LRS PageCenter X dengan nama pengguna dan kata sandi Direktori Aktif Anda.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan grup Impor untuk mengimpor output dari LRS VPSX ke LRS X. PageCenter	1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin, Admin keamanan, Grup. 4. Di bagian Grup, pilih Tambah untuk membuka formulir Preferensi grup. 5. Dalam formulir preferensi Grup, masukkan nilai untuk nama Grup dan Deskripsi. 6. Perluas opsi Umum, lalu pilih kotak centang Impor. 7. Untuk menyimpan perubahan, pilih Ok.	Arsitek awan
Tambahkan aturan keamanan ke grup Impor.	1. Buka menu konteks (klik kanan) untuk grup Impor. 2. Pilih Advance, lalu pilih Security. 3. Di bagian Keamanan, pilih Impor, dan pilih kotak centang Subfolder. 4. Untuk menyimpan perubahan, pilih Terapkan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna di LRS PageCenter X untuk melakukan impor output dari LRS VPSX/MFI.	Saat Anda membuat pengguna di LRS PageCenter X untuk melakukan impor output, nama pengguna harus sama dengan ID VPSX dari antrian keluaran cetak di LRS VPSX/MFI. Dalam contoh ini, ID VPSX adalah. VPS1 1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin, Admin keamanan, Pengguna. 4. Pilih Tambah untuk membuka formulir pemeliharaan profil Pengguna. 5. Dalam pemeliharaan profil Pengguna, untuk Nama pengguna, masukkan VPS1.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan pengguna LRS PageCenter X Impor ke grup Impor saja.	Untuk memberikan izin yang diperlukan untuk impor dokumen dari LRS VPSX ke LRS PageCenter X, lakukan hal berikut: 1. Connect ke EC2 instans LRS PageCenter X Anda. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin, Admin keamanan, Grup. 4. Di bagian Grup, buka menu konteks (klik kanan) untuk grup Impor saja, lalu pilih Advance, Security. 5. Pada halaman Folder Security Records (ImportOnly), pilih tab Pengguna. 6. Pada tab Pengguna, di bawah Nama, pilih pengguna VPS1dari daftar dropdown, dan pilih Terapkan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan LRS/DIS dengan AWS Managed Microsoft AD untuk VPSX/MFI EC2 instans LRS.	1. Connect ke VPSX/MFI EC2 instans LRS Anda. 2. Pada menu Mulai Windows, buka Antarmuka Web VPSX. 3. Di panel navigasi, pilih Keamanan, lalu pilih Konfigurasi. 4. Pada halaman Konfigura si Keamanan, di bagian Parameter Keamanan, untuk Jenis Keamanan, pilih LRS/DIS (Eksternal). 5. Masukkan preferensi Anda untuk sisa opsi di bagian Parameter Keamanan. 6. Pada menu Start Windows, buka folder LRS Output Management, pilih Server Start, dan kemudian pilih Server Stop. 7. Masuk ke LRS VPSX/MFI dengan nama pengguna dan kata sandi Direktori Aktif Anda.	Arsitek awan

Konfigurasikan Amazon FSx untuk Windows File Server sebagai penyimpanan data operasional untuk LRS X PageCenter

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sistem file untuk LRS X PageCenter.	Untuk menggunakan Amazon FSx untuk Windows File	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Server sebagai penyimpanan data operasional untuk LRS PageCenter X di lingkunga n Multi-AZ, ikuti petunjuk di Langkah 1: Buat sistem file Anda.	
Petakan berbagi file ke EC2 instance LRS PageCenter X.	Untuk memetakan file share yang dibuat pada langkah sebelumnya ke EC2 instance LRS PageCenter X, ikuti petunjuk di Langkah 2: Petakan berbagi file Anda ke EC2 instance yang menjalank an Windows Server.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Petakan Direktori Kontrol LRS PageCenter X dan Direktori Folder Master ke drive berbagi FSx jaringan Amazon.	1. Connect ke EC2 instans LRS PageCenter X Anda dengan mengikuti petunjuk dalam EC2 dokumentasi Amazon. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di Folder Explorer, pilih Admin, Konfigurasi. 4. Pada halaman Konfigura si, pilih Direktori, lalu pilih Direktori Kontrol. 5. Di Direktori Kontrol, masukkan`\\FSx file share DNS name\shar e\cntl` . 6. Di Direktori Folder Master, masukkan`\\FSx file share DNS name\shar e\mstr` .	Arsitek awan

Uji alur kerja manajemen keluaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai permintaan cetak batch dari BankDemo aplikasi Rocket Software.	1. Buka emulator 3270-term inal di instance Rocket Enterprise Server Anda. EC2 2. Connect ke BankDemo aplikasi dengan menjalank an perintah`connect 127.0.0.1:9278` .	Insinyur uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pada antarmuka baris perintah, untuk User id, masukkan BankDemo, untuk Kata Sandi, masukkan B0001. Untuk kunci yang tidak kosong. 4. Untuk opsi Permintaan pernyataan tercetak, masukkan X pada baris kosong. 5. Di bagian Kirim pernyataan demi, untuk Mail, masukkan Y, lalu tekan F10.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa output cetak di LRS X PageCenter.	1. Connect ke EC2 instans LRS PageCenter X Anda dengan mengikuti petunjuk dalam EC2 dokumentasi Amazon. 2. Pada menu Start Windows, buka PCX Web Interface. 3. Di panel navigasi, buka folder Uji, buka folder STD, lalu buka folder dengan tanggal kerja, seperti `08-03-2023` (MM-DD-YYYY). **Note** Ini adalah struktur folder yang sama seperti yang didefinisikan dalam cerita Buat aturan untuk merutekan dokumen keluaran ke folder tertentu di LRS PageCenter X. 4. Buka file `formtest-STD.txt` . Anda sekarang dapat melihat hasil cetak dari laporan akun dengan kolom untuk Akun No. , Deskripsi , Tanggal, Jumlah, dan Saldo. Sebagai contoh, lihat	Insinyur uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`batch_print_output` lampiran untuk pola ini.	

Sumber daya terkait

- LRS
- Aliran data Presentasi Fungsi Lanjutan (dokumentasi IBM)
- Aliran Data Terkondisi Jalur (LCD) (Dokumentasi Compart)
- Memberdayakan Beban Kerja Mainframe Perusahaan di AWS dengan Micro Focus (posting blog)
- Modernisasi beban kerja pencetakan online mainframe Anda di AWS (AWS Prescriptive Guidance)
- Modernisasi beban kerja pencetakan batch mainframe Anda di AWS (AWS Prescriptive Guidance)

Informasi tambahan

Pertimbangan-pertimbangan

Selama perjalanan modernisasi Anda, Anda dapat mempertimbangkan berbagai konfigurasi untuk batch mainframe dan proses online dan output yang dihasilkannya. Platform mainframe telah disesuaikan oleh setiap pelanggan dan vendor yang menggunakannya dengan persyaratan khusus yang secara langsung memengaruhi pencetakan. Misalnya, platform Anda saat ini mungkin menggabungkan aliran data IBM AFP atau Xerox LCD ke dalam alur kerja saat ini. Selain itu, karakter kontrol carriage mainframe dan kata-kata perintah saluran dapat memengaruhi tampilan halaman yang dicetak dan mungkin memerlukan penanganan khusus. Sebagai bagian dari proses perencanaan modernisasi, kami menyarankan Anda menilai dan memahami konfigurasi di lingkungan cetak spesifik Anda.

Cetak pengambilan data

Rocket Software Print Exit meneruskan informasi yang diperlukan untuk LRS VPSX/MFI untuk memproses file spool secara efektif. Informasi terdiri dari bidang yang dilewatkan di blok kontrol yang relevan, seperti berikut ini:

- NAMA KERJA
- PEMILIK (USERID)

- TUJUAN
- BENTUK
- NAMA BERKAS
- PENULIS

LRS VPSX/MFI mendukung mekanisme batch mainframe berikut untuk menangkap data dari Rocket Enterprise Server:

- print/spool Pemrosesan BATCH COBOL menggunakan pernyataan z/OS JCL DD/OUTPUT SYSOUT standar.
- print/spool Pemrosesan BATCH COBOL menggunakan pernyataan z/OS JCL CA-SPOOL SUBSYS DD standar.
- IMS/COBOL print/spoolpemrosesan menggunakan antarmuka CBLTDLI. Untuk daftar lengkap metode dan contoh pemrograman yang didukung, lihat dokumentasi LRS yang disertakan dengan lisensi produk Anda.

Pemeriksaan kesehatan armada printer

LRS VPSX/MFI (LRS LoadX) dapat melakukan pemeriksaan kesehatan deep dive, termasuk manajemen perangkat dan optimalisasi operasional. Manajemen perangkat dapat mendeteksi kegagalan pada perangkat printer dan merutekan permintaan cetak ke printer yang sehat. Untuk informasi selengkapnya tentang pemeriksaan kesehatan mendalam untuk armada printer, lihat dokumentasi LRS yang disertakan dengan lisensi produk Anda.

Cetak otentikasi dan otorisasi

LRS/DIS enables LRS applications to authenticate user IDs and passwords by using Microsoft Active Directory or a Lightweight Directory Access Protocol (LDAP) server. In addition to basic print authorization, LRS/DISjuga dapat menerapkan kontrol keamanan cetak tingkat butiran dalam kasus penggunaan berikut:

- Kelola siapa yang dapat menelusuri pekerjaan printer.
- Kelola tingkat penelusuran pekerjaan pengguna lain.
- Kelola tugas operasional—misalnya, keamanan tingkat perintah seperti menahan atau melepaskan, membersihkan, memodifikasi, menyalin, dan mengubah rute. Keamanan dapat diatur oleh User-ID atau grup, mirip dengan grup keamanan Active Directory atau grup LDAP.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Modernisasi beban kerja pencetakan batch mainframe dengan menggunakan Rocket Enterprise Server dan LRS AWS VPSX/LFI

Shubham Roy dan Kevin Yung, Amazon Web Services

Abraham Rondon, Fokus Mikro

Guy Tucker, Levi, Ray dan Shoup Inc

Ringkasan

Pola ini menunjukkan kepada Anda cara memodernisasi beban kerja pencetakan batch mainframe penting bisnis Anda di Amazon Web Services (AWS) Cloud dengan menggunakan Rocket Enterprise Server sebagai runtime untuk aplikasi mainframe modern dan VPSX/MFI LRS (Micro Focus Interface) sebagai server cetak. Pola ini didasarkan pada pendekatan modernisasi mainframe replatform. Dalam pendekatan ini, Anda memigrasikan pekerjaan batch mainframe Anda ke Amazon Elastic Compute Cloud (Amazon EC2) dan memigrasikan database mainframe Anda, seperti IBM for DB2) terintegrasi dengan AWS z/OS, to Amazon Relational Database Service (Amazon RDS). The authentication and authorization for the modernized print workflow is performed by AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD. The LRS Directory Information Server (LRS/DIS Managed Microsoft AD. Dengan memodernisasi beban kerja pencetakan batch Anda, Anda dapat mengurangi biaya infrastruktur TI, mengurangi utang teknis pemeliharaan sistem lama, menghapus silo data, meningkatkan kelincahan dan efisiensi dengan DevOps model, dan memanfaatkan sumber daya sesuai permintaan dan otomatisasi di AWS Cloud.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Beban kerja pencetakan mainframe atau manajemen output
- Pengetahuan dasar tentang cara membangun kembali dan mengirimkan aplikasi mainframe yang berjalan di Rocket Enterprise Server (Untuk informasi lebih lanjut, lihat lembar data Rocket Enterprise Server dalam dokumentasi Rocket.)
- Pengetahuan dasar tentang solusi dan konsep pencetakan cloud LRS
- Perangkat lunak dan lisensi Rocket Enterprise Server (Untuk informasi lebih lanjut, hubungi penjualan Rocket.)

- VPSX/MFI, LRS/Queue, and LRS/DISPerangkat lunak dan lisensi LRS (Untuk informasi lebih lanjut, hubungi penjualan LRS.)

> **Note**
>
> Untuk informasi selengkapnya tentang pertimbangan konfigurasi untuk beban kerja pencetakan batch mainframe, lihat Pertimbangan di bagian Informasi tambahan dari pola ini.

Versi produk

- Rocket Enterprise Server 6.0 (pembaruan produk 7)
- LRS VPSX/MFI V1R3 atau lebih tinggi

Arsitektur

Tumpukan teknologi sumber

- Sistem operasi — IBM z/OS
- Bahasa pemrograman — Common Business-Oriented Language (COBOL), job control language (JCL), dan Customer Information Control System (CICS)
- Database — IBM DB2 untuk z/OS dan Metode Akses Penyimpanan Virtual (VSAM)
- Keamanan — Fasilitas Kontrol Akses Sumber Daya (RACF), CA Top Secret untuk z/OS, dan Fasilitas Kontrol Akses 2 () ACF2
- Manajemen pencetakan dan output - Produk z/OS pencetakan mainframe IBM (IBM Tivoli Output Manager untuk z/OS, LRS, dan CA View)

Tumpukan teknologi target

- Sistem operasi - Microsoft Windows Server berjalan di Amazon EC2
- Hitung - Amazon EC2
- Bahasa pemrograman — COBOL, JCL, dan CICS
- Basis Data - Amazon RDS
- Keamanan — AWS Managed Microsoft AD
- Manajemen pencetakan dan output — Solusi pencetakan LRS di AWS

- Lingkungan runtime mainframe - Rocket Enterprise Server

Arsitektur sumber

Diagram berikut menunjukkan arsitektur keadaan saat ini yang khas untuk beban kerja pencetakan batch mainframe:

Diagram menunjukkan alur kerja berikut:

1. Pengguna melakukan transaksi bisnis pada sistem keterlibatan (SoE) yang dibangun di atas aplikasi IBM CICS yang ditulis dalam COBOL.
2. SoE memanggil layanan mainframe, yang mencatat data transaksi bisnis dalam database system-of-records (SoR) seperti IBM DB2 untuk z/OS.
3. SoR mempertahankan data bisnis dari BUMN.
4. Penjadwal pekerjaan batch memulai pekerjaan batch untuk menghasilkan output cetak.
5. Pekerjaan batch mengekstrak data dari database, memformat data sesuai dengan kebutuhan bisnis, dan kemudian menghasilkan output bisnis seperti laporan penagihan, kartu identitas, atau laporan pinjaman. Akhirnya, pekerjaan batch merutekan output ke manajemen keluaran pencetakan untuk pemrosesan dan pengiriman output, berdasarkan persyaratan bisnis.
6. Manajemen keluaran pencetakan menerima output cetak dari pekerjaan batch, dan kemudian mengirimkan output itu ke tujuan tertentu, seperti email, berbagi file yang menggunakan FTP aman, printer fisik yang menggunakan solusi pencetakan LRS (seperti yang ditunjukkan dalam pola ini), atau IBM Tivoli.

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk beban kerja pencetakan batch mainframe yang diterapkan di AWS Cloud:

Diagram menunjukkan alur kerja berikut:

1. Penjadwal pekerjaan batch memulai pekerjaan batch untuk membuat hasil cetak, seperti laporan penagihan, kartu identitas, atau laporan pinjaman.
2. Pekerjaan batch mainframe (direplatformed ke Amazon EC2) menggunakan runtime Rocket Enterprise Server untuk mengekstrak data dari database aplikasi, menerapkan logika bisnis

ke data, memformat data, dan kemudian mengirim data ke tujuan cetak dengan menggunakan Rocket Software Print Exit (dokumentasi Micro Focus).

3. Database aplikasi (SoR yang berjalan di Amazon RDS) menyimpan data untuk output cetak.
4. Solusi VPSX/MFI pencetakan LRS digunakan di Amazon EC2 dan data operasionalnya disimpan di Amazon Elastic Block Store (Amazon EBS). LRS VPSX/MFI menggunakan agen TCP/IP-based LRS/Queue transmisi untuk mengumpulkan data cetak melalui Rocket Software JES Print Exit API dan mengirimkan data ke tujuan printer tertentu.

> **Note**
>
> Solusi target biasanya tidak memerlukan perubahan aplikasi untuk mengakomodasi bahasa pemformatan mainframe, seperti IBM Advanced Function Presentation (AFP) atau Xerox Line Condition Data Stream (LCDS). Untuk informasi selengkapnya tentang penggunaan Perangkat Lunak Rocket untuk migrasi dan modernisasi aplikasi mainframe di AWS, lihat Memberdayakan Beban Kerja Mainframe Perusahaan di AWS dengan posting blog Micro Focus.

Arsitektur infrastruktur AWS

Diagram berikut menunjukkan arsitektur infrastruktur AWS yang sangat tersedia dan aman untuk beban kerja pencetakan batch mainframe:

Diagram menunjukkan alur kerja berikut:

1. Penjadwal batch memulai proses batch dan diterapkan di Amazon EC2 di beberapa Availability Zone for high availability (HA).

> **Note**
>
> Pola ini tidak mencakup implementasi penjadwal batch. Untuk informasi selengkapnya tentang implementasi, lihat dokumentasi vendor perangkat lunak untuk penjadwal Anda.

2. Pekerjaan batch mainframe (ditulis pada bahasa pemrograman seperti JCL atau COBOL) menggunakan logika bisnis inti untuk memproses dan menghasilkan output cetak, seperti laporan penagihan, kartu ID, dan laporan pinjaman. Pekerjaan ini diterapkan di Amazon EC2 di dua

Availability Zone untuk HA dan menggunakan Rocket Software Print Exit untuk merutekan output cetak ke LRS VPSX/MFI untuk pencetakan pengguna akhir.

3. LRS VPSX/MFI menggunakan TCP/IP-based LRS/Queue transmission agent to collect or capture print data from the Rocket Software JES Print Exit programming interface. Print Exit passes the necessary information to enable LRS VPSX/MFI to effectively process the spool file and dynamically build LRS/Queue perintah. Perintah kemudian dijalankan menggunakan fungsi bawaan standar dari Rocket Software.

> **Note**
>
> Untuk informasi selengkapnya tentang data cetak yang diteruskan dari Rocket Software Print Exit ke LRS/Queue dan mekanisme batch mainframe yang VPSX/MFI didukung LRS, lihat Mencetak pengambilan data di bagian Informasi tambahan dari pola ini.

4.
> **Note**
>
> Network Load Balancer menyediakan nama DNS untuk mengintegrasikan Rocket Enterprise Server dengan LRS VPSX/MFI. : LRS VPSX/MFI supports a Layer 4 load balancer. The Network Load Balancer also does a basic health check on LRS VPSX/MFI dan mengarahkan lalu lintas ke target terdaftar yang sehat.

5.
> **Note**
>
> Server VPSX/MFI cetak LRS digunakan EC2 di Amazon di dua Availability Zone untuk HA dan menggunakan Amazon EBS sebagai penyimpanan data operasional. LRS VPSX/MFI mendukung mode layanan aktif-aktif dan aktif-pasif. Arsitektur ini menggunakan multiple AZs dalam pasangan aktif-pasif sebagai siaga aktif dan panas. Network Load Balancer melakukan pemeriksaan kesehatan pada instans LRS dan mengarahkan lalu lintas ke VPSX/MFI EC2 instans siaga panas di AZ lain jika instans aktif dalam keadaan tidak sehat. Permintaan cetak disimpan dalam Antrian Pekerjaan LRS secara lokal di setiap instance. EC2 Jika terjadi pemulihan, instance yang gagal harus dimulai ulang agar layanan LRS dapat melanjutkan pemrosesan permintaan cetak. : LRS juga VPSX/MFI dapat melakukan pemeriksaan kesehatan di tingkat armada printer. Untuk informasi selengkapnya, lihat Pemeriksaan kesehatan armada printer di bagian Informasi tambahan dari pola ini.

6. AWS Managed Microsoft AD terintegrasi dengan LRS/DIS untuk melakukan autentikasi dan otorisasi alur kerja cetak. Untuk informasi selengkapnya, lihat Mencetak otentikasi dan otorisasi di bagian Informasi tambahan dari pola ini.
7. LRS VPSX/MFI menggunakan Amazon EBS untuk penyimpanan blok. Anda dapat mencadangkan data Amazon EBS dari EC2 instans aktif ke Amazon S3 point-in-time sebagai snapshot dan mengembalikannya ke volume EBS siaga panas. Untuk mengotomatiskan pembuatan, penyimpanan, dan penghapusan snapshot volume Amazon EBS, Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengatur frekuensi snapshot otomatis dan memulihkannya berdasarkan persyaratan RTO/RPO Anda.

Alat

Layanan AWS

- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans. EC2 Volume EBS berfungsi seperti perangkat blok mentah yang tidak terformat. Anda dapat memasang volume ini sebagai perangkat di instans Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, dan Anda dapat meningkatkan skala atau meningkatkan skala.
- Amazon Relational Database Service (Amazon RDS) adalah layanan web yang memudahkan penyiapan, pengoperasian, dan skala database relasional di AWS Cloud. Ini menyediakan efisiensi biaya, kapasitas resizable untuk database relasional dan mengelola tugas-tugas administrasi database umum.
- AWS Directory Service untuk Microsoft Active Directory, juga dikenal sebagai AWS Managed Microsoft AD, memungkinkan beban kerja sadar direktori dan sumber daya AWS Anda untuk menggunakan Microsoft Active Directory di AWS Cloud.

Alat lainnya

- LRS VPSX/MFI (Micro Focus Interface), dikembangkan bersama oleh LRS dan Rocket Software, menangkap output dari spool JES Rocket Enterprise Server dan andal mengirimkannya ke tujuan cetak tertentu.
- LRS Directory Information Server (LRS/DIS) digunakan untuk otentikasi dan otorisasi selama alur kerja cetak.

- TCP/IP-based LRS/Queue transmission agent is used by LRS VPSX/MFIuntuk mengumpulkan atau menangkap data cetak melalui antarmuka pemrograman Rocket Software JES Print Exit.
- Rocket Enterprise Server adalah lingkungan penyebaran aplikasi untuk aplikasi mainframe. Ini menyediakan lingkungan eksekusi untuk aplikasi mainframe yang dimigrasikan atau dibuat dengan menggunakan versi Rocket Software Enterprise Developer.

Epik

Siapkan Rocket Enterprise Server di Amazon EC2 dan gunakan aplikasi batch mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Rocket Enterprise Server dan gunakan aplikasi demo.	Siapkan Rocket Enterprise Server di Amazon EC2, lalu gunakan aplikasi BankDemo demonstrasi Perangkat Lunak Rocket di Amazon. EC2 BankDemo Aplikasi ini adalah aplikasi batch mainframe yang membuat dan kemudian memulai output cetak.	Arsitek awan

Siapkan server cetak LRS di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan lisensi produk LRS untuk pencetakan.	Untuk mendapatkan lisensi produk LRS untuk LRSVPSX/MFI, LRS/Queue, and LRS/DIS, hubungi tim Manajemen Output LRS. Anda harus memberikan nama host dari EC2 contoh di mana produk LRS akan diinstal.	Membangun memimpin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance Amazon EC2 Windows untuk menginstal LRS VPSX/MFI.	Luncurkan instans Amazon EC2 Windows dengan mengikuti petunjuk dari Luncurkan EC2 instans Amazon di EC2 dokumenta si Amazon. Instans Anda harus memenuhi persyarat an perangkat keras dan perangkat lunak berikut untuk LRS VPSX/LKM: • CPU - Inti Ganda • RAM — 16 GB • Drive - 500 GB • EC2 Contoh minimum - m5.xlarge • OS - Windows/Linux • Perangkat Lunak — Layanan Informasi Internet (IIS) atau Apache **Note** Persyaratan perangkat keras dan perangkat lunak sebelumny a ditujukan untuk armada printer kecil (sekitar 500-1000). Untuk mendapatkan persyaratan lengkap, konsultasikan dengan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kontak LRS dan AWS Anda. Saat Anda membuat instance Windows, lakukan hal berikut: 1. Konfirmasikan bahwa nama EC2 host adalah nama host yang sama yang digunakan untuk lisensi produk LRS. 2. Aktifkan CGI di Amazon EC2 dengan menyelesaikan yang berikut: a. Connect ke EC2 instans Anda dengan mengikuti instruksi dari	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS VPSX/MFI pada EC2 instance.	1. Connect ke EC2 instans Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. **Note** Buka tautan ke halaman unduhan produk dari email LRS yang seharusnya Anda terima. : Produk LRS didistribusikan melalui transfer file elektronik (EFT). 3. Unduh LRS VPSX/MFI dan unzip file (folder default:`c:\LRS`). 4. Luncurkan Pemasang Produk LRS dari folder unzip untuk menginstal LRS VPSX/MFI. 5. Pada menu Select Features, pilih VPSX® Server (V1R3.022), lalu pilih Next untuk memulai proses instalasi. Anda akan menerima pesan sukses ketika instalasi selesai.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS/antrian.	1. Connect ke EC2 instance Rocket Enterprise Server Anda dengan mengikuti petunjuk dari Langkah 2: Hubungkan ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka tautan ke halaman unduhan produk LRS dari email LRS yang seharusnya Anda terima, unduh LRS/antrian, lalu unzip file. 3. Pergi ke lokasi di mana Anda men-download file, dan kemudian meluncurkan installer produk LRS untuk menginstal LRS/antrian.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS/DIS.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka tautan ke halaman unduhan produk LRS dari email LRS yang seharusnya Anda terima, unduh LRS/DIS, lalu unzip file. 3. Pergi ke lokasi di mana Anda men-download file, dan kemudian meluncurkan LRS Product Installer. 4. Di LRS Product Installer, perluas LRS Misc Tools, pilih LRS DIS, lalu pilih Next. 5. Ikuti petunjuk lainnya di Pemasang Produk LRS untuk menyelesaikan proses instalasi.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup target dan daftarkan LRS VPSX/MFI EC2 sebagai target.	Buat grup target dengan mengikuti petunjuk dari Buat grup target untuk Network Load Balancer Anda dalam dokumentasi Elastic Load Balancing. Saat Anda membuat grup target, lakukan hal berikut: 1. Pada halaman Tentukan detail grup, untuk Pilih Jenis Target, pilih Contoh. 2. Untuk Protokol, pilih TCP. 3. Untuk Port, pilih 5500. 4. Pada halaman Daftar target, di bagian Instance yang tersedia, pilih instans LRS VPSX/MFI EC2 .	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Network Load Balancer.	Ikuti instruksi dari Buat Network Load Balancer dalam dokumentasi Elastic Load Balancing. Network Load Balancer Anda merutekan lalu lintas dari Rocket Enterpris e Server ke LRS. VPSX/MFI EC2 Saat Anda membuat Network Load Balancer, lakukan hal berikut pada halaman Listeners dan Routing: 1. Untuk Protokol, pilih TCP. 2. Untuk Port, pilih 5500. 3. Untuk tindakan Default, pilih Teruskan ke grup target yang Anda buat sebelumny a.	Arsitek awan

Integrasikan Rocket Enterprise Server dengan LRS VPSX/MFI dan LRS/Antrian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Rocket Enterprise Server untuk LRS/ Queue integrasi.	1. Connect ke EC2 instance Rocket Enterprise Server Anda dengan mengikuti petunjuk dari Langkah 2: Hubungkan ke instans Anda dalam EC2 dokumentasi Amazon.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di menu Start Windows, buka UI Administrasi Server Perusahaan Rocket. 3. Di bilah menu, pilih NATIVE. 4. Di panel navigasi, pilih Directory Server, lalu pilih BANKDEMO. 5. Dari Umum di panel navigasi kiri, gulir ke bawah ke bagian Tambahan untuk mengonfigurasi variabel lingkungan (LRSQ_ADDRESS, LRSQ_PORT, LRSQ_COMMAND) untuk menunjuk ke LRSQ. 6. Untuk LRSQ_ADDRESS, masukkan alamat IP atau nama DNS Network Load Balancer yang Anda buat sebelumnya. 7. Untuk LRSQ_PORT, masukkan VPSX LRSQ Listener Port (5500). 8. Untuk LRSQ_COMMAND, masukkan lokasi jalur LRSQ yang dapat dieksekusi. **Note** LRS saat ini mendukung batas	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	karakter maksimum 50 untuk nama DNS, tetapi ini dapat berubah di masa mendatang. Jika nama DNS Anda lebih besar dari 50, maka Anda dapat menggunakan alamat IP Network Load Balancer sebagai alternatif.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Rocket Enterprise Server untuk VPSX/MFI integrasi LRS.	1. Salin `VPSX_MFI_R2` folder dari VPSX/MFI installer LRS ke lokasi Rocket Enterprise Server di. `C\BANKDEMO\print` 2. Connect ke EC2 instance Rocket Enterprise Server Anda dengan mengikuti petunjuk dari Langkah 2: Hubungkan ke instans Anda dalam EC2 dokumentasi Amazon. 3. Di menu Start Windows, buka UI Administrasi Server Perusahaan Rocket. 4. Di bilah menu, pilih NATIVE. 5. Di panel navigasi, pilih Directory Server, lalu pilih BANKDEMO. 6. Di bawah BANKDEMO, pilih JES. 7. Di bawah Jalur Program JES, tambahkan `DLL(VPSX_MFI_R2)` jalur dari `C\BANKDEMO\print` lokasi.	Arsitek awan

Siapkan printer dan cetak pengguna di Rocket Enterprise Server dan LRS VPSX/MFI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kaitkan modul Keluar Cetak Perangkat Lunak Roket ke Proses Eksekusi Server printer batch Rocket Enterprise Server.	1. Connect ke EC2 instance Rocket Enterprise Server Anda dengan mengikuti petunjuk dari Langkah 2: Hubungkan ke instans Anda dalam EC2 dokumentasi Amazon. 2. Di menu Start Windows, buka UI Administrasi Server Perusahaan Rocket. 3. Di bilah menu, pilih NATIVE. 4. Di panel navigasi, pilih Directory Server, lalu pilih BANKDEMO. 5. Di bawah BANKDEMO, pilih JES, dan gulir ke bawah ke Printer. 6. Di Printer, kaitkan modul Keluar Cetak Perangkat Lunak Roket (LRSPRTE6 untuk Batch) ke printer batch Rocket Enterprise Server Proses Eksekusi Server (SEP). Ini memungkinkan perutean keluaran cetak ke LRS VPSX/MFI. 7. Masuk ke UI Administrasi Server Perusahaan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang konfigurasi, lihat Menggunakan Keluar di dokumentasi Fokus Mikro.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan printer di LRS VPSX/MFI.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka Antarmuka Web VPSX dari menu Start Windows. 3. Di panel navigasi, pilih Printer. 4. Pilih Tambah, lalu pilih Tambah Printer. 5. Pada halaman Konfigurasi Printer, untuk Nama Printer, masukkan Lokal. 6. Untuk ID VPSX, masukkan. VPS1 7. Untuk CommType, pilih TCPIP/LRSQ. 8. Untuk Host/IP Address, masukkan alamat IP printer fisik yang ingin Anda tambahkan. 9. Untuk Perangkat, masukkan nama perangkat Anda. 10. Pilih Driver Windows atau Driver Linux/Mac. 11. Pilih Tambahkan.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna cetak di LRS VPSX/LKM.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka Antarmuka Web VPSX dari menu Start Windows. 3. Di panel navigasi, pilih Keamanan, lalu pilih Pengguna. 4. Di kolom Nama Pengguna, pilih admin, lalu pilih Salin. 5. Di jendela Pemelihar aan Profil Pengguna, untuk Nama Pengguna, masukkan nama pengguna (misalnya, PrintUser). 6. Untuk Deskripsi, masukkan deskripsi singkat (misalnya, Pengguna untuk tes cetak). 7. Pilih Perbarui. Ini menciptak an pengguna cetak (misalnya, PrintUser). 8. Di panel navigasi, di bawah Pengguna, pilih pengguna baru yang Anda buat. 9. Dari menu Command, pilih Security. 10. Pada halaman Aturan Keamanan, pilih semua opsi keamanan printer dan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	keamanan pekerjaan yang berlaku, lalu pilih Simpan. 11. Untuk menambahkan pengguna cetak baru Anda ke grup Administrator, buka panel navigasi, pilih Keamanan, lalu pilih Konfigurasi. 12. Di jendela konfigurasi Keamanan, tambahkan pengguna cetak baru Anda ke kolom Administrator.	

Mengatur otentikasi cetak dan otorisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat domain AWS Managed Microsoft AD dengan pengguna dan grup.	1. Buat Active Directory di AWS Managed Microsoft AD dengan mengikuti petunjuk dari Buat direktori AWS Managed Microsoft AD Anda di dokumentasi AWS Directory Service. 2. Menerapkan EC2 instans (Active Directory manager) dan instal alat Active Directory untuk mengelola AWS Managed Microsoft AD Anda dengan mengikuti petunjuk dari Langkah 3: Menerapkan EC2 instans untuk mengelola AWS Managed Microsoft AD	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda dalam dokumentasi AWS Directory Service. 3. **Note** Connect ke EC2 instans Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. : Saat Anda terhubung ke EC2 instance, masukkan kredensi administrator Anda (untuk direktori yang Anda buat pada langkah pertama) di jendela Windows Security. 4. Di menu Start Windows, di bawah Windows Administrative Tools, pilih Active Directory Users and Computers. 5. Buat pengguna cetak di domain Active Directory dengan mengikuti langkah-langkah dari Buat pengguna di dokumentasi layanan AWS Directory.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bergabunglah dengan LRS VPSX/MFI EC2 ke domain AWS Managed Microsoft AD.	Bergabunglah dengan LRS VPSX/MFI EC2 ke domain AWS Managed Microsoft AD Anda secara otomatis (dokumentasi AWS Knowledge Center) atau secara manual (dokumentasi AWS Directory Service).	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan integrasikan LRS/DIS dengan AWS Managed Microsoft AD.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Di menu Start Windows, buka Antarmuka Web VPSX. 3. Di panel navigasi, pilih Keamanan, lalu pilih Konfigurasi. 4. Pada halaman Konfigurasi Keamanan, di bagian Parameter Keamanan, untuk Jenis Keamanan, pilih Internal. 5. Masukkan preferensi Anda untuk sisa opsi di bagian Parameter Keamanan. 6. Buka folder LRS Output Management dari menu Microsoft Windows Start, pilih Server Start, lalu pilih Server Stop. 7. Masuk ke LRS VPSX/MFI dengan nama pengguna dan kata sandi Direktori Aktif Anda.	Arsitek awan

Uji alur kerja cetak

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai permintaan cetak batch dari BankDemo aplikasi Rocket Software.	1. Buka emulator terminal 3270 di EC2 instance Rocket Enterprise Server Anda. 2. Connect ke BankDemo aplikasi dengan menjalankan perintah berikut: `connect 127.0.0.1 :9278` 3. Pada antarmuka baris BankDemo perintah, untuk User id, masukkan B0001. Untuk Kata Sandi, masukkan kunci yang tidak kosong. 4. Untuk opsi Permintaan pernyataan tercetak, masukkan X pada baris kosong. 5. Di bagian Kirim pernyataan demi, untuk Mail, masukkan Y, lalu tekan F10.	Insinyur uji
Periksa output cetak di LRS VPSX/MFI.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Di menu Start Windows, buka Antarmuka Web VPSX.	Insinyur uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di panel navigasi, pilih Printer, lalu pilih Output Queue. 4. Di kolom ID Spool, pilih ID spool untuk permintaan dalam antrian printer. 5. Pada tab Tindakan, di kolom perintah, pilih Browse. Anda sekarang dapat melihat hasil cetak dari laporan akun dengan kolom untuk Akun No. , Deskripsi, Tanggal, Jumlah, dan Saldo. Sebagai contoh, lihat lampiran batch_pri nt_output untuk pola ini.	

Sumber daya terkait

- Modernisasi Output LRS (dokumentasi LRS)
- ANSI dan kontrol kereta mesin (dokumentasi IBM)
- Kata-kata perintah saluran (dokumentasi IBM)
- Memberdayakan Beban Kerja Mainframe Perusahaan di AWS dengan Fokus Mikro (AWS Partner Network Blog)
- Membangun PAC Server Perusahaan Fokus Mikro dengan Amazon EC2 Auto Scaling dan Systems Manager (dokumentasi AWS Prescriptive Guidance)
- Aliran data Presentasi Fungsi Lanjutan (AFP) (dokumentasi IBM)
- Aliran Data Terkondisi Jalur (LCD) (Dokumentasi Compart)

Informasi tambahan

Pertimbangan-pertimbangan

Selama perjalanan modernisasi Anda, Anda dapat mempertimbangkan berbagai konfigurasi untuk proses batch mainframe dan output yang dihasilkannya. Platform mainframe telah disesuaikan oleh setiap pelanggan dan vendor yang menggunakannya dengan persyaratan khusus yang secara langsung memengaruhi pencetakan. Misalnya, platform Anda saat ini dapat menggabungkan IBM Advanced Function Presentation (AFP) atau Xerox Line Condition Data Stream (LCDS) ke dalam alur kerja saat ini. Selain itu, karakter kontrol carriage mainframe dan kata-kata perintah saluran dapat memengaruhi tampilan halaman yang dicetak dan mungkin memerlukan penanganan khusus. Sebagai bagian dari proses perencanaan modernisasi, kami menyarankan Anda menilai dan memahami konfigurasi di lingkungan cetak spesifik Anda.

Cetak pengambilan data

Rocket Software Print Exit meneruskan informasi yang diperlukan untuk memungkinkan LRS VPSX/MFI memproses file spool secara efektif. Informasi terdiri dari bidang yang dilewatkan di blok kontrol yang relevan, seperti:

- NAMA KERJA
- PEMILIK (USERID)
- TUJUAN
- BENTUK
- NAMA BERKAS
- PENULIS

LRS VPSX/MFI mendukung mekanisme batch mainframe berikut untuk menangkap data dari Rocket Enterprise Server.

- print/spool Pemrosesan BATCH COBOL menggunakan pernyataan z/OS JCL SYSOUT standar DD/OUTPUT
- print/spool Pemrosesan BATCH COBOL menggunakan pernyataan z/OS JCL CA-SPOOL SUBSYS DD standar
- IMS/COBOL print/spoolpemrosesan menggunakan antarmuka CBLTDLI (Untuk daftar lengkap metode dan contoh pemrograman yang didukung, lihat dokumentasi LRS yang disertakan dengan lisensi produk Anda.)

Pemeriksaan kesehatan armada printer

LRS VPSX/MFI (LRS LoadX) dapat melakukan pemeriksaan kesehatan deep dive, termasuk manajemen perangkat dan optimalisasi operasional. Manajemen perangkat dapat mendeteksi kegagalan pada perangkat printer dan merutekan permintaan cetak ke printer yang sehat. Untuk informasi selengkapnya tentang pemeriksaan kesehatan deep dive untuk armada printer, lihat dokumentasi LRS yang disertakan dengan lisensi produk Anda.

Cetak otentikasi dan otorisasi

LRS/DIS enables LRS applications to authenticate user IDs and passwords by using Microsoft Active Directory or an LDAP server. In addition to basic print authorization, LRS/DISjuga dapat menerapkan kontrol keamanan cetak tingkat butiran dalam kasus penggunaan berikut:

- Kelola siapa yang dapat menelusuri pekerjaan printer.
- Kelola tingkat penelusuran pekerjaan pengguna lain.
- Kelola tugas operasional. Misalnya, keamanan tingkat perintah seperti hold/release, purge, modify, copy, dan reroute. Keamanan dapat diatur oleh User-ID atau Grup (mirip dengan grup AD atau grup LDAP).

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Modernisasi mainframe: aktif AWS dengan Rocket DevOps Software Enterprise Suite

Kevin Yung, Amazon Web Services

Ringkasan

Tantangan pelanggan

Organizations yang menjalankan aplikasi inti pada perangkat keras mainframe biasanya menghadapi beberapa tantangan ketika perangkat keras perlu ditingkatkan untuk memenuhi tuntutan inovasi digital. Tantangan-tantangan ini termasuk kendala berikut.

- Pengembangan mainframe dan lingkungan pengujian tidak dapat diskalakan karena ketidakfleksibelan komponen perangkat keras mainframe dan tingginya biaya perubahan.
- Pengembangan mainframe menghadapi kekurangan keterampilan, karena pengembang baru tidak akrab dan tidak tertarik dengan alat pengembangan mainframe tradisional. Teknologi modern seperti kontainer, pipa integration/continuous pengiriman berkelanjutan (CI/CD), dan kerangka kerja pengujian modern tidak tersedia dalam pengembangan mainframe.

Hasil pola

Untuk mengatasi tantangan ini, Amazon Web Services (AWS) dan Rocket Software Micro Focus, Mitra AWS Partner Network (APN), telah berkolaborasi untuk membuat pola ini. Solusi ini dirancang untuk membantu Anda mencapai hasil berikut.

- Peningkatan produktivitas pengembang. Pengembang dapat diberikan instance pengembangan mainframe baru dalam beberapa menit.
- Penggunaan AWS Cloud untuk membuat lingkungan pengujian mainframe baru dengan kapasitas hampir tidak terbatas.
- Penyediaan cepat infrastruktur CI/CD mainframe baru. Penyediaan pada AWS dapat diselesaikan dalam waktu satu jam dengan menggunakan AWS CloudFormation dan. AWS Systems Manager
- Penggunaan asli AWS DevOps alat untuk pengembangan mainframe, termasuk,,, AWS CodeBuild AWS CodeCommit AWS CodePipeline AWS CodeDeploy, dan Amazon Elastic Container Registry (Amazon ECR).
- Mengubah pengembangan air terjun tradisional menjadi pengembangan tangkas dalam proyek mainframe.

Ringkasan teknologi

Dalam pola ini, tumpukan target berisi komponen-komponen berikut.

Komponen logis	Solusi implementasi	Deskripsi
Repositori kode sumber	AccuRev Server Perangkat Lunak Roket CodeCommit, Amazon ECR	Manajemen kode sumber — Solusinya menggunakan dua jenis kode sumber: • Kode sumber mainframe, misalnya, COBOL dan JCL. • AWS templat infrastruktur dan skrip otomatisasi Kedua jenis kode sumber memerlukan kontrol versi, tetapi mereka dikelola dengan cara yang berbeda SCMs. Kode sumber yang digunakan ke mainframe atau Rocket Software Enterprise Server dikelola di Rocket Software Micro Focus AccuRev Server. AWS template dan skrip otomatisasi dikelola di CodeCommit. Amazon ECR digunakan untuk repositori gambar Docker.
Contoh pengembang perusahaan	Amazon Elastic Compute Cloud (Amazon EC2), Pengembang Perusahaan Perangkat Lunak Roket untuk Eclipse	Pengembang mainframe dapat mengembangkan kode di Amazon EC2 dengan menggunakan Pengemban g Perusahaan Perangkat Lunak Rocket untuk Eclipse. Ini menghilangkan kebutuhan

		untuk mengandalkan perangkat keras mainframe untuk menulis dan menguji kode.
Manajemen lisensi Rocket Software Enterprise Suite	License Manager Software Enterprise Suite Rocket	Untuk manajemen lisensi dan tata kelola Rocket Software Enterprise Suite terpusat, solusinya menggunakan Rocket Software Enterpris e Suite License Manager untuk meng-host lisensi yang diperlukan.
Pipa CI/CD	CodePipeline, CodeBuild, CodeDeploy, Pengembang Perusahaan Perangkat Lunak Roket dalam wadah, Server Uji Perusahaan Perangkat Lunak Roket dalam wadah, Server Perusahaan Fokus Mikro Perangkat Lunak Roket	Tim pengembangan mainframe memerlukan CI/ CD pipeline untuk melakukan kompilasi kode, pengujian integrasi, dan pengujian regresi. Di AWS, CodePipel ine dan CodeBuild dapat bekerja dengan Pengemban g Perusahaan Perangkat Lunak Rocket dan Server Uji Perusahaan dalam wadah secara asli.

Prasyarat dan batasan

Prasyarat

Nama	Deskripsi
py3270	py3270 adalah antarmuka Python ke x3270, emulator terminal IBM 3270. Ini menyediakan API ke subproses x3270 atau s3270.

x3270	x3270 adalah emulator terminal IBM 3270 untuk X Window System dan Windows. Ini dapat digunakan oleh pengembang untuk pengujian unit secara lokal.
Robot-Kerangka-Mainframe-3270-Perpustakaan	Mainframe3270 adalah perpustakaan untuk Robot Framework berdasarkan proyek py3270.
Perangkat Lunak Roket Verastream	Rocket Software Verastream adalah platform integrasi yang memungkinkan pengujian aset mainframe seperti aplikasi seluler, aplikasi web, dan layanan web SOA diuji.
Penginstal dan lisensi Pengujian Fungsional Terpadu Perangkat Lunak Roket (UFT)	Rocket Software Unified Functional Testing adalah perangkat lunak yang menyediakan otomatisasi uji fungsional dan regresi untuk aplikasi dan lingkungan perangkat lunak.
Pemasang dan lisensi Server Perusahaan Perangkat Lunak Rocket	Enterprise Server menyediakan lingkungan runtime untuk aplikasi mainframe.
Penginstal dan lisensi Server Uji Perusahaan Perangkat Lunak Rocket	Rocket Software Enterprise Test Server adalah lingkungan pengujian aplikasi mainframe IBM.
AccuRev Penginstal dan lisensi Rocket Software untuk Server, dan Rocket Software Micro Focus AccuRev installer dan lisensi untuk sistem operasi Windows dan Linux	AccuRev menyediakan manajemen kode sumber (SCM). AccuRev Sistem ini dirancang untuk digunakan oleh tim orang yang sedang mengembangkan satu set file.
Pengembang Perusahaan Perangkat Lunak Roket untuk penginstal, tambalan, dan lisensi Eclipse	Enterprise Developer menyediakan pengembang mainframe platform untuk mengembangkan dan memelihara mainframe inti aplikasi online dan batch.

Batasan

- Membangun gambar Windows Docker tidak didukung di CodeBuild. Masalah yang dilaporkan ini membutuhkan dukungan dari tim Windows Kernel/HCS dan Docker. Solusinya adalah membuat

runbook build image Docker dengan menggunakan Systems Manager. Pola ini menggunakan solusi untuk membangun Rocket Software Enterpise Developer untuk gambar Eclipse dan Rocket Software Micro Focus Enterprise Test Server Container.

- Konektivitas virtual private cloud (VPC) dari belum CodeBuild didukung di Windows, sehingga polanya tidak menggunakan Rocket Software License Manager untuk mengelola lisensi di Rocket Software Enterprise Developer dan OpenText Rocket Software Enterprise Test Server container.

Versi produk

- Pengembang Perusahaan Perangkat Lunak Rocket 5.5 atau yang lebih baru
- Server Uji Perusahaan Perangkat Lunak Rocket 5.5 atau yang lebih baru
- Rocket Software Enterprise Server 5.5 atau yang lebih baru
- Rocket Software AccuRev 7.x atau yang lebih baru
- Gambar dasar Windows Docker untuk Pengembang Perusahaan Perangkat Lunak Roket dan Server Uji Perusahaan: microsoft/dotnet-framework-4.7.2-runtime
- Gambar dasar Linux Docker untuk AccuRev klien: amazonlinux: 2

Arsitektur

Lingkungan mainframe

Dalam pengembangan mainframe konvensional, pengembang perlu menggunakan perangkat keras mainframe untuk mengembangkan dan menguji program. Mereka menghadapi keterbatasan kapasitas, misalnya terbatas juta instruksi per detik (MIPS) untuk dev/test lingkungan, dan mereka harus bergantung pada alat yang tersedia di komputer mainframe.

Di banyak organisasi, pengembangan mainframe mengikuti metodologi pengembangan air terjun, dengan tim mengandalkan siklus panjang untuk merilis perubahan. Siklus rilis ini biasanya lebih lama dari pengembangan produk digital.

Diagram berikut menunjukkan beberapa proyek mainframe berbagi perangkat keras mainframe untuk pengembangannya. Dalam perangkat keras mainframe, mahal untuk meningkatkan pengembangan dan pengujian lingkungan untuk lebih banyak proyek.

AWSarsitektur

Pola ini memperluas pengembangan mainframe ke. AWS Cloud Pertama, ia menggunakan AccuRev SCM untuk meng-host kode sumber mainframe. AWS Kemudian itu membuat Enterprise Developer dan Enterprise Test Server tersedia untuk membangun dan menguji kode mainframe. AWS

Bagian berikut menjelaskan tiga komponen utama pola.

1. SCM

Dalam AWS, pola digunakan AccuRev untuk membuat satu set ruang kerja SCM dan kontrol versi untuk kode sumber mainframe. Arsitektur berbasis streamingnya memungkinkan pengembangan mainframe paralel untuk beberapa tim. Untuk menggabungkan perubahan, AccuRev gunakan konsep promosi. Untuk menambahkan perubahan itu ke ruang kerja lain, AccuRev gunakan konsep pembaruan.

Pada tingkat proyek, setiap tim dapat membuat satu atau lebih aliran AccuRev untuk melacak perubahan tingkat proyek. Ini disebut aliran proyek. Aliran proyek ini diwarisi dari aliran induk yang sama. Aliran induk digunakan untuk menggabungkan perubahan dari aliran proyek yang berbeda.

Setiap aliran proyek dapat mempromosikan kode ke AccuRev, dan pemicu pos promosi disiapkan untuk memulai pipa AWS CI/CD. Build yang berhasil untuk perubahan aliran proyek dapat dipromosikan ke aliran induknya untuk pengujian regresi lainnya.

Biasanya, aliran induk disebut aliran integrasi sistem. Ketika ada promosi dari aliran proyek ke aliran integrasi sistem, pemicu pasca promosi memulai CI/CD pipeline lain untuk menjalankan pengujian regresi.

Selain kode mainframe, pola ini mencakup AWS CloudFormation template, dokumen Otomasi Systems Manager, dan skrip. Mengikuti praktik infrastructure-as-code terbaik, mereka dikendalikan versi. CodeCommit

Jika Anda perlu menyinkronkan kode mainframe kembali ke lingkungan mainframe untuk penyebaran, Rocket Software menyediakan solusi Enterprise Sync, yang menyinkronkan kode dari SCM kembali ke AccuRev SCM mainframe.

2. Lingkungan pengembang dan pengujian

Dalam organisasi besar, penskalaan lebih dari seratus atau bahkan lebih dari seribu pengembang mainframe sangat menantang. Untuk mengatasi kendala ini, pola menggunakan instance Amazon EC2 Windows untuk pengembangan. Pada instance, Enterprise Developer untuk alat Eclipse diinstal.

Pengembang dapat melakukan semua pengujian kode mainframe dan debugging secara lokal pada instance.

AWS Systems Manager Dokumen State Manager dan Automation digunakan untuk mengotomatiskan penyediaan instans pengembang. Waktu rata-rata untuk membuat instance pengembang adalah dalam 15 menit. Perangkat lunak dan konfigurasi berikut disiapkan:

- AccuRev Klien Windows untuk memeriksa dan memasukkan kode sumber ke AccuRev
- Pengembang Perusahaan untuk alat Eclipse, untuk menulis, menguji, dan men-debug kode mainframe secara lokal
- Kerangka pengujian open source Kerangka pengujian pengembangan berbasis perilaku Python (BDD) Behavie, py3270, dan emulator x3270 untuk membuat skrip untuk menguji aplikasi
- Alat pengembang Docker untuk membangun image Enterprise Test Server Docker dan menguji aplikasi dalam wadah Enterprise Test Server Docker

Dalam siklus pengembangan, pengembang menggunakan EC2 instance untuk mengembangkan dan menguji kode mainframe secara lokal. Ketika perubahan lokal berhasil diuji, pengembang mempromosikan perubahan ke AccuRev server.

3. CI/CD jaringan pipa

Dalam polanya, CI/CD pipeline digunakan untuk pengujian integrasi dan uji regresi sebelum penerapan ke lingkungan produksi.

Seperti yang dijelaskan di bagian SCM, AccuRev menggunakan dua jenis aliran: aliran proyek dan aliran integrasi. Setiap aliran dihubungkan dengan CI/CD jaringan pipa. Untuk melakukan integrasi antara AccuRev server dan AWS CodePipeline, pola menggunakan skrip promosi AccuRev pos untuk membuat acara untuk memulai CI/CD.

Misalnya, ketika pengembang mempromosikan perubahan ke aliran proyek di AccuRev, ia memulai skrip promosi posting untuk dijalankan di AccuRev Server. Kemudian skrip mengunggah metadata perubahan ke bucket Amazon Simple Storage Service (Amazon S3) untuk membuat acara Amazon S3. Acara ini akan memulai pipeline yang CodePipeline dikonfigurasi untuk dijalankan.

Mekanisme memulai peristiwa yang sama digunakan untuk aliran integrasi dan jaringan pipa terkaitnya.

Dalam CI/CD pipeline, CodePipeline gunakan CodeBuild dengan wadah klien AccuRev Linux untuk memeriksa kode terbaru dari AccuRev aliran. Kemudian pipeline CodeBuild mulai menggunakan

wadah Enterprise Developer Windows untuk mengkompilasi kode sumber, dan menggunakan wadah Enterprise Test Server Windows CodeBuild untuk menguji aplikasi mainframe.

CI/CD Pipa dibangun menggunakan CloudFormation templat, dan cetak biru akan digunakan untuk proyek baru. Dengan menggunakan template, dibutuhkan waktu kurang dari satu jam bagi sebuah proyek untuk membuat CI/CD pipeline baru AWS.

Untuk meningkatkan kemampuan pengujian mainframe Anda AWS, pola tersebut membangun rangkaian DevOps pengujian Perangkat Lunak Rocket, Verastream, dan server UFT. Dengan menggunakan DevOps alat modern, Anda dapat menjalankan tes AWS sebanyak yang Anda butuhkan.

Contoh lingkungan pengembangan mainframe dengan Rocket Software on AWS ditunjukkan pada diagram berikut.

Tumpukan teknologi target

Bagian ini memberikan pandangan lebih dekat pada arsitektur masing-masing komponen dalam pola.

1. Repositori kode sumber - SCM AccuRev

AccuRev SCM diatur untuk mengelola versi kode sumber mainframe. Untuk ketersediaan tinggi, AccuRev mendukung mode primer dan replika. Operator dapat gagal ke replika saat melakukan pemeliharaan pada node utama.

Untuk mempercepat respons CI/CD pipeline, pola menggunakan Amazon CloudWatch Events untuk mendeteksi perubahan kode sumber dan memulai awal pipeline.

1. Pipeline diatur untuk menggunakan sumber Amazon S3.
2. Aturan CloudWatch Peristiwa diatur untuk menangkap peristiwa S3 dari bucket S3 sumber.
3. Aturan CloudWatch Events menetapkan target ke pipeline.
4. AccuRev SCM dikonfigurasi untuk menjalankan skrip promosi pos secara lokal setelah promosi selesai.
5. AccuRev SCM menghasilkan file XMLyang berisi metadata promosi, dan skrip mengunggah file XMLnya ke bucket S3 sumber.

6. Setelah upload, bucket S3 sumber mengirimkan event agar sesuai dengan aturan CloudWatch Events, dan aturan CloudWatch Events memulai pipeline untuk dijalankan.

Ketika pipeline berjalan, ia memulai CodeBuild proyek untuk menggunakan wadah klien AccuRev Linux untuk memeriksa kode mainframe terbaru dari aliran terkait AccuRev .

Diagram berikut menunjukkan setup AccuRev Server.

2. Templat Pengembang Perusahaan

Pola menggunakan EC2 template Amazon untuk menyederhanakan pembuatan instance pengembang. Dengan menggunakan State Manager, ia dapat menerapkan pengaturan perangkat lunak dan lisensi ke EC2 instance secara konsisten.

EC2 Template Amazon dibangun dalam pengaturan konteks VPC dan pengaturan instans default, dan mengikuti persyaratan penandaan perusahaan. Dengan menggunakan template, tim dapat membuat instance pengembangan baru mereka sendiri.

Ketika instans pengembang dimulai, dengan mengaitkan dengan tag, Systems Manager menggunakan State Manager untuk menerapkan otomatisasi. Otomatisasi mencakup langkah-langkah umum berikut.

1. Instal perangkat lunak Enterprise Developer dan instal tambalan.
2. Instal AccuRev klien untuk Windows.
3. Instal skrip yang telah dikonfigurasi sebelumnya bagi pengembang untuk bergabung dengan AccuRev aliran. Inisialisasi ruang kerja Eclipse.
4. Instal alat pengembangan, termasuk x3270, py3270, dan Docker.
5. Konfigurasikan pengaturan lisensi untuk menunjuk ke penyeimbang beban License Manager.

Diagram berikut menunjukkan instance pengembang Enterprise yang dibuat oleh EC2 template Amazon, dengan perangkat lunak dan konfigurasi yang diterapkan ke instance oleh State Manager. Instans pengembang perusahaan terhubung AWS License Manager untuk mengaktifkan lisensi mereka.

3. Pipa CI/CD

Seperti yang dijelaskan di bagian AWS arsitektur, dalam polanya, ada CI/CD jaringan pipa tingkat proyek dan jaringan pipa integrasi sistem. Setiap tim proyek mainframe membuat pipa atau beberapa CI/CD saluran pipa untuk membangun program yang mereka kembangkan dalam sebuah proyek. CI/CD Pipeline proyek ini memeriksa kode sumber dari AccuRev aliran terkait.

Dalam tim proyek, pengembang mempromosikan kode mereka di AccuRev aliran terkait. Kemudian promosi memulai pipeline proyek untuk membangun kode dan menjalankan tes integrasi.

Setiap CI/CD pipeline proyek menggunakan CodeBuild proyek dengan alat Pengembang Perusahaan gambar Amazon ECR dan alat Enterprise Test Server image Amazon ECR.

CodePipeline dan CodeBuild digunakan untuk membuat jaringan CI/CD pipa. Karena CodeBuild dan tidak CodePipeline memiliki biaya atau komitmen di muka, Anda hanya membayar untuk apa yang Anda gunakan. Dibandingkan dengan perangkat keras mainframe, AWS solusinya sangat mengurangi lead time penyediaan perangkat keras dan menurunkan biaya lingkungan pengujian Anda.

Dalam perkembangan modern, beberapa metodologi pengujian digunakan. Misalnya, pengembangan berbasis tes (TDD), BDD, dan Robot Framework. Dengan pola ini, pengembang dapat menggunakan alat modern ini untuk pengujian mainframe. Misalnya, dengan menggunakan x3270, py3270 dan alat uji Behaw python, Anda dapat menentukan perilaku aplikasi online. Anda juga dapat menggunakan kerangka kerja robot build mainframe 3270 di pipeline ini CI/CD .

Diagram berikut menunjukkan CI/CD pipa aliran tim.

Diagram berikut menunjukkan laporan CI/CD uji proyek yang dihasilkan oleh CodePipeline dalam Mainframe3270 Robot Framework.

Diagram berikut menunjukkan laporan CI/CD uji proyek yang dihasilkan oleh CodePipeline di Py3270 dan Behaw BDD.

Setelah pengujian tingkat proyek berhasil dilewati, kode yang diuji secara manual dipromosikan ke aliran integrasi di AccuRev SCM. Anda dapat mengotomatiskan langkah ini setelah tim memiliki kepercayaan diri pada cakupan pengujian dari pipa proyek mereka.

Saat kode dipromosikan, CI/CD pipeline integrasi sistem memeriksa kode gabungan dan melakukan tes regresi. Kode gabungan dipromosikan dari semua aliran proyek paralel.

Bergantung pada seberapa halus lingkungan pengujian diperlukan, pelanggan dapat memiliki lebih banyak CI/CD jaringan pipa integrasi sistem di lingkungan yang berbeda, misalnya UAT, Pra-Produksi.

Dalam polanya, alat yang digunakan dalam pipeline integrasi sistem adalah Enterprise Test Server, UFT Server, dan Verastream. Semua alat ini dapat digunakan ke dalam wadah Docker dan digunakan dengan. CodeBuild

Setelah berhasil menguji program mainframe, artefak disimpan, dengan kontrol versi, dalam ember S3.

Diagram berikut menunjukkan CI/CD pipa integrasi sistem.

Setelah artefak berhasil diuji dalam CI/CD jaringan pipa integrasi sistem, artefak dapat dipromosikan untuk penyebaran produksi.

Jika Anda perlu menyebarkan kode sumber kembali ke mainframe, Rocket Software menawarkan solusi Enterprise Sync untuk menyinkronkan kode sumber dari AccuRev kembali ke Mainframe Endeavour.

Diagram berikut menunjukkan CI/CD pipa produksi yang menyebarkan artefak ke Server Perusahaan. Dalam contoh ini, CodeDeploy mengatur penyebaran artefak mainframe yang diuji ke Enterprise Server.

Selain panduan arsitektur CI/CD pipeline, lihat posting AWS DevOps blog Otomatiskan ribuan pengujian mainframe AWS dengan Micro Focus Enterprise Suite untuk informasi lebih lanjut tentang pengujian aplikasi mainframe di dan. CodeBuild CodePipeline (Fokus Mikro sekarang adalah Perangkat Lunak Roket.) Lihat posting blog untuk praktik terbaik dan detail melakukan tes mainframe. AWS

Alat

AWS alat otomatisasi

- CloudFormation
- CloudWatch Acara Amazon
- AWS CodeBuild

- AWS CodeDeploy
- AWS CodePipeline
- Amazon ECR
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

Alat Perangkat Lunak Roket

- Pengembang Perusahaan Roket untuk Eclipse
- Server Uji Perusahaan Roket
- Server Perusahaan Roket (penyebaran produksi)
- Perangkat Lunak Roket AccuRev
- License Manager Software Enterprise Suite Rocket
- Perangkat Lunak Roket Verastream Host Integrator
- Perangkat Lunak Roket UFT One

Alat-alat lainnya

- x3270
- py3270
- Robot-Kerangka-Mainframe-3270-Perpustakaan

Epik

Buat infrastruktur AccuRev SCM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan server AccuRev SCM utama dengan menggunakan. CloudForm ation		AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna AccuRev Administrator.	Masuk ke AccuRev SCM Server, dan jalankan perintah CLI untuk membuat pengguna Administrator.	AccuRev Administrator Server SCM
Buat AccuRev aliran.	Buat AccuRev aliran yang mewarisi dari aliran atas secara berurutan: Produksi, Integrasi Sistem, Aliran tim.	AccuRev Administrator SCM
Buat akun AccuRev login pengembang.	Gunakan perintah AccuRev SCM CLI untuk AccuRev membuat akun login pengguna untuk pengembang mainframe.	AccuRev Administrator SCM

Buat template EC2 peluncuran Enterprise Developer Amazon

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan template EC2 peluncuran Amazon dengan menggunakan CloudForm ation.	Gunakan CloudFormation untuk menerapkan template EC2 peluncuran Amazon untuk instans Enterprise Developer. Template termasuk dokumen Automation Systems Manager untuk instance Pengembang Rocket Enterpris e.	AWS CloudFormation
Buat instance Enterprise Developer dari EC2 template Amazon.		Login Konsol AWS dan Keterampilan Pengembang Mainframe

Buat gambar Docker alat Pengembang Perusahaan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gambar Docker alat Pengembang Perusahaan.	Gunakan perintah Docker dan alat Enterprise Developer Dockerfile untuk membuat image Docker.	Docker
Buat repositori Docker di Amazon ECR.	Di konsol Amazon ECR, buat repositori untuk image Enterprise Developer Docker.	Amazon ECR
Dorong gambar Docker alat Pengembang Perusahaan ke Amazon ECR.	Jalankan perintah push Docker untuk mendorong image alat Docker Enterprise Developer untuk menyimpan nya di repositori Docker di Amazon ECR.	Docker

Buat gambar Enterprise Test Server Docker

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat image Enterprise Test Server Docker.	Gunakan perintah Docker dan Enterprise Test Server Dockerfile untuk membuat image Docker.	Docker
Buat repositori Docker di Amazon ECR.	Di konsol Amazon ECR, buat repositori Amazon ECR untuk image Enterprise Test Server Docker.	Amazon ECR
Dorong image Enterprise Test Server Docker ke Amazon ECR.	Jalankan perintah push Docker untuk mendorong dan menyimpan image Enterprise	Docker

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Test Server Docker di Amazon ECR.	

Buat CI/CD pipeline aliran tim

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CodeCommit repositori.	Di CodeCommit konsol, buat repositori berbasis Git untuk infrastruktur dan kode. CloudFormation	AWS CodeCommit
Unggah CloudFormation template dan kode otomatisasi ke dalam CodeCommit repositori.	Jalankan perintah Git push untuk mengunggah CloudFormation template dan kode otomatisasi ke dalam repositori.	Git
Terapkan CI/CD pipa aliran tim dengan menggunakan CloudFormation.	Gunakan CloudFormation template yang disiapkan untuk menerapkan CI/CD pipeline aliran tim.	AWS CloudFormation

Buat CI/CD pipa integrasi sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat gambar UFT Docker.	Gunakan perintah Docker dan UFT Dockerfile untuk membuat image Docker.	Docker
Buat repositori Docker di Amazon ECR untuk gambar UFT.	Di konsol Amazon ECR, buat repositori Docker untuk gambar UFT.	Amazon ECR

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dorong gambar UFT Docker ke Amazon ECR.	Jalankan perintah push Docker untuk mendorong dan menyimpan image Enterprise Test Server Docker di Amazon ECR.	Docker
Buat gambar Verastream Docker.	Gunakan perintah Docker dan Verastream Dockerfile untuk membuat image Docker.	Docker
Buat repositori Docker di Amazon ECR untuk gambar Verastream.	Di konsol Amazon ECR, buat repositori Docker untuk gambar Verastream.	Amazon ECR
Menyebarkan CI/CD pipa integrasi sistem dengan menggunakan CloudFormation.	Gunakan CloudFormation template yang disiapkan untuk menyebarkan CI/CD pipeline integrasi sistem.	AWS CloudFormation

Buat pipa penyebaran CI/CD produksi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan Enterprise Server dengan menggunakan AWS Quick Start.	Untuk menggunakan Enterprise Server dengan menggunakan CloudFormation, luncurkan Enterprise Server pada AWS Quick Start.	AWS CloudFormation
Menyebarkan pipa penyebaran CI/CD produksi.	Di CloudFormation konsol, gunakan CloudFormation template untuk menerapkan pipeline penerapan CI/CD produksi.	AWS CloudFormation

Sumber daya terkait

Referensi

- AWS DevOps Blog - Otomatiskan ribuan tes mainframe AWS dengan Micro Focus Enterprise Suite (Micro Focus sekarang adalah Rocket Software.)
- repositori py3270/py3270 GitHub
- GitHub Altran-PT-GDC/Robot-Framework-Mainframe-3270-repositori perpustakaan
- Selamat datang untuk berperilaku!
- Blog Mitra APN - Tag: Fokus Mikro (Fokus Mikro sekarang adalah Perangkat Lunak Roket.)
- Meluncurkan instance dari template peluncuran

AWS Marketplace

- Perangkat Lunak Roket UFT One

AWS Mulai Cepat

- Server Perusahaan Roket di AWS

Modernisasi beban kerja pencetakan online mainframe di AWS dengan menggunakan Micro Focus Enterprise Server dan LRS VPSX/MFI

Shubham Roy dan Kevin Yung, Amazon Web Services

Abraham Rondon, Fokus Mikro

Guy Tucker, Levi, Ray dan Shoup Inc

Ringkasan

Pola ini menunjukkan kepada Anda cara memodernisasi beban kerja pencetakan online mainframe bisnis penting Anda di Amazon Web Services (AWS) Cloud dengan menggunakan Micro Focus Enterprise Server sebagai runtime untuk aplikasi mainframe modern dan VPSX/MFI LRS (Micro Focus Interface) sebagai server cetak. Pola ini didasarkan pada pendekatan modernisasi mainframe replatform. Dalam pendekatan ini, Anda memigrasikan aplikasi online mainframe Anda ke Amazon Elastic Compute Cloud (Amazon EC2) dan memigrasikan database mainframe Anda, seperti IBM for DB2) z/OS, to Amazon Relational Database Service (Amazon RDS). The authentication and authorization for the modernized print workflow is performed by AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD. The LRS Directory Information Server (LRS/DIS terintegrasi dengan AWS Managed Microsoft AD untuk autentikasi dan otorisasi alur kerja cetak. Dengan memodernisasi beban kerja pencetakan online Anda, Anda dapat mengurangi biaya infrastruktur TI, mengurangi utang teknis pemeliharaan sistem lama, menghapus silo data, meningkatkan kelincahan dan efisiensi dengan DevOps model, dan memanfaatkan sumber daya dan otomatisasi sesuai permintaan di AWS Cloud.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Beban kerja pencetakan online mainframe atau manajemen output
- Pengetahuan dasar tentang cara membangun kembali dan mengirimkan aplikasi mainframe yang berjalan di Micro Focus Enterprise Server (Untuk informasi selengkapnya, lihat lembar data Enterprise Server dalam dokumentasi Micro Focus.)
- Pengetahuan dasar tentang solusi dan konsep pencetakan cloud LRS (Untuk informasi lebih lanjut, lihat Modernisasi Keluaran dalam dokumentasi LRS.)

- Perangkat lunak dan lisensi Micro Focus Enterprise Server (Untuk informasi lebih lanjut, hubungi penjualan Micro Focus.)
- VPSX/MFI, LRS/Queue, and LRS/DISPerangkat lunak dan lisensi LRS (Untuk informasi lebih lanjut, hubungi penjualan LRS.)

> **Note**
>
> Untuk informasi selengkapnya tentang pertimbangan konfigurasi untuk beban kerja pencetakan online mainframe, lihat Pertimbangan di bagian Informasi tambahan dari pola ini.

Versi produk

- Micro Focus Enterprise Server 8.0 atau yang lebih baru
- LRS VPSX/MFI V1R3 atau yang lebih baru

Arsitektur

Tumpukan teknologi sumber

- Sistem operasi — IBM z/OS
- Bahasa pemrograman — Common Business Oriented Language (COBOL) dan Customer Information Control System (CICS)
- Database — IBM DB2 untuk Sistem Manajemen Informasi z/OS IBM (IMS) dan Metode Akses Penyimpanan Virtual (VSAM)
- Keamanan — Fasilitas Kontrol Akses Sumber Daya (RACF), CA Top Secret untuk z/OS, dan Fasilitas Kontrol Akses 2 () ACF2
- Manajemen pencetakan dan output - Produk z/OS pencetakan mainframe IBM (IBM Infoprint Server untuk z/OS, LRS, dan CA View)

Tumpukan teknologi target

- Sistem operasi - Microsoft Windows Server berjalan di Amazon EC2
- Hitung - Amazon EC2
- Bahasa pemrograman — COBOL dan CICS

- Basis Data - Amazon RDS
- Keamanan — AWS Managed Microsoft AD
- Manajemen pencetakan dan output — Solusi pencetakan LRS di AWS
- Lingkungan runtime mainframe - Server Perusahaan Fokus Mikro

Arsitektur sumber

Diagram berikut menunjukkan arsitektur keadaan saat ini yang khas untuk beban kerja pencetakan online mainframe.

Diagram menunjukkan alur kerja berikut:

1. Pengguna melakukan transaksi bisnis pada sistem keterlibatan (SoE) yang dibangun di atas aplikasi IBM CICS yang ditulis dalam COBOL.
2. SoE memanggil layanan mainframe, yang mencatat data transaksi bisnis dalam database system-of-records (SoR) seperti IBM DB2 untuk z/OS.
3. SoR mempertahankan data bisnis dari BUMN.
4. Seorang pengguna memulai permintaan untuk menghasilkan output cetak dari CICS SoE, yang memulai aplikasi transaksi cetak untuk memproses permintaan cetak.
5. Aplikasi transaksi cetak (seperti program CICS dan COBOL) mengekstrak data dari database, memformat data sesuai dengan kebutuhan bisnis, dan menghasilkan output bisnis (data cetak) seperti laporan penagihan, kartu identitas, atau laporan pinjaman. Kemudian, aplikasi mengirimkan permintaan cetak dengan menggunakan Virtual Telecommunications Access Method (VTAM). Server z/OS cetak (seperti IBM Infoprint Server) menggunakan NetSpool atau komponen VTAM serupa untuk mencegat permintaan cetak, dan kemudian membuat set data keluaran cetak pada spool JES dengan menggunakan parameter keluaran JES. Parameter keluaran JES menentukan informasi routing yang digunakan server cetak untuk mengirimkan output ke printer jaringan tertentu. Istilah VTAM mengacu pada z/OS Communications Server dan System Network Architecture (SNA) elemen layanan z/OS.
6. Komponen transmisi keluaran pencetakan mentransmisikan dataset cetak keluaran dari spool JES ke printer jarak jauh atau server cetak, seperti LRS (seperti yang ditunjukkan dalam pola ini), IBM Infoprint Server, atau tujuan email.

Arsitektur target

Diagram berikut menunjukkan arsitektur untuk beban kerja pencetakan online mainframe yang diterapkan di AWS Cloud:

Diagram menunjukkan alur kerja berikut:

1. Pengguna memulai permintaan cetak dari antarmuka pengguna online (CICS) untuk membuat hasil cetak, seperti laporan penagihan, kartu ID, atau laporan pinjaman.
2. Aplikasi online mainframe (direplatformed ke Amazon EC2) menggunakan runtime Micro Focus Enterprise Server untuk mengekstrak data dari database aplikasi, menerapkan logika bisnis ke data, memformat data, dan kemudian mengirim data ke tujuan cetak dengan menggunakan Micro Focus CICS Print Exit (DFHUPRNT).
3. Database aplikasi (SoR yang berjalan di Amazon RDS) menyimpan data untuk output cetak.
4. Solusi VPSX/MFI pencetakan LRS digunakan di Amazon EC2, dan data operasionalnya disimpan di Amazon Elastic Block Store (Amazon EBS). LRS VPSX/MFI menggunakan nama TCP/IP-based LRS/Queue transmission agent to collect print data through the Micro Focus CICS Print Exit API (DFHUPRNT) and deliver the data to a specified printer destination. The original TERMID (TERM) that's used in the modernized CICS application is used as the VPSX/MFI Antrian.

> **Note**
>
> Solusi target biasanya tidak memerlukan perubahan aplikasi untuk mengakomodasi bahasa pemformatan mainframe, seperti IBM Advanced Function Presentation (AFP) atau Xerox Line Condition Data Stream (LCDS). Untuk informasi selengkapnya tentang penggunaan Micro Focus untuk migrasi dan modernisasi aplikasi mainframe di AWS, lihat Memberdayakan Beban Kerja Mainframe Perusahaan di AWS dengan Fokus Mikro dalam dokumentasi AWS.

Arsitektur infrastruktur AWS

Diagram berikut menunjukkan arsitektur infrastruktur AWS yang sangat tersedia dan aman untuk beban kerja pencetakan online mainframe:

Diagram menunjukkan alur kerja berikut:

1. Aplikasi online mainframe (ditulis pada bahasa pemrograman seperti CICS atau COBOL) menggunakan logika bisnis inti untuk memproses dan menghasilkan hasil cetak, seperti laporan

penagihan, kartu identitas, dan laporan pinjaman. Aplikasi online ini digunakan di Amazon EC2 di dua Availability Zones (AZ) untuk ketersediaan tinggi (HA) dan menggunakan Micro Focus CICS Print Exit untuk merutekan output cetak ke LRS VPSX/MFI untuk pencetakan pengguna akhir.

2. LRS VPSX/MFI menggunakan TCP/IP-based LRS/Queue transmission agent to collect or capture print data from the Micro Focus online Print Exit programming interface. Online Print Exit passes the necessary information to enable LRS VPSX/MFI to effectively process the print file and dynamically build LRS/Queue perintah.

> **Note**
>
> Untuk informasi lebih lanjut tentang berbagai metode pemrograman aplikasi CICS untuk pencetakan dan bagaimana mereka didukung di server Micro Focus Enterprise dan LRS VPSX/MFI, lihat Mencetak pengambilan data di bagian Informasi tambahan dari pola ini.

3.

> **Note**
>
> Network Load Balancer menyediakan nama DNS untuk mengintegrasikan Micro Focus Enterprise Server dengan LRS VPSX/MFI. : LRS VPSX/MFI supports a Layer 4 load balancer. The Network Load Balancer also does a basic health check on LRS VPSX/MFI dan mengarahkan lalu lintas ke target terdaftar yang sehat.

4. Server VPSX/MFI cetak LRS digunakan EC2 di Amazon di dua Availability Zone untuk HA dan menggunakan Amazon EBS sebagai penyimpanan data operasional. LRS VPSX/MFI mendukung mode layanan aktif-aktif dan aktif-pasif. Arsitektur ini menggunakan beberapa Availability Zone dalam pasangan aktif-pasif sebagai siaga aktif dan panas. Network Load Balancer melakukan pemeriksaan kesehatan pada instans LRS dan mengarahkan lalu lintas ke VPSX/MFI EC2 instans siaga panas di Availability Zone lain jika instans aktif dalam keadaan tidak sehat. Permintaan cetak disimpan dalam Antrian Job LRS secara lokal di setiap instance. EC2 Jika terjadi pemulihan, instance yang gagal harus dimulai ulang agar layanan LRS dapat melanjutkan pemrosesan permintaan cetak.

> **Note**
>
> LRS juga VPSX/MFI dapat melakukan pemeriksaan kesehatan di tingkat armada printer. Untuk informasi selengkapnya, lihat Pemeriksaan kesehatan armada printer di bagian Informasi tambahan dari pola ini.

5. AWS Managed Microsoft AD terintegrasi dengan LRS/DIS untuk melakukan autentikasi dan otorisasi alur kerja cetak. Untuk informasi selengkapnya, lihat Mencetak otentikasi dan otorisasi di bagian Informasi tambahan dari pola ini.
6. LRS VPSX/MFI menggunakan Amazon EBS untuk penyimpanan blok. Anda dapat mencadangkan data Amazon EBS dari EC2 instans aktif ke Amazon S3 point-in-time sebagai snapshot dan mengembalikannya ke volume EBS siaga panas. Untuk mengotomatiskan pembuatan, penyimpanan, dan penghapusan snapshot volume Amazon EBS, Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengatur frekuensi snapshot otomatis dan memulihkannya berdasarkan persyaratan RTO/RPO Anda.

Alat

Layanan AWS

- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon. EC2 Volume EBS berfungsi seperti perangkat blok mentah yang tidak terformat. Anda dapat memasang volume ini sebagai perangkat di instans Anda.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- AWS Directory Service untuk Microsoft Active Directory (AD), juga dikenal sebagai AWS Managed Microsoft Active Directory, memungkinkan beban kerja sadar direktori dan sumber daya AWS Anda untuk menggunakan Active Directory terkelola di AWS.

Alat-alat lainnya

- LRS VPSX/MFI (Micro Focus Interface), dikembangkan bersama oleh LRS dan Micro Focus, menangkap output dari spool Micro Focus Enterprise Server JES dan andal mengirimkannya ke tujuan cetak tertentu.
- LRS Directory Information Server (LRS/DIS) digunakan untuk otentikasi dan otorisasi selama alur kerja cetak.
- LRS/Queue is a TCP/IP-based LRS/Queue transmission agent, used by LRS VPSX/MFI, untuk mengumpulkan atau menangkap data cetak melalui antarmuka pemrograman Cetak Exit online Micro Focus.

- Micro Focus Enterprise Server adalah lingkungan penyebaran aplikasi untuk aplikasi mainframe. Ini menyediakan lingkungan eksekusi untuk aplikasi mainframe yang dimigrasikan atau dibuat dengan menggunakan versi Micro Focus Enterprise Developer.

Epik

Siapkan Micro Focus Enterprise Server di Amazon EC2 dan gunakan aplikasi online mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan Server Perusahaan Fokus Mikro dan gunakan aplikasi online demo.	Siapkan Server Perusahaan Fokus Mikro di Amazon EC2, lalu gunakan aplikasi Demo Akun Fokus Mikro (Demo ACCT) di Amazon EC2 dengan mengikuti petunjuk dari Tutorial: Dukungan CICS dalam dokumentasi Fokus Mikro. Aplikasi ACCT Demo adalah aplikasi mainframe online (CICS) yang membuat dan kemudian memulai output cetak.	Arsitek awan

Siapkan server cetak LRS di Amazon EC2

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan lisensi produk LRS untuk pencetakan.	Untuk mendapatkan lisensi produk LRS untuk LRSVPSX/MFI, LRS/Queue, and LRS/DIS, hubungi tim Manajemen Output LRS. Anda harus memberikan nama host dari	Membangun memimpin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	EC2 contoh di mana produk LRS akan diinstal.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat instance Amazon EC2 Windows untuk menginstal LRS VPSX/MFI.	Luncurkan instans Amazon EC2 Windows dengan mengikuti petunjuk dari Langkah 1: Luncurkan instance di EC2 dokumenta si Amazon. Instans Anda harus memenuhi persyarat an perangkat keras dan perangkat lunak berikut untuk LRS VPSX/LKM: • CPU - Inti Ganda • RAM — 16 GB • Penggerak - 500 GB • EC2 Contoh minimum - m5.xlarge • OS - Windows/Linux • Perangkat Lunak — Layanan Informasi Internet (IIS) atau Apache **Note** Persyaratan perangkat keras dan perangkat lunak sebelumny a ditujukan untuk armada printer kecil (sekitar 500-1000). Untuk mendapatkan persyaratan lengkap, konsultasikan dengan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kontak LRS dan AWS Anda. Saat Anda membuat instance Windows Anda, lakukan hal berikut: 1. Konfirmasikan bahwa nama EC2 host adalah nama host yang sama yang digunakan untuk lisensi produk LRS. 2. Aktifkan CGI di Amazon EC2 dengan menyelesaikan yang berikut ini: a. Connect ke EC2 instans Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. b. Di menu Start Windows, temukan dan buka Server Manager. c. Di Server Manager, pilih Dasbor, Mulai Cepat, Tambahkan peran dan fitur. Kemudian, pilih Peran server. d. Dalam peran Server, pilih WebServer (IIS), dan kemudian pilih Pengembangan Aplikasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	e. Dalam Pengembangan Aplikasi, pilih kotak centang CGI. f. Ikuti petunjuk pada Windows Server Manager Tambahkan peran dan fitur wizard untuk menginstal CGI. g. Buka port 5500 di firewall Windows EC2 instance untuk LRS/Queue komunikasi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS VPSX/MFI pada EC2 instance.	1. Connect ke EC2 instans Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. **Note** Buka tautan ke halaman unduhan produk dari email LRS yang seharusnya Anda terima. : Produk LRS didistribusikan melalui transfer file elektronik (EFT). 3. Unduh LRS VPSX/MFI dan unzip file (folder default:`c:\LRS`). 4. Luncurkan Pemasang Produk LRS dari folder yang tidak di-zip untuk menginstal LRS VPSX/MFI. 5. Pada menu Select Features, pilih VPSX® Server (V1R3.022), lalu pilih Next untuk memulai proses instalasi. Anda akan menerima pesan sukses ketika instalasi selesai.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS/antrian.	1. Connect ke EC2 instans Micro Focus Enterpris e Server Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka tautan ke halaman unduhan produk LRS dari email LRS yang seharusny a Anda terima, unduh LRS/ antrian, lalu unzip file. 3. Pergi ke lokasi di mana Anda men-download file, dan kemudian meluncurkan installer produk LRS untuk menginstal LRS/antrian.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal LRS/DIS.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka tautan ke halaman unduhan produk LRS dari email LRS yang seharusnya Anda terima, unduh LRS/DIS, lalu unzip file. 3. Pergi ke lokasi di mana Anda men-download file, dan kemudian meluncurkan LRS Product Installer. 4. Di LRS Product Installer, perluas LRS Misc Tools, pilih LRS DIS, lalu pilih Next. 5. Ikuti petunjuk lainnya di Pemasang Produk LRS untuk menyelesaikan proses instalasi.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup target dan daftarkan LRS VPSX/MFI EC2 sebagai target.	Buat grup target dengan mengikuti petunjuk dari Buat grup target untuk Network Load Balancer Anda dalam dokumentasi Elastic Load Balancing. Saat Anda membuat grup target, lakukan hal berikut: 1. Pada halaman Tentukan detail grup, untuk Pilih Jenis Target, pilih Contoh. 2. Untuk Protokol, pilih TCP. 3. Untuk Port, pilih 5500. 4. Pada halaman Daftar target, di bagian Instance yang tersedia, pilih instans LRS VPSX/MFI EC2 .	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat Network Load Balancer.	Ikuti instruksi dari Buat Network Load Balancer dalam dokumentasi Elastic Load Balancing. Network Load Balancer Anda merutekan lalu lintas dari Micro Focus Enterprise Server ke LRS. VPSX/MFI EC2 Saat Anda membuat Network Load Balancer, lakukan hal berikut pada halaman Listeners dan Routing: 1. Untuk Protokol, pilih TCP. 2. Untuk Port, pilih 5500. 3. Untuk tindakan Default, pilih Teruskan ke grup target yang Anda buat sebelumnya.	Arsitek awan

Integrasikan Micro Focus Enterprise Server dengan LRS VPSX/MFI dan LRS/Antrian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan Server Perusahaan Fokus Mikro untuk LRS/Queue integrasi.	1. Connect ke EC2 instans Micro Focus Enterprise Server Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di menu Start Windows, buka UI Administrasi Server Perusahaan Fokus Mikro. 3. Di bilah menu, pilih NATIVE. 4. Di panel navigasi, pilih Server Direktori, lalu pilih BANKDEMO atau wilayah server Enterprise Anda. 5. Dari Umum di panel navigasi kiri, gulir ke bawah ke bagian Tambahan untuk mengonfigurasi variabel lingkungan (LRSQ_ADDRESS, LRSQ_PORT, LRSQ_COMMAND) untuk menunjuk ke LRSQ. 6. Untuk LRSQ_ADDRESS, masukkan alamat IP atau nama DNS Network Load Balancer yang Anda buat sebelumnya. 7. Untuk LRSQ_PORT, masukkan VPSX LRSQ Listener Port (5500). 8. Untuk LRSQ_COMMAND, masukkan lokasi jalur LRSQ yang dapat dieksekusi. 9. **Note** LRS saat ini mendukung batas	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	karakter maksimum 50 untuk nama DNS, tetapi ini dapat berubah di masa mendatang . Jika nama DNS Anda lebih besar dari 50, maka Anda dapat menggunak an alamat IP Network Load Balancer sebagai alternatif.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jadikan CICS Print Exit (DFHUPRNT) tersedia untuk inisialisasi Micro Focus Enterprise Server.	1. Connect ke EC2 instans Micro Focus Enterprise Server Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. **Note** Salin CICS Print Exit (DFHUPRNT) dari folder VPSX/MFI executable LRS (bernama`VPSX_MFI_R2`) ke lokasi instance Micro Focus Enterprise Server. EC2 Untuk sistem 32 bit, lokasinya adalah`C:\Program Files (x86)\Micro Focus\Enterprise Server\bin`. Untuk sistem 64 bit, lokasinya adalah`C:\Program Files (x86)\Micro Focus\Enterprise Server\bin64`.: `DFHUPRNT_`	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`64.dll` File harus diganti namanya menjadi `DFHUPRNT.dll` saat disalin. Validasi bahwa Micro Focus Enterprise Server telah mendeteksi CICS Print Exit (DFHUPRNT) 1. Berhenti dan mulai Micro Focus Enterprise Server. 2. Di panel Administrasi Server Perusahaan Fokus Mikro, buka Monitor, Log, Konsol log. 3. Periksa log konsol untuk pesan berikut: “3270 pengguna printer keluar DFHUPRNT berhasil diinstal.”	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan ID terminal printer CICS (TERMIDs) sebagai Server Perusahaan Fokus Mikro.	Aktifkan pencetakan 3270 di Micro Focus Enterprise Server 1. Di panel Administrasi Server Perusahaan Fokus Mikro, buka CICS, Sumber Daya, Berdasarkan Grup. 2. Dari panel navigasi kiri, pilih SIT (Tabel Inisialis asi Sistem), lalu pilih BNKCICV. 3. Di bagian Umum, gulir ke bawah ke 3270, lalu pilih kotak centang 3270 Cetak. Tentukan terminal printer CICS di Micro Focus Enterpris e Server 1. Di panel Administrasi Server Perusahaan Fokus Mikro, buka CICS, Sumber Daya, Berdasarkan Jenis. 2. Dari panel navigasi kiri, pilih Istilah, lalu pilih Baru. Formulir Create Terminal Resource terbuka. 3. Untuk Nama, masukkan nama LRS Print Queue. (Catatan: Pola ini menggunakan “P275" sebagai ID terminal printer CICS dan Antrian Cetak LRS VPSX.)	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk Grup, masukkan BANKTERM. 5. Untuk Instal Otomatis — Model, masukkan NO. 6. Untuk Pengidentifikasi Terminal - Jenis terminal, masukkan DFHPRT32. 7. Untuk nama Net, masukkan VTAMP275. 8. Untuk Penggunaan Terminal, pilih kotak centang Dalam Layanan. 9. Gulir di bagian atas halaman, lalu pilih Simpan. 10. Pilih Instal. Pesan pop-up menampilkan pesan instalasi yang berhasil.	

Siapkan printer dan cetak pengguna di Micro Focus Enterprise Server dan LRS VPSX/MFI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat antrian cetak di LRS VPSX.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka Antarmuka Web VPSX dari menu Start Windows. 3. Di panel navigasi, pilih Printer.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Pilih Tambah, lalu pilih Tambah Printer. 5. Pada halaman Konfigurasi Printer, untuk Nama Printer, masukkan P275. 6. Untuk ID VPSX, masukkan. VPS1 7. Untuk CommType, pilih TCPIP/LRSQ. 8. Untuk Host/IP Address, masukkan alamat IP printer fisik yang ingin Anda tambahkan. 9. Untuk Perangkat, masukkan nama perangkat Anda. 10. Pilih Driver Windows atau Driver Linux/Mac. 11. Pilih Tambahkan. **Note** Antrian cetak harus setara dengan Cetak yang TERMIDs dibuat di Micro Focus Enterprise Server.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna cetak di LRS VPSX/LKM.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Buka Antarmuka Web VPSX dari menu Start Windows. 3. Di panel navigasi, pilih Keamanan, lalu pilih Pengguna. 4. Di kolom Nama Pengguna, pilih admin, lalu pilih Salin. 5. Di jendela Pemeliharaan Profil Pengguna, untuk Nama Pengguna, masukkan nama pengguna (misalnya, PrintUser). 6. Untuk Deskripsi, masukkan deskripsi singkat (misalnya, Pengguna untuk tes cetak). 7. Pilih Perbarui. Ini menciptakan pengguna cetak (misalnya, PrintUser). 8. Di panel navigasi, di bawah Pengguna, pilih pengguna baru yang Anda buat. 9. Dari menu Command, pilih Security. 10. Pada halaman Aturan Keamanan, pilih semua opsi keamanan printer dan	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	keamanan pekerjaan yang berlaku, lalu pilih Simpan. 11. Untuk menambahkan pengguna cetak baru Anda ke grup Administrator, buka panel navigasi, pilih Keamanan, lalu pilih Konfigurasi. 12. Di jendela konfigurasi Keamanan, tambahkan pengguna cetak baru Anda ke kolom Administrator.	

Mengatur otentikasi cetak dan otorisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat domain AWS Managed Microsoft AD dengan pengguna dan grup.	1. Buat Active Directory di AWS Managed Microsoft AD dengan mengikuti petunjuk dari Buat direktori AWS Managed Microsoft AD Anda di dokumentasi AWS Directory Service. 2. Menerapkan EC2 instans (Active Directory manager) dan instal alat Active Directory untuk mengelola AWS Managed Microsoft AD Anda dengan mengikuti petunjuk dari Langkah 3: Menerapkan EC2 instans untuk mengelola AWS Managed Microsoft AD	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda dalam dokumentasi AWS Directory Service. 3. **Note** Connect ke EC2 instans Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. : Saat Anda terhubung ke EC2 instance, masukkan kredensi administrator Anda (untuk direktori yang Anda buat pada langkah pertama) di jendela Windows Security. 4. Di menu Start Windows, di bawah Windows Administrative Tools, pilih Active Directory Users and Computers. 5. Buat pengguna cetak di domain Active Directory dengan mengikuti langkah-langkah dari Buat pengguna di dokumentasi layanan AWS Directory.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bergabunglah dengan LRS VPSX/MFI EC2 ke domain AWS Managed Microsoft AD.	Bergabunglah dengan LRS VPSX/MFI EC2 ke domain AWS Managed Microsoft AD Anda secara otomatis (dokumentasi AWS Knowledge Center) atau secara manual (dokumentasi AWS Directory Service).	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan integrasikan LRS/DIS dengan AWS Managed Microsoft AD.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Di menu Start Windows, buka Antarmuka Web VPSX. 3. Di panel navigasi, pilih Keamanan, lalu pilih Konfigurasi. 4. Pada halaman Konfigurasi Keamanan, di bagian Parameter Keamanan, untuk Jenis Keamanan, pilih Internal. 5. Masukkan preferensi Anda untuk sisa opsi di bagian Parameter Keamanan. 6. Buka folder LRS Output Management dari menu Microsoft Windows Start, pilih Server Start, lalu pilih Server Stop. 7. Masuk ke LRS VPSX/MFI dengan nama pengguna dan kata sandi Direktori Aktif Anda.	Arsitek awan

Uji alur kerja cetak online

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Memulai permintaan cetak online dari aplikasi Demo Micro Focus ACCT.	1. **Note** Buka emulator terminal TN327 0 di EC2 instance Micro Focus Enterpris e Server Anda. (: Pola ini menggunak an 3270 emulator terminal.) 2. Connect ke emulator terminal TN327 0 (Rumba). Untuk alamat nama Host, gunakan 127.0.0.1. Untuk Telnet Port, gunakan 9270. 3. Setelah menghubungkan ke layar 3270, tekan CTL+SHIFT+Z untuk menghapus layar. 4. Untuk memulai aplikasi Demo ACCT, di layar yang jelas, masukkan ACCT. Ini membuka layar utama aplikasi ACCT Demo online (CICS). Catatan: Layar utama mencakup opsi menu seperti File akun, Untuk mencari berdasark an nama, masukkan, Jenis permintaan, Akun, dan Printer.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Untuk mengirimkan permintaan cetak dari aplikasi ACCT Demo online (CICS), masukkan P di kolom tipe permintaan, 11111 di bidang akun, dan P275 di bidang printer. Pastikan untuk mengatur nilai di bidang printer ke nilai ID terminal printer CICS. 6. Tekan Enter. Pesan “Permintaan Cetak Terjadwal” muncul di bagian bawah layar. Ini menegaskan bahwa permintaan cetak online dihasilkan dari aplikasi Demo ACCT dan dikirim ke LRS VPS/MFI untuk pemrosesan cetak.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa output cetak di LRS VPSX/MFI.	1. Connect ke VPSX/MFI EC2 instans LRS Anda dengan mengikuti petunjuk dari Langkah 2: Connect ke instans Anda dalam EC2 dokumentasi Amazon. 2. Di menu Start Windows, buka Antarmuka Web VPSX. 3. Di panel navigasi, pilih Printer, lalu pilih Output Queue. Temukan antrian cetak P275 yang Anda buat untuk pencetakan online sebelumnya. 4. Untuk antrian cetak (P275), di kolom ID Spool, pilih ID spool untuk permintaan dalam antrian printer. 5. Pada tab Actions, di kolom COMMAND, pilih Browse. Anda sekarang dapat melihat hasil cetak dari laporan akun dengan kolom untuk Nomor Akun, NAMA KELUARGA, PERTAMA, ALAMAT, TELEPON, No. Kartu yang Diterbitkan, Tanggal dikeluarkan, Jumlah, dan Saldo.	Insinyur uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Sebagai contoh, lihat lampiran online_print_output untuk pola ini.	

Sumber daya terkait

- Modernisasi Output LRS (dokumentasi LRS)
- Konsep jaringan VTAM (dokumentasi IBM)
- Ringkasan tipe unit logis (LU) (dokumentasi IBM)
- ANSI dan kontrol kereta mesin (dokumentasi IBM)
- Memberdayakan Beban Kerja Mainframe Perusahaan di AWS dengan Fokus Mikro (AWS Partner Network Blog)
- Membangun PAC Server Perusahaan Fokus Mikro dengan Amazon EC2 Auto Scaling dan Systems Manager (dokumentasi AWS Prescriptive Guidance)
- Aliran data Presentasi Fungsi Lanjutan (AFP) (dokumentasi IBM)
- Aliran Data Terkondisi Jalur (LCD) (Dokumentasi Compart)

Informasi tambahan

Pertimbangan-pertimbangan

Selama perjalanan modernisasi Anda, Anda dapat mempertimbangkan berbagai konfigurasi untuk proses online mainframe dan output yang dihasilkannya. Platform mainframe telah disesuaikan oleh setiap pelanggan dan vendor yang menggunakannya dengan persyaratan khusus yang secara langsung memengaruhi pencetakan. Misalnya, platform Anda saat ini dapat menggabungkan IBM Advanced Function Presentation (AFP) atau Xerox Line Condition Data Stream (LCDS) ke dalam alur kerja saat ini. Selain itu, karakter kontrol carriage mainframe dan kata-kata perintah saluran dapat memengaruhi tampilan halaman yang dicetak dan mungkin memerlukan penanganan khusus. Sebagai bagian dari proses perencanaan modernisasi, kami menyarankan Anda menilai dan memahami konfigurasi di lingkungan cetak spesifik Anda.

Cetak pengambilan data

Bagian ini merangkum metode pemrograman aplikasi CICS yang dapat Anda gunakan dalam lingkungan mainframe IBM untuk pencetakan. VPSX/MFI Komponen LRS menyediakan teknik untuk memungkinkan program aplikasi yang sama untuk membuat data dengan cara yang sama. Tabel berikut menjelaskan bagaimana setiap metode pemrograman aplikasi didukung dalam aplikasi CICS modern yang berjalan di AWS dan Micro Focus Enterprise Server dengan server cetak VPSX/MFI LRS.

Metode	Deskripsi	Support untuk metode dalam lingkungan modern
EXEC CICS MENGIRIM TEKS.. atau EXEC CICS KIRIM PETA..	Metode CICS dan VTAM ini bertanggung jawab untuk membuat dan mengirimkan aliran data cetak 3270/SCS ke LUTYPE0,, dan perangkat cetak. LUTYPE1 LUTYPE3	Antarmuka program aplikasi (API) Micro Focus online Print Exit (DFHUPRNT) memungkinkan data cetak diproses VPSX/MFI ketika aliran data cetak 3270/SCS dibuat dengan menggunakan salah satu metode ini.
EXEC CICS MENGIRIM TEKS.. atau EXEC CICS KIRIM PETA.. (dengan perangkat lunak mainframe IBM pihak ketiga)	Metode CICS dan VTAM bertanggung jawab untuk membuat dan mengirimk an aliran data cetak 3270/ SCS ke LUTYPE0,, dan perangkat cetak. LUTYPE1 LUTYPE3 Produk perangkat lunak pihak ketiga mencegat data cetak, mengonversi data menjadi data format cetak standar dengan karakter ASA/MCH kontrol, dan menempatkan data pada spool JES untuk diproses oleh sistem pencetakan berbasis mainframe yang menggunak an JES.	API Micro Focus online Print Exit (DFHUPRNT) memungkin kan data cetak diproses VPSX/MFI ketika aliran data cetak 3270/SCS dibuat dengan menggunakan salah satu metode ini.

EXEC CICS SPOOLOPEN	Metode ini digunakan oleh program aplikasi CICS untuk menulis data langsung ke spool JES. Data kemudian menjadi tersedia untuk diproses oleh sistem pencetakan berbasis mainframe yang menggunakan JES.	Micro Focus Enterprise Server menggulung data ke spool Enterprise Server di mana ia dapat diproses oleh VPSX/MFI Batch Print Exit (LRSPRTE6) yang menggulung data ke VPSX.
DRS/API	Antarmuka program yang disediakan LRS digunakan untuk menulis data cetak ke JES.	VPSX/MFI menyediakan antarmuka pengganti yang menggulung data cetak langsung ke VPSX.

Pemeriksaan kesehatan armada printer

LRS VPSX/MFI (LRS LoadX) dapat melakukan pemeriksaan kesehatan deep dive, termasuk manajemen perangkat dan optimalisasi operasional. Manajemen perangkat dapat mendeteksi kegagalan pada perangkat printer dan merutekan permintaan cetak ke printer yang sehat. Untuk informasi selengkapnya tentang pemeriksaan kesehatan deep dive untuk armada printer, lihat dokumentasi LRS yang disertakan dengan lisensi produk Anda.

Cetak otentikasi dan otorisasi

LRS/DIS enables LRS applications to authenticate user IDs and passwords by using Microsoft Active Directory or an LDAP server. In addition to basic print authorization, LRS/DISjuga dapat menerapkan kontrol keamanan cetak tingkat butiran dalam kasus penggunaan berikut:

- Kelola siapa yang dapat menelusuri pekerjaan printer.
- Kelola tingkat penelusuran pekerjaan pengguna lain.
- Kelola tugas operasional. Misalnya, keamanan tingkat perintah seperti hold/release, purge, modify, copy, dan reroute. Keamanan dapat diatur oleh User-ID atau Grup (mirip dengan grup AD atau grup LDAP).

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Pindahkan file mainframe langsung ke Amazon S3 menggunakan Transfer Family

Luis Gustavo Dantas, Amazon Web Services

Ringkasan

Sebagai bagian dari perjalanan modernisasi, Anda dapat menghadapi tantangan untuk mentransfer file antara server lokal dan Amazon Web Services (AWS) Cloud. Mentransfer data dari mainframe dapat menjadi tantangan yang signifikan karena mainframe biasanya tidak dapat mengakses penyimpanan data modern seperti Amazon Simple Storage Service (Amazon S3), Amazon Elastic Block Store (Amazon EBS), atau Amazon Elastic File System (Amazon EFS).

Banyak pelanggan menggunakan sumber daya pementasan menengah, seperti server Linux, Unix, atau Windows lokal, untuk mentransfer file ke AWS Cloud. Anda dapat menghindari metode tidak langsung ini dengan menggunakan AWS Transfer Family dengan Secure Shell (SSH) File Transfer Protocol (SFTP) untuk mengunggah file mainframe langsung ke Amazon S3.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Virtual Private Cloud (VPC) dengan subnet yang dapat dijangkau oleh platform lama Anda
- Endpoint Transfer Family untuk VPC Anda
- File Metode Akses Penyimpanan Virtual Mainframe (VSAM) yang dikonversi ke file berurutan dengan panjang tetap (dokumentasi IBM)

Batasan

- SFTP mentransfer file dalam mode biner secara default, yang berarti bahwa file diunggah ke Amazon S3 dengan encoding EBCDIC dipertahankan. Jika file Anda tidak berisi data biner atau dikemas, maka Anda dapat menggunakan subperintah sftp ascii (dokumentasi IBM) untuk mengonversi file Anda menjadi teks selama transfer.
- Anda harus membongkar file mainframe (AWS Prescriptive Guidance) yang berisi konten yang dikemas dan biner untuk menggunakan file-file ini di lingkungan target Anda.
- Objek Amazon S3 dapat berkisar dalam ukuran dari minimal 0 byte hingga maksimum 5 TB. Untuk informasi selengkapnya tentang kemampuan Amazon S3, lihat Amazon S3. FAQs

Arsitektur

Tumpukan teknologi sumber

- Bahasa kontrol pekerjaan (JCL)
- z/OS Unix shell dan ISPF
- SFTP
- VSAM dan file datar

Tumpukan teknologi target

- Transfer Family
- Amazon S3
- Amazon Virtual Private Cloud (Amazon VPC)

Arsitektur target

Diagram berikut menunjukkan arsitektur referensi untuk menggunakan Transfer Family dengan SFTP untuk mengunggah file mainframe langsung ke bucket S3.

Diagram menunjukkan alur kerja berikut:

1. Anda menggunakan pekerjaan JCL untuk mentransfer file mainframe Anda dari mainframe lama ke AWS Cloud melalui Direct Connect.
2. Direct Connect memungkinkan lalu lintas jaringan Anda tetap berada di jaringan global AWS dan melewati internet publik. Direct Connect juga meningkatkan kecepatan jaringan, mulai dari 50 Mbps dan skala hingga 100 Gbps.
3. Titik akhir VPC memungkinkan koneksi antara sumber daya VPC Anda dan layanan yang didukung tanpa menggunakan internet publik. Akses ke Transfer Family dan Amazon S3 mencapai ketersediaan tinggi dengan berlangsung melalui antarmuka jaringan elastis yang terletak di dua subnet pribadi dan Availability Zone.
4. Transfer Family mengautentikasi pengguna dan menggunakan SFTP untuk menerima file Anda dari lingkungan lama dan memindahkannya ke bucket S3.

Otomatisasi dan skala

Setelah layanan Transfer Family tersedia, Anda dapat mentransfer file dalam jumlah tak terbatas dari mainframe ke Amazon S3 dengan menggunakan pekerjaan JCL sebagai klien SFTP. Anda juga dapat mengotomatiskan transfer file dengan menggunakan penjadwal pekerjaan batch mainframe untuk menjalankan pekerjaan SFTP ketika Anda siap untuk mentransfer file mainframe.

Alat

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.
- AWS Transfer Family memungkinkan Anda menskalakan transfer business-to-business file berulang dengan aman ke Amazon S3 dan Amazon EFS dengan menggunakan protokol SFTP, FTPS, dan FTP.

Epik

Buat bucket S3 dan kebijakan akses

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket S3 untuk meng-host file yang Anda transfer dari lingkungan lama Anda.	AWS Umum
Buat peran dan kebijakan IAM.	Transfer Family menggunakan peran AWS Identity and Access Management (IAM) untuk memberikan akses ke bucket S3 yang Anda buat sebelumnya. Buat peran IAM yang mencakup kebijakan IAM berikut:	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	{ "Version": "2012-10-17", "Statement": [{ "Sid": "UserFolderListing", "Action": ["s3:ListBucket", "s3:GetBucketLocat ion"], "Effect": "Allow", "Resource": ["arn:aws:s3:::<your- bucket-name>"] }, { "Sid": "HomeDirObjectAcce ss", "Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObjectAcl", "s3:GetObject", "s3:DeleteObjectVe rsion", "s3:DeleteObject",	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"s3:PutObjectAcl",
	
	"s3:GetObjectVersion"
	],
	"Resource":
	"arn:aws:s3:::<your-bucket-name>/*"
	}
	]
	}</pre>
	Note
	Anda harus memilih kasus penggunaan Transfer saat Anda membuat peran IAM.	

Tentukan layanan transfer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server SFTP.	1. Masuk ke konsol AWS Management, buka konsol Transfer Family, lalu pilih Buat server. 2. Pilih hanya SFTP (SSH File Transfer Protocol) - transfer file melalui protokol Secure Shell dan kemudian pilih Berikutnya. 3. Untuk penyedia Identitas, pilih Layanan dikelola dan kemudian pilih Berikutnya.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk tipe Endpoint, pilih VPC yang dihosting. 5. Untuk Akses, pilih Internal. 6. Untuk VPC, pilih VPC Anda. 7. Di bagian Availability Zones, pilih Availability Zones dan subnet Anda. 8. Di bagian Grup Keamanan, pilih grup keamanan Anda, lalu pilih Berikutnya. 9. Untuk Domain, pilih Amazon S3 lalu pilih Berikutnya. 10. Tinggalkan opsi default pada halaman Konfigurasi detail tambahan dan kemudian pilih Berikutnya. 11. Pilih Buat server. **Note** Untuk informasi selengkapnya tentang cara menyiapkan server SFTP, lihat Membuat server berkemampuan SFTP (Panduan Pengguna AWS Transfer Family).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Dapatkan alamat server.	1. Buka konsol Transfer Family dan pilih ID server Anda di kolom Server ID. 2. Di bagian Detail titik akhir, untuk tipe Endpoint, pilih ID endpoint. Ini membawa Anda ke konsol VPC Amazon. 3. Pada tab Detail konsol VPC Amazon, temukan nama DNS di sebelah nama DNS.	AWS Umum
Buat key pair klien SFTP.	Buat key pair SSH untuk Microsoft Windows atau macOS/Linux/UNIX.	AWS Umum, SSH

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pengguna SFTP.	1. Buka konsol Transfer Family, pilih Server dari panel navigasi, lalu pilih server Anda. 2. Di kolom ID Server, pilih ID server untuk server Anda dan kemudian pilih Tambah pengguna. 3. Untuk Username, masukkan nama pengguna yang cocok dengan nama pengguna key pair SSH Anda. 4. Untuk Peran, pilih peran IAM yang Anda buat sebelumnya. 5. Untuk direktori Home, pilih bucket S3 yang Anda buat sebelumnya. 6. Untuk kunci publik SSH, masukkan key pair yang Anda buat sebelumnya. 7. Pilih Tambahkan.	AWS Umum

Transfer file mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim kunci pribadi SSH ke mainframe.	Gunakan SFTP atau SCP untuk mengirim kunci pribadi SSH ke lingkungan lama. Contoh SFTP:	Mainframe, shell z/OS Unix, FTP, SCP

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`sftp [USERNAME@mainframeIP]`
	`[password]`
	`cd [/u/USERNAME]`
	`put [your-key-pair-file]`
	
	Contoh SCP:
	
	`scp [your-key-pair-file] [USERNAME@MainframeIP]:/[u/USERNAME]`
	
	Selanjutnya, simpan kunci SSH dalam sistem file z/OS Unix di bawah nama pengguna yang nantinya akan menjalankan tugas batch transfer file (misalnya,`/u/CONTROLM`).
	
	Note
	Untuk informasi selengkapnya tentang z/OS Unix shell, lihat Pengantar z/OS shell (dokumentasi IBM).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat klien JCL SFTP.	Karena mainframe tidak memiliki klien SFTP asli, Anda harus menggunakan utilitas BPXBATCH untuk menjalankan klien SFTP dari shell Unix. z/OS
	
	Di editor ISPF, buat klien JCL SFTP. Contoh:
	
	`//JOBNAM JOB ...`
	`//************`
	`**********`
	`**********`
	`**********`
	`****`
	`//SFTP EXEC PGM=BPXBATCH,REGION=0M`
	`//STDPARM DD *`
	`SH cp "//'MAINFRAME.FILE.NAME'" filename.txt;`
	`echo 'put filename.txt' > uplcmd;`
	`sftp -b uplcmd -i ssh_private_key_file ssh_username@<transfer service ip or DNS>;`
	`//SYSPRINT DD SYSOUT=*`
	`//STDOUT DD SYSOUT=*`
	`//STDENV DD *`
	`//STDERR DD SYSOUT=*`
	
	Note
	Untuk informasi selengkapnya tentang cara menjalankan
		JCL, Mainframe, z/OS cangkang Unix

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	perintah di shell z/OS Unix, lihat Utilitas BPXBATCH (dokumentasi IBM). Untuk informasi selengkapnya tentang cara membuat atau mengedit pekerjaan JCL di z/OS, lihat Apa itu ISPF? dan Editor ISPF (dokumentasi IBM).	
Jalankan klien JCL SFTP.	1. Di editor ISPF, masukkan SUB, lalu tekan tombol ENTER setelah pekerjaan JCL dibuat. 2. Pantau aktivitas pekerjaan batch transfer file mainframe di SDSF. **Note** Untuk informasi selengkapnya tentang cara memeriksa aktivitas pekerjaan batch, lihat Panduan Pengguna z/OS SDSF (dokumentasi IBM).	Mainframe, JCL, ISPF

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi transfer file.	1. Masuk ke konsol AWS Management, buka konsol Amazon S3, lalu pilih Bucket dari panel navigasi. 2. Pilih ember yang terkait dengan Transfer Family Anda. 3. Di bagian Objek pada tab Objek, temukan file yang Anda transfer dari mainframe.	AWS Umum
Otomatiskan klien JCL SFTP.	Gunakan penjadwal pekerjaan untuk secara otomatis memicu klien JCL SFTP. **Note** Anda dapat menggunakan penjadwal pekerjaan mainframe, seperti BMC Control-M atau CA Workload Automation, untuk mengotomatiskan pekerjaan batch untuk transfer file berdasarkan waktu dan dependens i pekerjaan batch lainnya.	Job scheduler

Sumber daya terkait

- Cara kerja AWS Transfer Family

Optimalkan kinerja aplikasi modern AWS Blu Age Anda

Vishal Jaswani, Manish Roy, dan Himanshu Sah, Amazon Web Services

Ringkasan

Aplikasi mainframe yang dimodernisasi dengan AWS Blu Age memerlukan pengujian fungsional dan kesetaraan kinerja sebelum diterapkan ke produksi. Dalam pengujian kinerja, aplikasi modern dapat bekerja lebih lambat daripada sistem lama, terutama dalam pekerjaan batch yang kompleks. Perbedaan ini ada karena aplikasi mainframe bersifat monolitik, sedangkan aplikasi modern menggunakan arsitektur multitier. Pola ini menyajikan teknik pengoptimalan untuk mengatasi kesenjangan kinerja ini untuk aplikasi yang dimodernisasi dengan menggunakan refactoring otomatis dengan Blu Age. AWS

Pola ini menggunakan kerangka modernisasi AWS Blu Age dengan Java asli dan kemampuan penyetelan basis data untuk mengidentifikasi dan mengatasi kemacetan kinerja. Pola ini menjelaskan bagaimana Anda dapat menggunakan profil dan pemantauan untuk mengidentifikasi masalah kinerja dengan metrik seperti waktu eksekusi SQL, pemanfaatan memori, dan pola. I/O Kemudian menjelaskan bagaimana Anda dapat menerapkan pengoptimalan yang ditargetkan, termasuk restrukturisasi kueri database, caching, dan penyempurnaan logika bisnis.

Peningkatan waktu pemrosesan batch dan pemanfaatan sumber daya sistem membantu Anda mencocokkan tingkat kinerja mainframe dalam sistem modern Anda. Pendekatan ini mempertahankan kesetaraan fungsional selama transisi ke arsitektur berbasis cloud modern.

Untuk menggunakan pola ini, atur sistem Anda dan identifikasi hotspot kinerja dengan mengikuti instruksi di bagian Epics, dan terapkan teknik pengoptimalan yang dibahas secara rinci di bagian Arsitektur.

Prasyarat dan batasan

Prasyarat

- Aplikasi modern AWS Blu Age
- JProfiler Lisensi
- Hak istimewa administratif untuk menginstal klien database dan alat profil
- AWS Sertifikasi Blu Age Level 3
- Pemahaman tingkat menengah tentang kerangka AWS Blu Age, struktur kode yang dihasilkan, dan pemrograman Java

Batasan

Kemampuan dan fitur pengoptimalan berikut berada di luar cakupan pola ini:

- Optimalisasi latensi jaringan antara tingkatan aplikasi
- Pengoptimalan tingkat infrastruktur melalui jenis instans Amazon Elastic Compute Cloud (EC2Amazon) dan pengoptimalan penyimpanan
- Pengujian beban pengguna bersamaan dan pengujian stres

Versi produk

- JProfiler versi 13.0 atau yang lebih baru (kami sarankan versi terbaru)
- pgAdmin versi 8.14 atau yang lebih baru

Arsitektur

Pola ini mengatur lingkungan pembuatan profil untuk aplikasi AWS Blu Age dengan menggunakan alat seperti dan JProfiler pgAdmin. Ini mendukung optimasi melalui DAOManager dan SQLExecution Builder yang APIs disediakan oleh AWS Blu Age.

Sisa bagian ini memberikan informasi terperinci dan contoh untuk mengidentifikasi hotspot kinerja dan strategi pengoptimalan untuk aplikasi modern Anda. Langkah-langkah di bagian Epik merujuk kembali ke informasi ini untuk panduan lebih lanjut.

Mengidentifikasi hotspot kinerja dalam aplikasi mainframe modern

Dalam aplikasi mainframe modern, hotspot kinerja adalah area spesifik dalam kode yang menyebabkan perlambatan atau inefisiensi yang signifikan. Hotspot ini sering disebabkan oleh perbedaan arsitektur antara mainframe dan aplikasi modern. Untuk mengidentifikasi kemacetan kinerja ini dan mengoptimalkan kinerja aplikasi modern Anda, Anda dapat menggunakan tiga teknik: SQL logging, rencana kueri, dan analisis. `EXPLAIN` JProfiler

Teknik identifikasi hotspot: pencatatan SQL

Aplikasi Java modern, termasuk yang telah dimodernisasi dengan menggunakan AWS Blu Age, memiliki kemampuan bawaan untuk mencatat kueri SQL. Anda dapat mengaktifkan logger tertentu dalam proyek AWS Blu Age untuk melacak dan menganalisis pernyataan SQL yang dijalankan oleh aplikasi Anda. Teknik ini sangat berguna untuk mengidentifikasi pola akses database yang tidak

efisien, seperti permintaan individu yang berlebihan atau panggilan database yang tidak terstruktur dengan baik, yang dapat dioptimalkan melalui batching atau penyempurnaan kueri.

Untuk mengimplementasikan logging SQL di aplikasi modern AWS Blu Age Anda, atur level log ke untuk pernyataan SQL dalam `application.properties` file `DEBUG` untuk menangkap detail eksekusi kueri:

```
level.org.springframework.beans.factory.support.DefaultListableBeanFactory : WARN
level.com.netfective.bluage.gapwalk.runtime.sort.internal: WARN
level.org.springframework.jdbc.core.StatementCreatorUtils: DEBUG
level.com.netfective.bluage.gapwalk.rt.blu4iv.dao: DEBUG
level.com.fiserv.signature: DEBUG
level.com.netfective.bluage.gapwalk.database.support.central: DEBUG
level.com.netfective.bluage.gapwalk.rt.db.configuration.DatabaseConfiguration: DEBUG
level.com.netfective.bluage.gapwalk.rt.db.DatabaseInteractionLoggerUtils: DEBUG
level.com.netfective.bluage.gapwalk.database.support.AbstractDatabaseSupport: DEBUG
level.com.netfective.bluage.gapwalk.rt: DEBUG
```

Pantau kueri frekuensi tinggi dan berkinerja lambat dengan menggunakan data yang dicatat untuk mengidentifikasi target pengoptimalan. Fokus pada kueri dalam proses batch karena biasanya memiliki dampak kinerja tertinggi.

Teknik identifikasi hotspot: Kueri JELASKAN rencana

Metode ini menggunakan kemampuan perencanaan kueri sistem manajemen database relasional. Anda dapat menggunakan perintah seperti `EXPLAIN` di PostgreSQL atau MySQL, atau di Oracle`EXPLAIN PLAN`, untuk memeriksa bagaimana database Anda bermaksud menjalankan kueri yang diberikan. Output dari perintah ini memberikan wawasan berharga ke dalam strategi eksekusi kueri, termasuk apakah indeks akan digunakan atau pemindaian tabel lengkap akan dilakukan. Informasi ini sangat penting untuk mengoptimalkan kinerja kueri, terutama dalam kasus di mana pengindeksan yang tepat dapat secara signifikan mengurangi waktu eksekusi.

Ekstrak kueri SQL yang paling berulang dari log aplikasi dan analisis jalur eksekusi kueri berkinerja lambat dengan menggunakan `EXPLAIN` perintah yang khusus untuk database Anda. Berikut adalah contoh untuk database PostgreSQL.

Kueri:

```
SELECT * FROM tenk1 WHERE unique1 < 100;
```

`EXPLAIN`perintah:

```
EXPLAIN SELECT * FROM tenk1 where unique1 < 100;
```

Output:

```
Bitmap Heap Scan on tenk1 (cost=5.06..224.98 rows=100 width=244)
Recheck Cond: (unique1 < 100)
-> Bitmap Index Scan on tenk1_unique1 (cost=0.00..5.04 rows=100 width=0)
Index Cond: (unique1 < 100)
```

Anda dapat menafsirkan `EXPLAIN` output sebagai berikut:

- Baca `EXPLAIN` rencana dari operasi terdalam ke terluar (bawah ke atas).
- Carilah istilah-istilah kunci. Misalnya, `Seq Scan` menunjukkan pemindaian tabel lengkap dan `Index Scan` menunjukkan penggunaan indeks.
- Periksa nilai biaya: Angka pertama adalah biaya awal, dan angka kedua adalah total biaya.
- Lihat `rows` nilai untuk perkiraan jumlah baris output.

Dalam contoh ini, mesin kueri menggunakan pemindaian indeks untuk menemukan baris yang cocok, dan kemudian hanya mengambil baris tersebut (`Bitmap Heap Scan`). Ini lebih efisien daripada memindai seluruh tabel, meskipun biaya akses baris individu lebih tinggi.

Operasi pemindaian tabel dalam output `EXPLAIN` rencana menunjukkan indeks yang hilang. Optimalisasi membutuhkan pembuatan indeks yang sesuai.

Teknik identifikasi hotspot: analisis JProfiler

JProfiler adalah alat profil Java komprehensif yang membantu Anda mengatasi kemacetan kinerja dengan mengidentifikasi panggilan database yang lambat dan panggilan intensif CPU. Alat ini sangat efektif dalam mengidentifikasi kueri SQL yang lambat dan penggunaan memori yang tidak efisien.

Contoh analisis untuk kueri:

```
select evt.
 com.netfective.bluage.gapwalk.rt.blu4iv.dao.Blu4ivTableManager.queryNonTrasactional
```

Tampilan JProfiler Hot Spots memberikan informasi berikut:

- Kolom waktu
  - Menunjukkan total durasi eksekusi (misalnya, 329 detik)

    - Menampilkan persentase total waktu aplikasi (misalnya, 58,7%)
    - Membantu mengidentifikasi operasi yang paling memakan waktu
- Kolom Waktu Rata-rata
    - Menunjukkan durasi per eksekusi (misalnya, 2.692 mikrodetik)
    - Menunjukkan kinerja operasi individu
    - Membantu mendeteksi operasi individu yang lambat
- Kolom acara
    - Menunjukkan jumlah eksekusi (misalnya, 122,387 kali)
    - Menunjukkan frekuensi operasi
    - Membantu mengidentifikasi metode yang sering disebut

Untuk hasil contoh:

- Frekuensi tinggi: 122.387 eksekusi menunjukkan potensi pengoptimalan
- Masalah kinerja: 2.692 mikrodetik untuk waktu rata-rata menunjukkan inefisiensi
- Dampak kritis: 58,7% dari total waktu menunjukkan kemacetan utama

JProfiler dapat menganalisis perilaku runtime aplikasi Anda untuk mengungkapkan hotspot yang mungkin tidak terlihat melalui analisis kode statis atau pencatatan SQL. Metrik ini membantu Anda mengidentifikasi operasi yang memerlukan pengoptimalan dan menentukan strategi pengoptimalan yang paling efektif. Untuk informasi selengkapnya tentang JProfiler fitur, lihat JProfiler dokumentasi.

Bila Anda menggunakan ketiga teknik ini (SQL logging, query `EXPLAIN` plan, dan JProfiler) dalam kombinasi, Anda bisa mendapatkan tampilan holistik dari karakteristik kinerja aplikasi Anda. Dengan mengidentifikasi dan menangani hotspot kinerja yang paling penting, Anda dapat menjembatani kesenjangan kinerja antara aplikasi mainframe asli Anda dan sistem berbasis cloud modern Anda.

Setelah Anda mengidentifikasi hotspot kinerja aplikasi Anda, Anda dapat menerapkan strategi pengoptimalan, yang dijelaskan di bagian berikutnya.

Strategi optimasi untuk modernisasi mainframe

Bagian ini menguraikan strategi utama untuk mengoptimalkan aplikasi yang telah dimodernisasi dari sistem mainframe. Ini berfokus pada tiga strategi: menggunakan yang ada APIs, menerapkan caching yang efektif, dan mengoptimalkan logika bisnis.

Strategi optimasi: Menggunakan yang ada APIs

AWS Blu Age menyediakan beberapa antarmuka DAO yang kuat APIs yang dapat Anda gunakan untuk mengoptimalkan kinerja. Dua antarmuka utama — DAOManager dan SQLExecution Builder — menawarkan kemampuan untuk meningkatkan kinerja aplikasi.

DAOManager

DAOManager berfungsi sebagai antarmuka utama untuk operasi database dalam aplikasi modern. Ini menawarkan beberapa metode untuk meningkatkan operasi database dan meningkatkan kinerja aplikasi, terutama untuk membuat, membaca, memperbarui, dan menghapus (CRUD) operasi langsung dan pemrosesan batch.

- Gunakan SetMaxResults. Di DAOManager API, Anda dapat menggunakan SetMaxResultsmetode ini untuk menentukan jumlah maksimum catatan yang akan diambil dalam satu operasi database. Secara default, hanya DAOManager mengambil 10 catatan sekaligus, yang dapat menyebabkan beberapa panggilan database saat memproses kumpulan data besar. Gunakan pengoptimalan ini ketika aplikasi Anda perlu memproses sejumlah besar catatan dan saat ini membuat beberapa panggilan database untuk mengambilnya. Ini sangat berguna dalam skenario pemrosesan batch di mana Anda melakukan iterasi melalui kumpulan data besar. Dalam contoh berikut, kode di sebelah kiri (sebelum optimasi) menggunakan nilai pengambilan data default dari 10 catatan. Kode di sebelah kanan (setelah optimasi) ditetapkan setMaxResultsuntuk mengambil 100.000 catatan sekaligus.

  > **Note**
  >
  > Pilih ukuran batch yang lebih besar dengan hati-hati dan periksa ukuran objek, karena pengoptimalan ini meningkatkan jejak memori.

- Ganti SetOnGreatorOrEqual dengan SetOnEqual. Optimasi ini melibatkan perubahan metode yang Anda gunakan untuk mengatur kondisi untuk mengambil catatan. SetOnGreatorOrEqualMetode ini mengambil catatan yang lebih besar dari atau sama dengan nilai yang ditentukan, sedangkan hanya SetOnEqualmengambil catatan yang sama persis dengan nilai yang ditentukan.

  Gunakan SetOnEqualseperti yang diilustrasikan dalam contoh kode berikut, ketika Anda tahu bahwa Anda memerlukan pencocokan yang tepat dan saat ini Anda menggunakan SetOnGreatorOrEqualmetode diikuti oleh readNextEqual(). Optimalisasi ini mengurangi pengambilan data yang tidak perlu.

- Gunakan operasi batch write dan update. Anda dapat menggunakan operasi batch untuk mengelompokkan beberapa operasi tulis atau pembaruan ke dalam satu transaksi database. Ini mengurangi jumlah panggilan database dan secara signifikan dapat meningkatkan kinerja untuk operasi yang melibatkan beberapa catatan.

  Dalam contoh berikut, kode di sebelah kiri melakukan operasi tulis dalam satu lingkaran, yang memperlambat kinerja aplikasi. Anda dapat mengoptimalkan kode ini dengan menggunakan operasi penulisan batch: Selama setiap iterasi `WHILE` loop, Anda menambahkan catatan ke batch hingga ukuran batch mencapai ukuran 100 yang telah ditentukan sebelumnya. Anda kemudian dapat menyiram batch ketika mencapai ukuran batch yang telah ditentukan, dan kemudian flush semua catatan yang tersisa ke database. Ini sangat berguna dalam skenario di mana Anda memproses kumpulan data besar yang memerlukan pembaruan.

- Tambahkan indeks. Menambahkan indeks adalah optimasi tingkat database yang dapat secara signifikan meningkatkan kinerja kueri. Indeks memungkinkan database untuk dengan cepat menemukan baris dengan nilai kolom tertentu tanpa memindai seluruh tabel. Gunakan pengindeksan pada kolom yang sering digunakan dalam `WHERE` klausa, `JOIN` kondisi, atau pernyataan. `ORDER BY` Ini sangat penting untuk tabel besar atau ketika pengambilan data cepat sangat penting.

SQLExecutionPembangun

SQLExecutionBuilder adalah API fleksibel yang dapat Anda gunakan untuk mengendalikan kueri SQL yang akan dieksekusi, mengambil kolom tertentu saja, `INSERT` dengan menggunakan`SELECT`, dan menggunakan nama tabel dinamis. Dalam contoh berikut, SQLExecutor Builder menggunakan kueri kustom yang Anda tentukan.

Memilih antara DAOManager dan SQLExecution Builder

Pilihan antara ini APIs tergantung pada kasus penggunaan spesifik Anda:

- Gunakan DAOManager saat Anda ingin AWS Blu Age Runtime untuk menghasilkan kueri SQL alih-alih menulisnya sendiri.
- Pilih SQLExecution Builder saat Anda perlu menulis kueri SQL untuk memanfaatkan fitur khusus database atau menulis kueri SQL yang optimal.

Strategi pengoptimalan: Caching

Dalam aplikasi modern, menerapkan strategi caching yang efektif dapat secara signifikan mengurangi panggilan database dan meningkatkan waktu respons. Ini membantu menjembatani kesenjangan kinerja antara mainframe dan lingkungan cloud.

Dalam aplikasi AWS Blu Age, implementasi caching sederhana menggunakan struktur data internal seperti peta hash atau daftar array, sehingga Anda tidak perlu menyiapkan solusi caching eksternal yang memerlukan biaya dan restrukturisasi kode. Pendekatan ini sangat efektif untuk data yang sering diakses tetapi jarang berubah. Saat Anda menerapkan caching, pertimbangkan batasan memori dan pola pembaruan untuk memastikan bahwa data yang di-cache tetap konsisten dan memberikan manfaat kinerja aktual.

Kunci keberhasilan caching adalah mengidentifikasi data yang tepat untuk cache. Dalam contoh berikut, kode di sebelah kiri selalu membaca data dari tabel, sedangkan kode di sebelah kanan membaca data dari tabel ketika peta hash lokal tidak memiliki nilai untuk kunci yang diberikan. `cacheMap`adalah objek peta hash yang dibuat dalam konteks program dan dihapus dalam metode pembersihan konteks program.

Caching dengan DAOManager:

Caching dengan SQLExecution Builder:

Strategi optimasi: Optimalisasi logika bisnis

Optimalisasi logika bisnis berfokus pada restrukturisasi kode yang dihasilkan secara otomatis dari AWS Blu Age agar lebih selaras dengan kemampuan arsitektur modern. Ini menjadi perlu ketika kode yang dihasilkan mempertahankan struktur logika yang sama dengan kode mainframe lama, yang mungkin tidak optimal untuk sistem modern. Tujuannya adalah untuk meningkatkan kinerja sambil mempertahankan kesetaraan fungsional dengan aplikasi asli.

Pendekatan pengoptimalan ini melampaui tweak API sederhana dan strategi caching. Ini melibatkan perubahan bagaimana aplikasi memproses data dan berinteraksi dengan database. Pengoptimalan umum termasuk menghindari operasi baca yang tidak perlu untuk pembaruan sederhana, menghapus panggilan database yang berlebihan, dan merestrukturisasi pola akses data agar lebih selaras dengan arsitektur aplikasi modern. Berikut adalah beberapa contoh:

- Memperbarui data secara langsung di database.Merestrukturisasi logika bisnis Anda dengan menggunakan pembaruan SQL langsung alih-alih beberapa DAOManager operasi dengan loop.

Misalnya, kode berikut (sisi kiri) membuat beberapa panggilan database dan menggunakan memori yang berlebihan. Secara khusus, ia menggunakan beberapa operasi baca dan tulis database dalam loop, pembaruan individual alih-alih pemrosesan batch, dan pembuatan objek yang tidak perlu untuk setiap iterasi.

Kode dioptimalkan berikut (sisi kanan) menggunakan operasi pembaruan SQL Langsung tunggal. Secara khusus, ia menggunakan panggilan database tunggal alih-alih beberapa panggilan dan tidak memerlukan loop karena semua pembaruan ditangani dalam satu pernyataan. Optimalisasi ini memberikan kinerja dan pemanfaatan sumber daya yang lebih baik, dan mengurangi kompleksitas. Ini mencegah injeksi SQL, menyediakan caching rencana kueri yang lebih baik, dan membantu meningkatkan keamanan.

> **Note**
>
> Selalu gunakan kueri berparameter untuk mencegah injeksi SQL dan memastikan manajemen transaksi yang tepat.

- Mengurangi panggilan database yang berlebihan. Panggilan database redundan dapat secara signifikan mempengaruhi kinerja aplikasi, terutama ketika mereka terjadi dalam loop. Teknik optimasi yang sederhana namun efektif adalah menghindari pengulangan kueri database yang sama beberapa kali. Perbandingan kode berikut menunjukkan bagaimana memindahkan panggilan `retrieve()` database di luar loop mencegah eksekusi berlebihan dari kueri identik, yang meningkatkan efisiensi.

- Mengurangi panggilan database dengan menggunakan **`JOIN`** klausa SQL. Implementasikan SQLExecution Builder untuk meminimalkan panggilan ke database. SQLExecutionBuilder memberikan kontrol lebih besar atas pembuatan SQL dan sangat berguna untuk kueri kompleks yang DAOManager tidak dapat ditangani secara efisien. Misalnya, kode berikut menggunakan beberapa DAOManager panggilan:

```
List<Employee> employees = daoManager.readAll();
for(Employee emp : employees) {
    Department dept = deptManager.readById(emp.getDeptId());  // Additional call for
 each employee
    Project proj = projManager.readById(emp.getProjId());     // Another call for
 each employee
    processEmployeeData(emp, dept, proj);
```

```
}
```

Kode yang dioptimalkan menggunakan satu panggilan database di SQLExecution Builder:

```
SQLExecutionBuilder builder = new SQLExecutionBuilder();
builder.append("SELECT e.*, d.name as dept_name, p.name as proj_name");
builder.append("FROM employee e");
builder.append("JOIN department d ON e.dept_id = d.id");
builder.append("JOIN project p ON e.proj_id = p.id");
builder.append("WHERE e.status = ?", "ACTIVE");

List<Map<String, Object>> results = builder.execute();  // Single database call
for(Map<String, Object> result : results) {
    processComplexData(result);
}
```

Menggunakan strategi optimasi bersama

Ketiga strategi ini bekerja secara sinergis: APIs menyediakan alat untuk akses data yang efisien, caching mengurangi kebutuhan untuk pengambilan data berulang, dan optimasi logika bisnis memastikan bahwa ini APIs digunakan dengan cara yang paling efektif. Pemantauan dan penyesuaian reguler dari pengoptimalan ini memastikan peningkatan kinerja berkelanjutan sambil mempertahankan keandalan dan fungsionalitas aplikasi yang dimodernisasi. Kunci sukses terletak pada pemahaman kapan dan bagaimana menerapkan setiap strategi berdasarkan karakteristik aplikasi dan tujuan kinerja Anda.

Alat

- JProfileradalah alat profil Java yang dirancang untuk pengembang dan insinyur kinerja. Ini menganalisis aplikasi Java dan membantu mengidentifikasi kemacetan kinerja, kebocoran memori, dan masalah threading. JProfiler menawarkan CPU, memori, dan thread profiling serta database dan Java virtual machine (JVM) monitoring untuk memberikan wawasan tentang perilaku aplikasi.

> **Note**
>
> Sebagai alternatif JProfiler, Anda dapat menggunakan Java VisualVM. Ini adalah alat profil dan pemantauan kinerja open source gratis untuk aplikasi Java yang menawarkan pemantauan real-time penggunaan CPU, konsumsi memori, manajemen thread, dan

statistik pengumpulan sampah. Karena Java VisualVM adalah alat JDK bawaan, ini lebih hemat biaya daripada untuk kebutuhan profil dasar. JProfiler

- pgAdmin adalah alat administrasi dan pengembangan open source untuk PostgreSQL. Ini menyediakan antarmuka grafis yang membantu Anda membuat, memelihara, dan menggunakan objek database. Anda dapat menggunakan pgAdmin untuk melakukan berbagai tugas, mulai dari menulis kueri SQL sederhana hingga mengembangkan basis data yang kompleks. Fitur-fiturnya termasuk sintaks yang menyoroti editor SQL, editor kode sisi server, agen penjadwalan untuk SQL, shell, dan tugas batch, dan dukungan untuk semua fitur PostgreSQL untuk pengguna PostgreSQL pemula dan berpengalaman.

Praktik terbaik

Mengidentifikasi hotspot kinerja:

- Dokumentasikan metrik kinerja dasar sebelum Anda memulai pengoptimalan.
- Tetapkan target peningkatan kinerja yang jelas berdasarkan persyaratan bisnis.
- Saat melakukan benchmarking, nonaktifkan logging verbose, karena dapat memengaruhi kinerja.
- Siapkan rangkaian uji kinerja dan jalankan secara berkala.
- Gunakan pgAdmin versi terbaru. (Versi lama tidak mendukung paket `EXPLAIN` kueri.)
- Untuk benchmarking, lepaskan JProfiler setelah pengoptimalan Anda selesai karena menambah latensi.
- Untuk benchmarking, pastikan untuk menjalankan server dalam mode mulai alih-alih mode debug, karena mode debug menambah latensi.

Strategi optimasi:

- Konfigurasikan SetMaxResultsnilai dalam `application.yaml` file, untuk menentukan batch berukuran tepat sesuai dengan spesifikasi sistem Anda.
- Konfigurasikan SetMaxResultsnilai berdasarkan volume data dan kendala memori.
- Ubah SetOnGreatorOrEqualmenjadi SetOnEqualhanya ketika panggilan berikutnya`.readNextEqual()`.
- Dalam operasi penulisan atau pembaruan batch, tangani batch terakhir secara terpisah, karena mungkin lebih kecil dari ukuran batch yang dikonfigurasi dan dapat dilewatkan oleh operasi tulis atau pembaruan.

Caching:

- Bidang yang diperkenalkan untuk caching di`processImpl`, yang bermutasi dengan setiap eksekusi, harus selalu didefinisikan dalam konteks itu. `processImpl` Bidang juga harus dibersihkan dengan menggunakan `cleanUp()` metode `doReset()` or.
- Saat Anda menerapkan cache dalam memori, ukuran cache yang tepat. Cache yang sangat besar yang disimpan dalam memori dapat mengambil semua sumber daya, yang dapat memengaruhi kinerja keseluruhan aplikasi Anda.

SQLExecutionPembangun:

- Untuk kueri yang Anda rencanakan untuk digunakan di SQLExecution Builder, gunakan nama kunci seperti`PROGRAMNAME_STATEMENTNUMBER`.
- Saat Anda menggunakan SQLExecution Builder, selalu periksa `Sqlcod` bidangnya. Bidang ini berisi nilai yang menentukan apakah kueri dijalankan dengan benar atau mengalami kesalahan apa pun.
- Gunakan kueri berparameter untuk mencegah injeksi SQL.

Optimalisasi logika bisnis:

- Pertahankan kesetaraan fungsional saat merestrukturisasi kode, dan jalankan pengujian regresi dan perbandingan basis data untuk subset program yang relevan.
- Pertahankan snapshot profil untuk perbandingan.

Epik

Instal JProfiler dan pgAdmin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan konfigurasikan JProfiler.	1. Pada halaman JProfiler unduhan, unduh JProfiler untuk sistem operasi Anda (Windows, macOS, atau Linux).	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Selesaikan proses instalasi dengan mengikuti instruksi dalam JProfiler dokumentasi. 3. Lakukan konfigurasi awal: a. Luncurkan JProfiler aplikasi. b. Aktifkan kunci lisensi. c. Konfigurasikan pengaturan JVM default. 4. Verifikasi instalasi: a. Peluncuran JProfiler. b. Jalankan sesi pembuatan profil pengujian. Untuk petunjuk, lihat Melampirkan ke layanan lokal di bagian Membuat Profil JVM pada dokumentasi. JProfiler	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan konfigurasikan pgAdmin.	Pada langkah ini, Anda menginstal dan mengkonfigurasi klien DB untuk menanyakan database Anda. Pola ini menggunakan database PostgreSQL dan pgAdmin sebagai klien database. Jika Anda menggunakan mesin database lain, ikuti dokumentasi untuk klien DB yang sesuai. 1. Unduh dan Instal pgAdmin untuk sistem operasi Anda dari halaman unduhan pgAdmin. Atur kata sandi utama selama instalasi. 2. Connect ke server database. Untuk petunjuk, lihat dokumentasi pgAdmin. 3. Verifikasi instalasi dengan menggunakan Pgadmin Query Tool untuk menjalankan query SQL dasar.	Pengembang aplikasi

Identifikasi hotspot

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan pencatatan kueri SQL di aplikasi AWS Blu Age Anda.	Aktifkan logger untuk pencatatan kueri SQL di `application.properties` file aplikasi AWS	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Blu Age Anda, seperti yang dijelaskan di bagian Arsitektur.	
Menghasilkan dan menganalisis `EXPLAIN` rencana kueri untuk mengidentifikasi hotspot kinerja database.	Untuk detailnya, lihat bagian Arsitektur.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat JProfiler snapshot untuk menganalisis kasus uji berkinerja lambat.	1. Peluncuran JProfiler: a. Buka JProfiler aplikasi. b. Di Start Center, pilih tab Quick Attach. c. Pilih proses Tomcat dari daftar. d. Pilih Mulai. 2. Konfigurasikan pengaturan profil: a. Pilih Opsi Pengambilan Sampel yang Direkomen dasikan. b. Simpan opsi default di layar konfigurasi, lalu pilih OK. c. Tunggu status Terhubung ditampilkan di bilah status. 3. Rekam data kinerja: a. Pada bilah alat, pilih Mulai Merekam. b. Jalankan kasus uji Anda dan tunggu penyelesa iannya. c. Pilih Stop Recording. 4. Simpan dan lihat snapshot: a. Pilih Simpan Snapshot. b. Pilih lokasi tempat Anda ingin menyimpan snapshot, lalu pilih Simpan.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Analisis JProfiler snapshot untuk mengidentifikasi kemacetan kinerja.	Ikuti langkah-langkah ini untuk menganalisis JProfiler snapshot. 1. Buka snapshot: a. Pilih File, Buka Snapshot. b. Arahkan ke lokasi snapshot yang disimpan. c. Pilih Buka jendela baru untuk memulai analisis. 2. Menganalisis operasi basis data: a. Arahkan ke bagian JDBC/Database, dan pilih tab JPA/Hibernate. b. Tinjau kueri berdasarkan waktu (yang merupakan opsi pengurutan default). c. Pilih kueri paling atas untuk pengoptimalan. 3. Tinjau penggunaan CPU: a. Di bagian CPU, pilih tampilan Hot Spots. b. Identifikasi kode paling atas dalam persentase waktu untuk pengoptim alan. c. Perhatikan pola yang berulang. 4. Setelah Anda mengident ifikasi hotspot, buat garis	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dasar dan kemudian terapkan strategi pengoptimalan yang sesuai, seperti yang dibahas dalam epos berikut. Fokus pada pengoptimalan ini: • Untuk kueri lambat, gunakan SQLExecution Builder atau DAOManager optimasi. • Untuk akses data yang sering, terapkan mekanisme caching. • Untuk logika bisnis yang kompleks, sederhanakan logika, hindari membuat terlalu banyak panggilan database, dan tentukan apakah kode dapat disederhanakan dengan menjalankan kueri sederhana. Misalnya, jika kode menyisipkan data dari Tabel A ke Tabel B, tulis ulang kode ini untuk menjalankan perintah SQL yang mirip dengan: `INSERT INTO TABLE A SELECT * FROM TABLE B` 5. Lanjutkan untuk menganalisis dan mengoptimalkan kode dalam urutan dampak	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(tertinggi ke terendah) hingga aplikasi memenuhi persyaratan kinerja Anda. Untuk informasi selengkapnya tentang penggunaan JProfiler , lihat bagian Arsitektur dan JProfiler dokumentasi.	

Tetapkan garis dasar

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan garis dasar kinerja sebelum Anda menerapkan pengoptimalan.	1. Sebagai pengukuran kinerja awal, jalankan kasus pengujian Anda tanpa pengoptimalan apa pun. Buat JProfiler snapshot dari status saat ini, dan dokumentasikan metrik berikut: • Total waktu eksekusi • Jumlah panggilan basis data • Persentase pemanfaatan CPU • Pola penggunaan memori 2. Buat baseline dokumen: a. Simpan semua metrik dengan cap waktu dan kondisi pengujian. b. Perhatikan faktor eksternal apa pun yang	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mungkin memengaruhi kinerja. c. Simpan JProfiler snapshot untuk perbandingan setelah pengoptimalan. 3. Rencanakan perubahan optimasi yang dijelaskan dalam epik berikutnya: a. Tinjau metrik dasar. b. Identifikasi operasi yang ingin Anda optimalkan. c. Buat cadangan kode saat ini. d. Dokumentasikan perbaikan yang diharapkan. 4. Ukur setiap perubahan sebelum implementasi: a. Jalankan kasus uji dengan menggunakan kondisi dasar. b. Catat waktu eksekusi. c. Dokumentasikan perilaku saat ini.	

Terapkan strategi pengoptimalan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Optimalkan panggilan baca.	Optimalkan pengambilan data dengan menggunakan	Pengembang aplikasi, DAOManager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	DAOManager SetMaxRes ultsmetode. Untuk informasi lebih lanjut tentang pendekata n ini, lihat bagian Arsitektur.	
Refactor logika bisnis untuk menghindari beberapa panggilan ke database.	Kurangi panggilan database dengan menggunakan `JOIN` klausa SQL. Untuk detail dan contoh, lihat Optimasi logika bisnis di bagian Arsitektur.	Pengembang aplikasi, SQLExecution Pembangun
Refactor kode untuk menggunakan caching untuk mengurangi latensi panggilan baca.	Untuk informasi tentang teknik ini, lihat Caching di bagian Arsitektur.	Pengembang aplikasi
Tulis ulang kode yang tidak efisien yang menggunak an beberapa DAOManage r operasi untuk operasi pembaruan sederhana.	Untuk informasi selengkap nya tentang memperbarui data secara langsung di database, lihat Optimasi logika bisnis di bagian Arsitektur.	Pengembang aplikasi

Strategi pengoptimalan uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi setiap perubahan optimasi secara iteratif sambil mempertahankan kesetaraan fungsional.	1. Setelah Anda menerapkan pengoptimalan: a. Jalankan kasus uji yang sama dari baseline yang Anda buat. b. Catat waktu eksekusi baru.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	c. Verifikasi bahwa hasil cocok dengan output dasar. 2. jika optimasi menghasilkan peningkatan, dokumenta sikan keuntungan aktual, dan kemudian lanjutkan ke optimasi berikutnya. Jika Anda menemukan masalah, putar kembali perubahan, tinjau pendekatan Anda, dan coba solusi alternatif. 3. Lacak kemajuan untuk setiap perubahan: a. Bandingkan metrik baru dengan metrik dasar. b. Hitung persentase peningkatan. c. Pertahankan total keuntungan yang berjalan. **Note** Menggunakan metrik dasar sebagai referensi memastika n pengukuran yang akurat dari setiap dampak pengoptimalan sambil	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mempertahankan keandalan sistem.	

Pemecahan masalah

Isu	Solusi
Ketika Anda menjalankan aplikasi modern, Anda melihat pengecualian dengan kesalahan `Query_ID not found`.	Untuk menyelesaikan masalah ini: 1. Periksa apakah file bernama `query.sql` ada di `sql` folder di direktori kerja server aplikasi. 2. Verifikasi bahwa nama `.sql` file cocok dengan nama program. Misalnya, untuk program bernama`ABC`, nama file harus`ABC.sql`.
Anda telah menambahkan indeks, tetapi Anda tidak melihat peningkatan kinerja apa pun.	Ikuti langkah-langkah ini untuk memastikan bahwa mesin kueri menggunakan indeks: 1. Jika Anda menggunakan PostgreSQL, jalankan perintah. `EXPLAIN <sql query>` 2. Periksa output dari perintah untuk `Seq Scan` atau`Index Scan`. `Seq Scan`berarti bahwa kumpulan indeks tidak tersedia pada kolom yang digunakan `where` klausa kueri SQL. 3. Buat indeks pada kumpulan kolom yang benar dan cobalah untuk membentuk kueri yang menggunakan indeks yang dibuat.
Anda menemukan out-of-memory pengecual ian.	Verifikasi bahwa kode melepaskan memori yang dipegang oleh struktur data.

Isu	Solusi
Operasi penulisan batch menghasilkan catatan yang hilang dalam tabel	Tinjau kode untuk memastikan bahwa operasi penulisan tambahan dilakukan ketika jumlah batch tidak nol.
Pencatatan SQL tidak muncul di log aplikasi.	1. Verifikasi lokasi dan sintaks file konfigurasi logging. 2. Pastikan bahwa tingkat log diatur `DEBUG` untuk paket tertentu. Contoh: `level.com.netfective.bluage .gapwalk.rt.blu4iv.dao: DEBUG level.org.springframework.jdbc .core: DEBUG` 3. Periksa izin file log server aplikasi. 4. Mulai ulang server aplikasi setelah perubahan konfigurasi.

Sumber daya terkait

- Aplikasi refactoring secara otomatis dengan AWS Blu Age (Panduan Pengguna)AWS Mainframe Modernization
- Dokumentasi pgAdmin
- JProfiler dokumentasi

Mengamankan dan merampingkan akses pengguna dalam database federasi Db2 di AWS dengan menggunakan konteks tepercaya

Sai Parthasaradhi, Amazon Web Services

Ringkasan

Banyak perusahaan yang memigrasikan beban kerja mainframe lama mereka ke Amazon Web Services (AWS). Migrasi ini mencakup pemindahan IBM Db2 untuk z/OS database ke Db2 untuk Linux, Unix, dan Windows (LUW) di Amazon Elastic Compute Cloud (Amazon). EC2 Selama migrasi bertahap dari lokasi ke AWS, pengguna mungkin perlu mengakses data di IBM Db2 z/OS dan di Db2 LUW di EC2 Amazon hingga semua aplikasi dan database sepenuhnya dimigrasikan ke Db2 LUW. Dalam skenario akses data jarak jauh seperti itu, otentikasi pengguna dapat menjadi tantangan karena platform yang berbeda menggunakan mekanisme otentikasi yang berbeda.

Pola ini mencakup cara mengatur server federasi pada Db2 untuk LUW dengan Db2 untuk z/OS sebagai database jarak jauh. Pola ini menggunakan konteks tepercaya untuk menyebarkan identitas pengguna dari Db2 LUW ke Db2 z/OS tanpa mengautentikasi ulang pada database jarak jauh. Untuk informasi selengkapnya tentang konteks tepercaya, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Instans Db2 yang berjalan pada instans Amazon EC2
- Db2 jarak jauh untuk z/OS database yang berjalan di tempat
- Jaringan lokal yang terhubung ke AWS melalui AWS Site-to-SiteVPN atau AWS Direct Connect

Arsitektur

Arsitektur target

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- AWS Site-to-Site VPN membantu Anda meneruskan lalu lintas antar instans yang Anda luncurkan di AWS dan jaringan jarak jauh Anda sendiri.

Alat-alat lainnya

- db2cli adalah perintah antarmuka baris perintah interaktif Db2 (CLI).

Epik

Aktifkan federasi pada database Db2 LUW yang berjalan di AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan federasi pada DB2 LUW DB.	Untuk mengaktifkan federasi pada DB2 LUW, jalankan perintah berikut. `update dbm cfg using federated YES`	DBA
Mulai ulang database.	Untuk me-restart database, jalankan perintah berikut. `db2stop force; db2start;`	DBA

Katalog database jarak jauh

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Katalog subsistem Db2 z/OS jarak jauh.	Untuk membuat katalog z/OS database Db2 jarak jauh di Db2 LUW yang berjalan	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di AWS, gunakan perintah contoh berikut. `catalog TCPIP NODE tcpnode REMOTE mainframehost SERVER mainframeport`	
Katalog database jarak jauh.	Untuk membuat katalog database jarak jauh, gunakan perintah contoh berikut. `catalog db dbnam1 as ndbnam1 at node tcpnode`	DBA

Buat definisi server jarak jauh

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan kredensi pengguna untuk database z/OS Db2 jarak jauh.	Sebelum melanjutkan dengan langkah selanjutnya, kumpulkan informasi berikut: • Nama z/OS subsistem Db2 — Nama Db2 yang dikatalogkan di LUW dari z/OS langkah sebelumnya (misalnya,) `ndbnam1` • Versi Db2 - z/OS Versi z/OS subsistem Db2 (misalnya,) `12` • ID z/OS pengguna Db2 — Pengguna dengan hak istimewa BIND, yang diperlukan untuk membuat	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	hanya definisi server (misalnya,) `dbuser1` • z/OS Kata sandi Db2 — Kata sandi untuk `dbuser1` (misalnya,) `dbpasswd` • Pengguna z/OS proxy Db2 — ID pengguna proxy, yang akan digunakan untuk membuat koneksi tepercaya (misalnya,) `zproxy` • Kata sandi z/OS proxy Db2 — Kata sandi untuk `zproxy` pengguna (misalnya,) `zproxy`	
Buat pembungkus DRDA.	Untuk membuat pembungkus DRDA, jalankan perintah berikut. `CREATE WRAPPER DRDA;`	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat definisi server.	Untuk membuat definisi server, jalankan perintah contoh berikut. `CREATE SERVER ndbserver TYPE DB2/ZOS VERSION 12 WRAPPER DRDA AUTHORIZATION "dbuser1" PASSWORD "dbpasswd " OPTIONS ( DBNAME 'ndbnam1',FED_PROXY_USER 'ZPROXY' );` Dalam definisi ini, `FED_PROXY_USER` menentukan pengguna proxy yang akan digunakan untuk membangun koneksi tepercaya ke database Db2 z/OS . ID pengguna otorisasi dan kata sandi diperlukan hanya untuk membuat objek server jarak jauh di database Db2 LUW. Mereka tidak akan digunakan nanti selama runtime.	DBA

Buat pemetaan pengguna

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pemetaan pengguna untuk pengguna proxy.	Untuk membuat pemetaan pengguna untuk pengguna proxy, jalankan perintah berikut.	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`CREATE USER MAPPING FOR ZPROXY SERVER ndbserver OPTIONS (REMOTE_AUTHID 'ZPROXY', REMOTE_PASSWORD 'zproxy');`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pemetaan pengguna untuk setiap pengguna di Db2 LUW.	Buat pemetaan pengguna untuk semua pengguna di database Db2 LUW di AWS yang perlu mengakses data jarak jauh melalui pengguna proxy. Untuk membuat pemetaan pengguna, jalankan perintah berikut. `CREATE USER MAPPING FOR PERSON1 SERVER ndbserver OPTIONS (REMOTE_AUTHID 'USERZID', USE_TRUSTED_CONTEXT 'Y');` Pernyataan tersebut menetapkan bahwa pengguna di Db2 LUW (`PERSON1`) dapat membuat koneksi tepercaya ke database z/OS Db2 jarak jauh (). `USE_TRUSTED_CONTEXT 'Y'` Setelah koneksi dibuat melalui pengguna proxy, pengguna dapat mengakses data dengan menggunakan ID z/OS pengguna Db2 ()`REMOTE_AUTHID 'USERZID'` .	DBA

Buat objek konteks tepercaya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat objek konteks tepercaya.	Untuk membuat objek konteks tepercaya pada z/OS database Db2 jarak jauh, gunakan perintah contoh berikut. `CREATE TRUSTED CONTEXT CTX_LUW_ZOS BASED UPON CONNECTIO N USING SYSTEM AUTHID ZPROXY ATTRIBUTES ( ADDRESS '10.10.10.10' ) NO DEFAULT ROLE ENABLE WITH USE FOR PUBLIC WITHOUT AUTHENTIC ATION;` Dalam definisi ini, `CTX_LUW_ZOS` adalah nama arbitrer untuk objek konteks tepercaya. Objek berisi ID pengguna proxy dan alamat IP server dari mana koneksi tepercaya harus berasal. Dalam contoh ini, server database Db2 LUW di AWS. Anda dapat menggunakan nama domain alih-alih alamat IP. Klausul `WITH USE FOR PUBLIC WITHOUT AUTHENTIC ATION` ini menunjukkan bahwa mengalihkan ID	DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengguna pada koneksi tepercaya diperbolehkan untuk setiap ID pengguna. Password tidak perlu disediakan.	

Sumber daya terkait

- Fasilitas Kontrol Akses Sumber Daya IBM (RACF)
- Federasi IBM Db2 LUW
- Konteks tepercaya

Informasi tambahan

Konteks tepercaya Db2

Konteks tepercaya adalah objek database Db2 yang mendefinisikan hubungan kepercayaan antara server federasi dan server database jarak jauh. Untuk menentukan hubungan tepercaya, konteks tepercaya menentukan atribut kepercayaan. Ada tiga jenis atribut kepercayaan:

- ID otorisasi sistem yang membuat permintaan koneksi database awal
- Alamat IP atau nama domain dari mana koneksi dibuat
- Pengaturan enkripsi untuk komunikasi data antara server database dan klien database

Koneksi tepercaya dibuat ketika semua atribut permintaan koneksi cocok dengan atribut yang ditentukan dalam objek konteks tepercaya apa pun yang ditentukan di server. Ada dua jenis koneksi tepercaya: implisit dan eksplisit. Setelah koneksi terpercaya implisit dibuat, pengguna mewarisi peran yang tidak tersedia bagi mereka di luar cakupan definisi koneksi tepercaya tersebut. Setelah koneksi tepercaya eksplisit dibuat, pengguna dapat diaktifkan pada koneksi fisik yang sama, dengan atau tanpa otentikasi. Selain itu, pengguna Db2 dapat diberikan peran yang menentukan hak istimewa yang hanya untuk digunakan dalam koneksi tepercaya. Pola ini menggunakan koneksi tepercaya eksplisit.

Konteks tepercaya dalam pola ini

Setelah pola selesai, PERSON1 pada Db2 LUW mengakses data jarak jauh dari Db2 z/OS dengan menggunakan konteks tepercaya federasi. Koneksi untuk PERSON1 dibuat melalui pengguna proxy jika koneksi berasal dari alamat IP atau nama domain yang ditentukan dalam definisi konteks tepercaya. Setelah koneksi dibuat, PERSON1 ID z/OS pengguna Db2 yang sesuai diaktifkan tanpa autentikasi ulang, dan pengguna dapat mengakses data atau objek berdasarkan hak istimewa Db2 yang disiapkan untuk pengguna tersebut.

Manfaat konteks tepercaya federasi

- Pendekatan ini mempertahankan prinsip hak istimewa terkecil dengan menghilangkan penggunaan ID pengguna umum atau ID aplikasi yang membutuhkan superset dari semua hak istimewa yang diperlukan oleh semua pengguna.
- Identitas sebenarnya dari pengguna yang melakukan transaksi pada database federasi dan jarak jauh selalu diketahui dan dapat diaudit.
- Kinerja meningkat karena koneksi fisik digunakan kembali di seluruh pengguna tanpa perlu server federasi untuk mengautentikasi ulang.

Transfer z/OS data Db2 skala besar ke Amazon S3 dalam file CSV

Bruno Sahinoglu, Abhijit Kshirsagar, dan Ivan Schuster, Amazon Web Services

Ringkasan

Mainframe masih merupakan sistem catatan di banyak perusahaan, yang berisi sejumlah besar data termasuk entitas data induk dengan catatan transaksi bisnis saat ini serta historis. Hal ini sering tersilo dan tidak mudah diakses oleh sistem terdistribusi dalam perusahaan yang sama. Dengan munculnya teknologi cloud dan demokratisasi big data, perusahaan tertarik untuk menggunakan wawasan yang tersembunyi dalam data mainframe untuk mengembangkan kemampuan bisnis baru.

Dengan tujuan itu, perusahaan ingin membuka data Db2 mainframe mereka ke lingkungan Amazon Web Services (AWS) Cloud mereka. Alasan bisnisnya banyak dan metode transfer berbeda dari kasus ke kasus. Anda mungkin lebih suka menghubungkan aplikasi Anda langsung ke mainframe, atau Anda mungkin lebih suka mereplikasi data Anda dalam waktu dekat. Jika kasus penggunaannya adalah untuk memberi makan gudang data atau data lake, memiliki up-to-date salinan tidak lagi menjadi perhatian, dan prosedur yang dijelaskan dalam pola ini mungkin cukup, terutama jika Anda ingin menghindari biaya lisensi produk pihak ketiga. Kasus penggunaan lain mungkin transfer data mainframe untuk proyek migrasi. Dalam skenario migrasi, data diperlukan untuk melakukan pengujian kesetaraan fungsional. Pendekatan yang dijelaskan dalam posting ini adalah cara yang hemat biaya untuk mentransfer data Db2 ke lingkungan AWS Cloud.

Karena Amazon Simple Storage Service (Amazon S3) adalah salah satu layanan AWS yang paling terintegrasi, Anda dapat mengakses data dari sana dan mengumpulkan wawasan secara langsung dengan menggunakan layanan AWS lainnya seperti Amazon Athena, fungsi AWS Lambda, atau Amazon. QuickSight Anda juga dapat memuat data ke Amazon Aurora atau Amazon DynamoDB dengan menggunakan AWS Glue atau AWS Database Migration Service (AWS DMS). Dengan tujuan itu, ini menjelaskan cara membongkar data Db2 dalam file CSV dalam format ASCII pada mainframe dan mentransfer file ke Amazon S3.

Untuk tujuan ini, skrip mainframe telah dikembangkan untuk membantu menghasilkan bahasa kontrol pekerjaan (JCLs) untuk membongkar dan mentransfer tabel Db2 sebanyak yang Anda butuhkan.

Prasyarat dan batasan

Prasyarat

- Pengguna sistem z/OS operasi IBM dengan otorisasi untuk menjalankan restrukturisasi Extended Execuutor (REXX) dan skrip JCL.

- Akses ke z/OS Unix System Services (USS) untuk menghasilkan kunci pribadi dan publik SSH (Secure Shell).
- Bucket S3 yang bisa ditulis. Untuk informasi selengkapnya, lihat Membuat bucket S3 pertama Anda di dokumentasi Amazon S3.
- Server berkemampuan AWS Transfer Family SSH File Transfer Protocol (SFTP) menggunakan Layanan yang dikelola sebagai penyedia identitas dan Amazon S3 sebagai layanan penyimpanan AWS. Untuk informasi selengkapnya, lihat Membuat server berkemampuan SFTP dalam dokumentasi AWS Transfer Family.

Batasan

- Pendekatan ini tidak cocok untuk sinkronisasi data hampir real-time atau real-time.
- Data hanya dapat dipindahkan dari Db2 z/OS ke Amazon S3, bukan sebaliknya.

Arsitektur

Tumpukan teknologi sumber

- Mainframe menjalankan Db2 pada z/OS

Tumpukan teknologi target

- AWS Transfer Family
- Amazon S3
- Amazon Athena
- Amazon QuickSight
- AWS Glue
- Amazon Relational Database Service (Amazon RDS)
- Amazon Aurora
- Amazon Redshift

Arsitektur sumber dan target

Diagram berikut menunjukkan proses untuk menghasilkan, mengekstraksi, dan mentransfer z/OS data Db2 dalam format ASCII CSV ke bucket S3.

1. Daftar tabel dipilih untuk migrasi data dari katalog Db2.
2. Daftar ini digunakan untuk mendorong pembuatan pekerjaan bongkar dengan kolom numerik dan data dalam format eksternal.
3. Data kemudian ditransfer ke Amazon S3 dengan menggunakan AWS Transfer Family.
4. Pekerjaan AWS Glue extract, transform, and load (ETL) dapat mengubah data dan memuatnya ke bucket yang diproses dalam format yang ditentukan, atau AWS Glue dapat memasukkan data langsung ke database.
5. Amazon Athena dan Amazon QuickSight dapat digunakan untuk menanyakan dan merender data untuk mendorong analitik.

Diagram berikut menunjukkan aliran logis dari seluruh proses.

1. JCL pertama, yang disebut TABNAME, akan menggunakan utilitas Db2 DSNTIAUL untuk mengekstrak dan menghasilkan daftar tabel yang Anda rencanakan untuk dibongkar dari Db2. Untuk memilih tabel Anda, Anda harus secara manual menyesuaikan input SQL untuk memilih dan menambahkan kriteria filter untuk menyertakan satu atau lebih skema Db2.
2. JCL kedua, yang disebut REXXEXEC, akan menggunakan kerangka JCL dan program REXX yang disediakan untuk memproses daftar Tabel yang dibuat oleh JCL TABNAME dan menghasilkan satu JCL per nama tabel. Setiap JCL akan berisi satu langkah untuk membongkar tabel dan langkah lain untuk mengirim file ke bucket S3 dengan menggunakan protokol SFTP.
3. Langkah terakhir terdiri dari menjalankan JCL untuk membongkar tabel dan mentransfer file ke AWS. Seluruh proses dapat diotomatisasi menggunakan penjadwal di tempat atau di AWS.

Alat

Layanan AWS

- Amazon Athena adalah layanan kueri interaktif yang membantu Anda menganalisis data secara langsung di Amazon Simple Storage Service (Amazon S3) dengan menggunakan SQL standar.
- Amazon Aurora adalah mesin database relasional yang dikelola sepenuhnya yang dibangun untuk cloud dan kompatibel dengan MySQL dan PostgreSQL.

- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- Amazon QuickSight adalah layanan intelijen bisnis skala cloud (BI) yang membantu Anda memvisualisasikan, menganalisis, dan melaporkan data Anda dalam satu dasbor.
- Amazon Redshift adalah layanan gudang data skala petabyte terkelola di AWS Cloud.
- Amazon Relational Database Service (Amazon RDS) membantu Anda menyiapkan, mengoperasikan, dan menskalakan database relasional di AWS Cloud.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Transfer Family adalah layanan transfer aman yang memungkinkan Anda mentransfer file masuk dan keluar dari layanan penyimpanan AWS.

Alat mainframe

- SSH File Transfer Protocol (SFTP) adalah protokol transfer file aman yang memungkinkan login jarak jauh ke dan transfer file antar server. SSH menyediakan keamanan dengan mengenkripsi semua lalu lintas.
- DSNTIAUL adalah program sampel yang disediakan oleh IBM untuk membongkar data.
- DSNUTILB adalah program batch utilitas yang disediakan oleh IBM untuk membongkar data dengan opsi berbeda dari DSNTIAUL.
- z/OS OpenSSH adalah port dari Open Source Software SSH yang berjalan pada Unix System Service di bawah jaringan sistem operasi IBM. z/OS. SSH is a secure, encrypted connection program between two computers running on a TCP/IP Ini menyediakan beberapa utilitas, termasuk ssh-keygen.
- Skrip REXX (Restructured Extended Execuutor) digunakan untuk mengotomatiskan pembuatan JCL dengan langkah-langkah Db2 Unload dan SFTP.

Kode

Kode untuk pola ini tersedia di repositori GitHub unloaddb2.

Praktik terbaik

Untuk pembongkaran pertama, yang dihasilkan JCLs harus membongkar seluruh data tabel.

Setelah pembongkaran penuh pertama, lakukan pembongkaran tambahan untuk kinerja dan penghematan biaya yang lebih baik. pdate kueri SQL di dek JCL template untuk mengakomodasi perubahan apa pun pada proses pembongkaran.

Anda dapat mengonversi skema secara manual atau dengan menggunakan skrip di Lambda dengan Db2 SYSBUNCH sebagai masukan. Untuk proses industri, AWS Schema Conversion Tool (SCT) adalah opsi yang lebih disukai.

Terakhir, gunakan penjadwal berbasis mainframe atau penjadwal di AWS dengan agen di mainframe untuk membantu mengelola dan mengotomatiskan seluruh proses.

Epik

Siapkan ember S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Untuk petunjuk, lihat Membuat bucket S3 pertama Anda.	AWS Umum

Mengatur server Transfer Family

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat server berkemampuan SFTP.	Untuk membuka dan membuat server SFTP di konsol AWS Transfer Family, lakukan hal berikut: 1. Pada halaman Pilih protokol, pilih kotak centang SFTP (SSH File Transfer Protocol) — transfer file melalui Secure Shell. 2. Untuk penyedia identitas, pilih Layanan dikelola. 3. Untuk endpoint, pilih Publicly accessible.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk domain, pilih Amazon S3. 5. Pada halaman Konfigurasi detail tambahan, pertahankan pengaturan default. 6. Buat server.	
Buat peran IAM untuk Transfer Family.	Untuk membuat peran AWS Identity and Access Management (IAM) bagi Transfer Family untuk mengakses Amazon S3, ikuti petunjuk di Membuat peran dan kebijakan IAM.	Administrator AWS
Tambahkan pengguna yang dikelola layanan Amazon S3.	Untuk menambahkan pengguna yang dikelola layanan Amazon S3, ikuti petunjuk dalam dokumentasi AWS, dan gunakan ID pengguna mainframe Anda.	AWS Umum

Amankan protokol komunikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kunci SSH.	Di bawah lingkungan USS mainframe Anda, jalankan perintah berikut. `ssh-keygen -t rsa`	Pengembang mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Note Saat diminta untuk frasa sandi, biarkan kosong.	

Berikan tingkat otorisasi yang tepat ke folder SSH dan file kunci.	Secara default, kunci publik dan pribadi akan disimpan di direktori pengguna`/u/home/username/.ssh` .
	Anda harus memberikan otorisasi 644 ke file kunci dan 700 ke folder.
	`chmod 644 .ssh/id_rsa`
	`chmod 700 .ssh`
		Pengembang mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Salin konten kunci publik ke pengguna yang dikelola layanan Amazon S3 Anda.	Untuk menyalin konten kunci publik buatan AS, buka konsol AWS Transfer Family. 1. Di panel navigasi, pilih Server. 2. Pilih pengenal di kolom ID Server untuk melihat detail Server 3. Di bawah Pengguna, pilih nama pengguna untuk melihat detail Pengguna 4. Di bawah kunci publik SSH, pilih Tambahkan kunci publik SSH untuk menambahkan kunci publik ke pengguna. Untuk kunci publik SSH, masukkan kunci publik Anda. Kunci Anda divalidasi oleh layanan sebelum Anda dapat menambahkan pengguna baru Anda. 5. Pilih Tambah kunci.	Pengembang mainframe

Menghasilkan JCLs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan daftar tabel Db2 dalam lingkup.	Berikan masukan SQL untuk membuat daftar tabel yang dicakup untuk migrasi data. Langkah ini mengharuskan	Pengembang mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda untuk menentukan kriteria pemilihan quering tabel katalog Db2 SYSIBM.SYSTABLES menggunakan klausa SQL where. Filter dapat disesuaikan untuk menyertakan skema atau nama tabel tertentu yang dimulai dengan awalan tertentu atau berdasarkan stempel waktu untuk pembongkaran tambahan. Output ditangkap dalam kumpulan data sekuensial fisik (PS) pada mainframe. Dataset ini akan bertindak sebagai input untuk fase berikutnya dari generasi JCL. Sebelum Anda menggunakan JCL TABNAME (Anda dapat mengganti namanya jika perlu), buat perubahan berikut: 1. Gantikan <Jobcard>dengan kelas pekerjaan dan pengguna yang berwenang untuk menjalankan utilitas Db2. 2. Gantikan < HLQ1 > atau sesuaikan nama dataset keluaran untuk memenuhi standar situs Anda. 3. Perbarui tumpukan STEPLIB PDSEs	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	(kumpulan data yang dipartisi diperpanjang) sesuai dengan standar situs Anda. Contoh dalam pola ini menggunakan default IBM.
	4. Ganti nama PLAN dan LIB dengan nilai khusus instalasi Anda.
	5. Gantikan <Schema>dan <Prefix>dengan kriteria pilihan Anda untuk katalog Db2.
	6. Simpan JCL yang dihasilkan di pustaka PDS (kumpulan data yang dipartisi).
	7. Kirim JCL.
	
	Pekerjaan ekstraksi daftar tabel Db2
	
	`<Jobcard>`
	`//*`
	`//* UNLOAD ALL THE TABLE`
	`NAMES FOR A PARTICULAR`
	`SCHEMA`
	`//*`
	`//STEP01 EXEC PGM=IEFBR`
	`14`
	`//*`
	`//DD1 DD DISP=(MOD`
	`,DELETE,DELETE),`
	`// UNIT=SYSDA,`
	`// SPACE=(1000,`
	`(1,1)),`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`//  DSN=<HLQ1>.DSN81210.TABLIST`
	`//*`
	`//DD2  DD  DISP=(MOD,DELETE,DELETE),`
	`//  UNIT=SYSDA,`
	`//  SPACE=(1000,(1,1)),`
	`//  DSN=<HLQ1>.DSN81210.SYSPUNCH`
	`//*`
	`//UNLOAD  EXEC PGM=IKJEFT01,DYNAMNBR=20`
	`//SYSTSPRT DD`
	` SYSOUT=*`
	`//STEPLIB  DD`
	` DISP=SHR,DSN=DSNC10.DBCG.SDSNEXIT`
	`//  DD  DISP=SHR,DSN=DSNC10.SDSNLOAD`
	`//  DD  DISP=SHR,DSN=CEE.SCEERUN`
	`//  DD  DISP=SHR,DSN=DSNC10.DBCG.RUNLIB.LOAD`
	`//SYSTSIN  DD  *`
	`  DSN SYSTEM(DBCG)`
	`  RUN  PROGRAM(DSNTIAUL) PLAN(DSNTIB12) PARMS('SQL') -`
	`  LIB('DSNC10.DBCG.RUNLIB.LOAD')`
	`  END`
	`//SYSPRINT DD SYSOUT=*`
	`//*`
	`//SYSUDUMP DD SYSOUT=*`
	`//*`
	`//SYSREC00 DD DISP=(NEW,CATLG,DELETE),`
	`//  UNIT=SYSDA,SPACE=(32760,(1000,500)),`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`//  DSN=<HLQ1>.DSN81210.TABLIST`
	`//*`
	`//SYSPUNCH DD DISP=(NEW,CATLG,DELETE),`
	`//  UNIT=SYSDA,SPACE=(32760,(1000,500)),`
	`//  VOL=SER=SCR03,RECFM=FB,LRECL=120,BLKSIZE=12`
	`//  DSN=<HLQ1>.DSN81210.SYSPUNCH`
	`//*`
	`//SYSIN  DD *`
	`  SELECT CHAR(CREATOR), CHAR(NAME)`
	`  FROM SYSIBM.SYSTABLES`
	`  WHERE OWNER = '<Schema>'`
	`  AND NAME LIKE '<Prefix>%'`
	`  AND TYPE = 'T';`
	`/*`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah template JCL.	Template JCL yang disediakan dengan pola ini berisi kartu pekerjaan generik dan nama perpustakaan. Namun, sebagian besar situs mainframe akan memiliki standar penamaan sendiri untuk nama dataset, nama perpustakaan, dan kartu pekerjaan. Misalnya, kelas pekerjaan tertentu mungkin diperlukan untuk menjalankan pekerjaan Db2. Implementasi Job Entry Subsytem JES2 dan JES3 dapat memaksakan perubahan tambahan. Pustaka beban standar mungkin memiliki kualifikasi pertama yang berbeda dariSYS1, yang merupakan default IBM. Oleh karena itu, sesuaikan template untuk memperhitungkan standar spesifik situs Anda sebelum Anda menjalankannya. Buat perubahan berikut pada kerangka JCL UNLDSKEL: 1. Ubah kartu pekerjaan dengan kelas pekerjaan dan pengguna yang berwenang untuk menjalankan utilitas Db2.	Pengembang mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Kustomisasi nama dataset keluaran untuk memenuhi standar situs Anda. 3. Perbarui tumpukan STEPLIB PDSEs sesuai dengan standar situs Anda. Contoh dalam pola ini menggunakan default IBM. 4. Gantikan <DSN> dengan nama subsistem Db2 dan ID korelasi Anda. 5. Simpan JCL yang dihasilkan di pustaka PDS yang merupakan bagian dari tumpukan ISPSLIB Anda, yang merupakan pustaka template kerangka standar untuk ISPF. Bongkar dan kerangka SFTP JCL `//&USRPFX.U JOB (DB2UNLOAD),'JOB', CLASS=A,MSGCLASS=A, //  TIME=1440 ,NOTIFY=&USRPFX //* DELETE DATASETS //STEP01  EXEC PGM=IEFBR14 //DD01  DD DISP=(MOD ,DELETE,DELETE), //  UNIT=SYSD A, //  SPACE=(TR K,(1,1)),`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`// DSN=&USRPFX..DB2.P`
	`UNCH.&JOBNAME`
	`//DD02 DD DISP=(MOD`
	`,DELETE,DELETE),`
	`// UNIT=SYSD`
	`A,`
	`// SPACE=(TR`
	`K,(1,1)),`
	`// DSN=&USRPFX..DB2.U`
	`NLOAD.&JOBNAME`
	`//*`
	`//* RUNNING DB2`
	` EXTRACTION BATCH JOB`
	` FOR AWS DEMO`
	`//*`
	`//UNLD01 EXEC`
	` PGM=DSNUTILB,REGIO`
	`N=0M,`
	`// PARM='<DSN>,UNLOAD'`
	`//STEPLIB DD`
	` DISP=SHR,DSN=DSNC1`
	`0.DBCG.SDSNEXIT`
	`// DD DISP=SHR,`
	`DSN=DSNC10.SDSNLOAD`
	`//SYSPRINT DD SYSOUT=*`
	`//UTPRINT DD SYSOUT=*`
	`//SYSOUT DD SYSOUT=*`
	`//SYSPUN01 DD`
	` DISP=(NEW,CATLG,DE`
	`LETE),`
	`// SPACE=(CY`
	`L,(1,1),RLSE),`
	`// DSN=&USRPFX..DB2.P`
	`UNCH.&JOBNAME`
	`//SYSREC01 DD`
	` DISP=(NEW,CATLG,DE`
	`LETE),`
	`// SPACE=(CY`
	`L,(10,50),RLSE),`
	`// DSN=&USRPFX..DB2.U`
	`NLOAD.&JOBNAME`
	`//SYSPRINT DD SYSOUT=*`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>//SYSIN DD *
	  UNLOAD
	  DELIMITED COLDEL ','
	  FROM TABLE &TABNAME
	  UNLDDN SYSREC01
	  PUNCHDDN SYSPUN01
	  SHRLEVEL CHANGE
	 ISOLATION UR;
	 /*
	//*
	//* FTP TO AMAZON S3
	 BACKED FTP SERVER IF
	 UNLOAD WAS SUCCESSFUL
	//*
	//SFTP EXEC PGM=BPXBA
	TCH,COND=(4,LE),RE
	GION=0M
	//STDPARM DD *
	 SH cp "//'&USRP
	FX..DB2.UNLOAD.&JO
	BNAME'"
	   &TABNAME..csv;
	 echo "ascii
	 " >> uplcmd;
	 echo "PUT &TABNAME.
	.csv " >>>> uplcmd;
	 sftp -b uplcmd -i .ssh/
	id_rsa &FTPUSER.
	@&FTPSITE;
	 rm &TABNAME..csv;
	 //SYSPRINT DD SYSOUT=*
	 //STDOUT DD SYSOUT=*
	 //STDENV DD *
	 //STDERR DD SYSOUT=*</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan Mass Unload JCL.	Langkah ini melibatkan menjalankan skrip REXX di bawah lingkungan ISPF dengan menggunakan JCL. Berikan daftar tabel dalam lingkup yang dibuat pada langkah pertama sebagai masukan untuk pembuatan JCL massal terhadap nama`TABLIST DD`. JCL akan menghasilkan satu JCL baru per nama tabel dalam kumpulan data partisi yang ditentukan pengguna yang ditentukan terhadap nama tersebut. `ISPFILE DD` Alokasikan perpustak aan ini sebelumnya. Setiap JCL baru akan memiliki dua langkah: satu langkah untuk membongkar tabel Db2 ke dalam file, dan satu langkah untuk mengirim file ke ember S3. Buat perubahan berikut di JCL REXXEXEC (Anda dapat mengubah nama): 1. Gantikan `Job card user ID` dengan ID pengguna mainframe yang memiliki otoritas bongkar pada tabel. Ganti `SYSPROCISPPLIB`,`ISPSLIB`,	Pengembang mainframe

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan `ISPTLIB <HLQ1>` nilai atau sesuaikan `DSN` untuk memenuhi standar situs Anda. Untuk mengetahui nilai khusus instalasi Anda, Anda menggunakan perintah. `TSO ISRDDN` 2. Gantikan `<MFUSER>` dengan ID pengguna yang memiliki hak istimewa menjalankan pekerjaan di instalasi Anda. 3. Gantikan `<FTPUSER>` dengan ID pengguna yang memiliki hak istimewa USS dan FTP dalam instalasi Anda.. Diasumsikan bahwa ID pengguna ini dan kunci keamanan SSH-nya ada di direktori Unix Systems Services yang sesuai pada mainframe. 4. Gantikan `<AWS TransferFamily IP>` dengan alamat IP AWS Transfer Family atau nama domain. Alamat ini akan digunakan untuk langkah SFTP. 5. Kirim JCL setelah menerapkan akomodasi standar situs dan memperbarui program	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	REXX seperti yang dijelaskan di bawah ini.
	
	Pekerjaan generasi JCL massal
	
	`//RUNREXX JOB (CREATEJC L),'RUNS ISPF TABLIST', CLASS=A,MSGCLASS=A,`
	
	`//  TIME=1440 ,NOTIFY=&SYSUID`
	`//* Most of the values  required can be updated  to your site specific`
	`//* values using the  command 'TSO ISRDDN' in  your ISPF session.`
	`//* Update all the lines  tagged with //update  marker to desired`
	`//* site specific  values.`
	`//ISPF EXEC PGM=IKJEF T01,REGION=2048K,D YNAMNBR=25`
	`//SYSPROC  DD  DISP=SHR,DSN=USER. Z23D.CLIST`
	`//SYSEXEC  DD  DISP=SHR,DSN=<HLQ1 >.TEST.REXXLIB`
	`//ISPPLIB  DD  DISP=SHR,DSN=ISP.S ISPPENU`
	`//ISPSLIB  DD  DISP=SHR,DSN=ISP.S ISPSENU`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`//  DD DISP=SHR,DSN=<HLQ1>.TEST.ISPSLIB`
	`//ISPMLIB  DD DSN=ISP.SISPMENU,DISP=SHR`
	`//ISPTLIB  DD DDNAME=ISPTABL`
	`//  DD DSN=ISP.SISPTENU,DISP=SHR`
	`//ISPTABL  DD LIKE=ISP.SISPTENU,UNIT=VIO`
	`//ISPPROF  DD LIKE=ISP.SISPTENU,UNIT=VIO`
	`//ISPLOG  DD SYSOUT=*,RECFM=VA,LRECL=125`
	`//SYSPRINT  DD SYSOUT=*`
	`//SYSTSPRT  DD SYSOUT=*`
	`//SYSUDUMP  DD SYSOUT=*`
	`//SYSDBOUT  DD SYSOUT=*`
	`//SYSTSPRT  DD SYSOUT=*`
	`//SYSUDUMP  DD SYSOUT=*`
	`//SYSDBOUT  DD SYSOUT=*`
	`//SYSHELP  DD DSN=SYS1.HELP,DISP=SHR`
	`//SYSOUT  DD SYSOUT=*`
	`//* Input list of tablenames`
	`//TABLIST  DD DISP=SHR,DSN=<HLQ1>.DSN81210.TABLIST`
	`//* Output pds`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>//ISPFILE DD
	DISP=SHR,DSN=<HLQ1
	>.TEST.JOBGEN
	//SYSTSIN DD *
	ISPSTART CMD(ZSTEPS
	<MFUSER> <FTPUSER> <AWS
	TransferFamily IP>)
	/*</pre>
	Sebelum Anda menggunakan skrip REXX, buat perubahan berikut:
	1. Simpan skrip REXX di pustaka PDS yang ditentukan di bawah `SYSEXEC` tumpukan di JCL REXXEXEC yang diedit pada langkah sebelumnya dengan ZSTEPS sebagai nama anggota. Jika Anda ingin mengganti namanya, Anda harus memperbarui JCL untuk mengakomodasi kebutuhan Anda.
	2. Skrip ini menggunakan opsi jejak untuk mencetak informasi tambahan jika ada kesalahan. Sebagai gantinya, Anda dapat menambahkan kode penanganan kesalahan setelah `EXECIO`,`ISPEXEC`, dan `TSO` pernyataan, dan menghapus baris jejak.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Skrip ini menghasilkan nama anggota dengan menggunakan konvensi L0Dnnnnn penamaan, yang dapat mendukung hingga 100.000 anggota. Jika Anda memiliki lebih dari 100.000 tabel, gunakan awalan yang lebih pendek, dan sesuaikan angka dalam `tempjob` pernyataan. Skrip ZSTEPS REXX `/*REXX - - -

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>tabname =
	schema !! "." !! table
	Say tabname
	tempjob= "LOD" !!
	RIGHT("0000" !! i, 5)
	jobname=tempjob
	Say tempjob
	ADDRESS ISPEXEC
	"FTOPEN "
	ADDRESS ISPEXEC
	"FTINCL UNLDSKEL"
	/* member will be
	saved in ISPDSN library
	allocated in JCL */
	ADDRESS ISPEXEC
	"FTCLOSE NAME("tem
	pjob")"
	END
	
	ADDRESS TSO "FREE
	F(TABLIST) "
	ADDRESS TSO "FREE
	F(ISPFILE) "
	
	exit 0</pre>	

Jalankan JCLs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan langkah Bongkar Db2.	Setelah pembuatan JCL, Anda akan memiliki JCLs sebanyak mungkin tabel yang perlu dibongkar. Cerita ini menggunakan contoh yang dihasilkan JCL untuk menjelaskan struktur	Pengembang mainframe, Insinyur sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan langkah-langkah terpentin g. Tidak ada tindakan yang diperlukan di pihak Anda. Informasi berikut hanya untuk referensi. Jika niat Anda adalah mengirimkan JCLs yang telah Anda hasilkan pada langkah sebelumnya, lewati ke Kirim LODnnnnn JCLs tugas. Saat membongkar data Db2 menggunakan JCL dengan utilitas DSNUTILB Db2 yang disediakan IBM, Anda harus memastikan bahwa data yang dibongkar tidak mengandung data numerik terkompresi. Untuk mencapai ini, gunakan parameter DSNUTILB`DELIMITED` . `DELIMITED` Parameter mendukung pembongka ran data dalam format CSV dengan menambahkan karakter sebagai pembatas dan tanda kutip ganda untuk bidang teks, menghapus padding di kolom VARCHAR, dan mengubah semua bidang numerik menjadi FORMAT EKSTERNAL, termasuk bidang DATE.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Contoh berikut menunjukkan seperti apa langkah pembongkaran dalam JCL yang dihasilkan, menggunakan karakter koma sebagai pembatas. `UNLOAD DELIMITED COLDEL ',' FROM TABLE SCHEMA_NAME.TBNAME UNLDDN SYSREC01 PUNCHDDN SYSPUN01 SHRLEVEL CHANGE ISOLATION UR;`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lakukan langkah SFTP.	Untuk menggunakan protokol SFTP dari JCL, gunakan utilitas BPXBATCH. Utilitas SFTP tidak dapat mengakses kumpulan data MVS secara langsung. Anda dapat menggunakan perintah copy (`cp`) untuk menyalin file sekuensial `&USRPFX..DB2.UNLOAD.&JOBNAME` ke direktori USS, di mana ia menjadi`&TABNAME..csv` . Jalankan `sftp` perintah menggunakan kunci pribadi (`id_rsa`) dan gunakan ID pengguna RACF sebagai nama pengguna untuk terhubung ke alamat IP AWS Transfer Family. `SH cp "//'&USRPFX..DB2.UNLOAD.&JOBNAME'" &TABNAME..csv; echo "ascii " >> uplcmd; echo "PUT &TABNAME..csv " >>>> uplcmd; sftp -b uplcmd -i .ssh/id_rsa &FTPUSER.@&FTP_TF_SITE; rm &TABNAME..csv;`	Pengembang mainframe, Insinyur sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim LODnnnnn JCLs.	JCL sebelumnya telah menghasilkan semua tabel LODnnnnn JCL yang perlu dibongkar, diubah menjadi CSV, dan ditransfer ke bucket S3. Jalankan `submit` perintah pada semua JCLs yang telah dihasilkan.	Pengembang mainframe, Insinyur sistem

Sumber daya terkait

Untuk informasi selengkapnya tentang berbagai alat dan solusi yang digunakan dalam dokumen ini, lihat berikut ini:

- Panduan Pengguna z/OS OpenSSH
- Db2 z/OS - Contoh pernyataan kontrol UNLOAD
- Db2 z/OS - Membongkar file yang dibatasi
- Transfer Family - Buat server berkemampuan SFTP
- Transfer Family — Bekerja dengan pengguna yang dikelola layanan

Informasi tambahan

Setelah Anda memiliki data Db2 di Amazon S3, Anda memiliki banyak cara untuk mengembangkan wawasan baru. Karena Amazon S3 terintegrasi dengan layanan analitik data AWS, Anda dapat dengan bebas mengkonsumsi atau mengekspos data ini di sisi terdistribusi. Misalnya, Anda dapat melakukan hal berikut:

- Bangun data lake di Amazon S3, dan ekstrak wawasan berharga dengan menggunakan query-in-place, analitik, dan alat pembelajaran mesin tanpa memindahkan data.
- Memulai fungsi Lambda dengan menyiapkan alur kerja pemrosesan pasca-unggah yang terintegrasi dengan AWS Transfer Family.

- Kembangkan layanan mikro baru untuk mengakses data di Amazon S3 atau dalam database yang dikelola sepenuhnya dengan menggunakan AWS Glue, yang merupakan layanan integrasi data tanpa server yang memudahkan untuk menemukan, menyiapkan, dan menggabungkan data untuk analitik, pembelajaran mesin, dan pengembangan aplikasi.

Dalam kasus penggunaan migrasi, karena Anda dapat mentransfer data apa pun dari mainframe ke S3, Anda dapat melakukan hal berikut:

- Pensiun infrastruktur fisik, dan buat strategi pengarsipan data yang hemat biaya dengan Amazon S3 Glacier dan S3 Glacier Deep Archive.
- Buat solusi pencadangan dan pemulihan yang dapat diskalakan, tahan lama, dan aman dengan Amazon S3 dan layanan AWS lainnya, seperti S3 Glacier dan Amazon Elastic File System (Amazon EFS), untuk menambah atau mengganti kemampuan lokal yang ada.

Lebih banyak pola

- Terapkan Otomatisasi Keamanan untuk AWS WAF solusi dengan menggunakan Terraform
- Replikasi database mainframe ke AWS dengan menggunakan Exactly Connect

Manajemen

Topik

- Manajemen biaya
- Komputasi pengguna akhir
- Komputasi performa tinggi
- Awan hibrida
- Manajemen & tata kelola
- Pesan & komunikasi
- Strategi multi-akun

Manajemen biaya

Topik

- Buat laporan biaya dan penggunaan terperinci untuk pekerjaan AWS Glue dengan menggunakan AWS Cost Explorer
- Membuat laporan biaya dan penggunaan terperinci untuk klaster EMR Amazon dengan menggunakan AWS Cost Explorer
- Lebih banyak pola

Buat laporan biaya dan penggunaan terperinci untuk pekerjaan AWS Glue dengan menggunakan AWS Cost Explorer

Parijat Bhide dan Aromal Raj Jayarajan, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara melacak biaya penggunaan pekerjaan integrasi data AWS Glue dengan mengonfigurasi tag alokasi biaya yang ditentukan pengguna. Anda dapat menggunakan tag ini untuk membuat laporan biaya dan penggunaan terperinci di AWS Cost Explorer untuk pekerjaan di berbagai dimensi. Misalnya, Anda dapat melacak biaya penggunaan di tingkat tim, proyek, atau pusat biaya.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Satu atau lebih pekerjaan AWS Glue yang mengaktifkan tag yang ditentukan pengguna

Arsitektur

Tumpukan teknologi target

- AWS Glue
- AWS Cost Explorer

Diagram berikut menunjukkan bagaimana Anda dapat menerapkan tag untuk melacak biaya penggunaan untuk pekerjaan AWS Glue.

Diagram menunjukkan alur kerja berikut:

1. Insinyur data atau administrator AWS membuat tag alokasi biaya yang ditentukan pengguna untuk pekerjaan AWS Glue.
2. Administrator AWS mengaktifkan tag.
3. Tag melaporkan metadata ke AWS Cost Explorer.

Alat

- AWS Glue adalah layanan ekstrak, transformasi, dan beban (ETL) yang dikelola sepenuhnya. Ini membantu Anda mengkategorikan, membersihkan, memperkaya, dan memindahkan data dengan andal antara penyimpanan data dan aliran data.
- AWS Cost Explorer membantu Anda melihat dan menganalisis biaya dan penggunaan AWS Anda.

Epik

Membuat dan mengaktifkan tag untuk pekerjaan AWS Glue

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tag alokasi biaya yang ditentukan pengguna untuk pekerjaan AWS Glue Anda.	Untuk menambahkan tag ke pekerjaan AWS Glue yang ada 1. Masuk ke konsol AWS Management, lalu buka konsol AWS Glue. 2. Di panel navigasi kiri, di bawah ETL, pilih Pekerjaan. 3. Di bagian Pekerjaan Anda, pilih nama pekerjaan yang Anda beri tag. 4. Pilih tab Detail tugas. Kemudian, perluas bagian Advanced properties. 5. Untuk Tag, pilih Tambahkan tag baru. 6. Untuk Key, masukkan nama untuk tag Anda. 7. (Opsional) Untuk Nilai, masukkan nilai yang ingin Anda kaitkan dengan kunci.	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. (Opsional) Ulangi langkah 5-7 untuk setiap tag yang ingin Anda buat untuk pekerjaan itu. 9. Pilih Simpan. Untuk menambahkan tag ke pekerjaan AWS Glue baru 1. Buat pekerjaan AWS Glue baru berdasarkan persyaratan kasus penggunaan Anda. Untuk petunjukn ya, lihat Bekerja dengan pekerjaan di AWS Glue Console di Panduan Pengembang AWS Glue. 2. Saat Anda mengonfigurasi setelan rincian Job, ikuti langkah 4-9 dari bagian Untuk menambahkan tag ke bagian pekerjaan AWS Glue yang ada pada tugas ini. **Note** Untuk informasi selengkapnya, lihat tag AWS di AWS Glue di Panduan Pengembang AWS Glue.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan tag alokasi biaya yang ditentukan pengguna.	Ikuti petunjuk dalam Mengaktifkan tag alokasi biaya yang ditentukan pengguna di Panduan Pengguna Penagihan AWS.	Administrator AWS

Buat laporan biaya dan penggunaan untuk pekerjaan AWS Glue Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat laporan biaya dan penggunaan untuk pekerjaan AWS Glue Anda dengan menggunakan filter tag di AWS Cost Explorer.	1. Masuk ke AWS Management Console dan buka konsol AWS Cost Management. 2. Di panel navigasi kiri, pilih Laporan. 3. Pilih Buat laporan baru. 4. Untuk Pilih jenis laporan, pilih Biaya dan penggunaan (disarankan). Kemudian, pilih Create Report. 5. Untuk Filter, pilih Layanan. Dropdown Layanan muncul. 6. Pilih kotak centang di sebelah Glue. Kemudian, pilih Terapkan filter. 7. Untuk Filter, pilih Tag. Dropdown Tag muncul. 8. Pilih Tim. Kemudian, pilih kotak centang di samping tim yang telah Anda tetapkan tag. Kecualikan tim mana pun yang	AWS umum, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	belum Anda tetapkan tag. Kemudian, pilih Terapkan filter. 9. Di bagian atas grafik, pilih Tag. Kemudian, pilih tag untuk pekerjaan AWS Glue yang ingin Anda buat laporan. 10. Di bagian atas bagan, pilih dropdown 3 Bulan Terakhir dan pilih jangka waktu yang ingin dicakup oleh laporan. Kemudian, pilih dropdown Bulanan dan pilih bagaimana Anda ingin item baris dalam laporan dikumpulkan berdasarkan jangka waktu. 11. Pilih Simpan sebagai. Kemudian, masukkan judul untuk laporan Anda. 12. Pilih Simpan Laporan. Untuk informasi selengkapnya, lihat Menjelajahi data Anda menggunakan Cost Explorer di Panduan Pengguna AWS Cost Management.	

Membuat laporan biaya dan penggunaan terperinci untuk klaster EMR Amazon dengan menggunakan AWS Cost Explorer

Parijat Bhide dan Aromal Raj Jayarajan, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara melacak biaya penggunaan klaster EMR Amazon dengan mengonfigurasi tag alokasi biaya yang ditentukan pengguna. Anda dapat menggunakan tag ini untuk membuat laporan biaya dan penggunaan terperinci di AWS Cost Explorer untuk klaster di berbagai dimensi. Misalnya, Anda dapat melacak biaya penggunaan di tingkat tim, proyek, atau pusat biaya.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Satu atau lebih cluster EMR yang memiliki tag yang ditentukan pengguna diaktifkan

Arsitektur

Tumpukan teknologi target

- Amazon EMR
- AWS Cost Explorer

Arsitektur target

Diagram berikut menunjukkan bagaimana Anda dapat menerapkan tag untuk melacak biaya penggunaan untuk kluster EMR Amazon tertentu.

Diagram menunjukkan alur kerja berikut:

1. Insinyur data atau administrator AWS membuat tag alokasi biaya yang ditentukan pengguna untuk klaster EMR Amazon.
2. Administrator AWS mengaktifkan tag.
3. Tag melaporkan metadata ke AWS Cost Explorer.

Alat

Alat

- Amazon EMR adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar di AWS untuk memproses dan menganalisis data dalam jumlah besar.
- AWS Cost Explorer membantu Anda melihat dan menganalisis biaya dan penggunaan AWS Anda.

Epik

Buat dan aktifkan tag untuk kluster EMR Amazon Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tag alokasi biaya yang ditentukan pengguna untuk kluster EMR Amazon Anda.	Untuk menambahkan tag ke kluster EMR Amazon yang ada Ikuti petunjuk di Menambahkan tag ke klaster yang ada di Panduan Manajemen EMR Amazon. Untuk menambahkan tag ke kluster EMR Amazon baru Ikuti petunjuk di Tambahkan tag ke cluster baru di Panduan Manajemen EMR Amazon. Untuk informasi selengkapnya tentang cara menyiapkan klaster EMR Amazon, lihat Merencanakan dan mengonfigurasi klaster di Panduan Manajemen EMR Amazon.	Insinyur data
Aktifkan tag alokasi biaya yang ditentukan pengguna.	Ikuti petunjuk dalam Mengaktifkan tag alokasi biaya yang	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ditentukan pengguna di Panduan Pengguna Penagihan AWS.	

Buat laporan biaya dan penggunaan untuk klaster EMR Amazon Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat laporan biaya dan penggunaan untuk klaster EMR Amazon Anda dengan menggunakan filter tag di AWS Cost Explorer.	1. Masuk ke AWS Management Console dan buka konsol AWS Cost Management. 2. Di panel navigasi kiri, pilih Laporan. 3. Pilih Buat laporan baru. 4. Untuk Pilih jenis laporan, pilih Biaya dan penggunaan (disarankan). Kemudian, pilih Create Report. 5. Untuk Filter, pilih Layanan. Dropdown Layanan muncul. 6. Pilih kotak centang di sebelah EMR (Elastic MapReduce) dan EC2 - Instances (Elastic Compute Cloud — Compute). Kemudian, pilih Terapkan filter. 7. Untuk Filter, pilih Tag. Dropdown Tag muncul. 8. Pilih Tim. Kemudian, pilih kotak centang di samping tim yang telah Anda	AWS umum, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tetapkan tag. Kecualikan tim mana pun yang belum Anda tetapkan tag. Kemudian, pilih Terapkan filter. 9. Di bagian atas grafik, pilih Tag. Kemudian, pilih tag untuk kluster EMR Amazon yang ingin Anda buat laporannya. 10. Di bagian atas bagan, pilih dropdown 3 Bulan Terakhir dan pilih jangka waktu yang ingin dicakup oleh laporan. Kemudian, pilih dropdown Bulanan dan pilih bagaimana Anda ingin item baris dalam laporan dikumpulkan berdasarkan jangka waktu. 11. Pilih Simpan sebagai. Kemudian, masukkan judul untuk laporan Anda. 12. Pilih Simpan Laporan. Untuk informasi selengkapnya, lihat Menjelajahi data Anda menggunakan Cost Explorer di Panduan Pengguna AWS Cost Management.	

Lebih banyak pola

- Mengotomatiskan operasi AWS infrastruktur dengan menggunakan Amazon Bedrock
- Inventarisasi AWS sumber daya secara otomatis di beberapa akun dan Wilayah
- Otomatiskan pembuatan sumber daya WorkSpaces Aplikasi Amazon menggunakan AWS CloudFormation
- Secara otomatis mengarsipkan item ke Amazon S3 menggunakan DynamoDB TTL
- Secara otomatis berhenti dan memulai instans Amazon RDS DB menggunakan Windows AWS Systems Manager Pemeliharaan
- Buat laporan biaya dan penggunaan terperinci untuk Amazon RDS dan Amazon Aurora
- Perkirakan biaya penyimpanan untuk tabel Amazon DynamoDB
- Perkirakan biaya tabel DynamoDB untuk kapasitas sesuai permintaan
- Siapkan penskalaan otomatis berbasis peristiwa di Amazon EKS dengan menggunakan Amazon EKS Pod Identity dan KEDA
- Koordinasikan ketergantungan sumber daya dan eksekusi tugas dengan menggunakan konstruksi AWS Fargate WaitCondition kait

Komputasi pengguna akhir

Topik

- Menerapkan otentikasi SAMP 2.0 untuk Amazon WorkSpaces dengan menggunakan Auth0 dan AWS Managed Microsoft AD
- Lebih banyak pola

Menerapkan otentikasi SAMP 2.0 untuk Amazon WorkSpaces dengan menggunakan Auth0 dan AWS Managed Microsoft AD

Siva Vinnakota dan Shantanu Padhye, Amazon Web Services

Ringkasan

Pola ini mengeksplorasi bagaimana Anda dapat mengintegrasikan Auth0 AWS Directory Service for Microsoft Active Directory untuk membuat solusi otentikasi SAMP 2.0 yang kuat untuk lingkungan Amazon Anda. WorkSpaces Ini menjelaskan cara membangun federasi di antara ini Layanan AWS untuk mengaktifkan fitur-fitur canggih seperti otentikasi multi-faktor (MFA) dan alur login khusus sambil mempertahankan akses desktop yang mulus. AWS Managed Microsoft AD Baik Anda mengelola hanya segelintir pengguna atau ribuan, integrasi ini membantu memberikan fleksibilitas dan keamanan bagi organisasi Anda. Pola ini menyediakan langkah-langkah untuk proses penyiapan sehingga Anda dapat menerapkan solusi ini di lingkungan Anda sendiri.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Managed Microsoft AD
- Desktop yang disediakan di Amazon WorkSpaces Personal yang terkait dengan AWS Managed Microsoft AD
- Instans Amazon Elastic Compute Cloud (Amazon EC2)
- Akun Auth0

Batasan

Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Proses otentikasi SAMP 2.0 untuk aplikasi WorkSpaces klien terdiri dari lima langkah yang diilustrasikan dalam diagram berikut. Langkah-langkah ini mewakili alur kerja tipikal untuk masuk. Anda dapat menggunakan pendekatan terdistribusi ini untuk otentikasi setelah Anda mengikuti

petunjuk dalam pola ini, untuk membantu menyediakan metode terstruktur dan aman untuk akses pengguna.

Alur kerja:

1. Pendaftaran. Pengguna meluncurkan aplikasi klien untuk WorkSpaces dan memasukkan kode WorkSpaces registrasi untuk direktori berkemampuan SAML WorkSpaces mereka. WorkSpaces mengembalikan URL penyedia identitas Auth0 (iDP) ke aplikasi klien.
2. Masuk. WorkSpaces Klien mengalihkan ke browser web pengguna dengan menggunakan URL Auth0. Pengguna mengautentikasi dengan nama pengguna dan kata sandi mereka. Auth0 mengembalikan pernyataan SAMP ke browser klien. Pernyataan SAMP adalah token terenkripsi yang menegaskan identitas pengguna.
3. Otentikasi. Browser klien memposting pernyataan SAMP ke AWS Sign-In titik akhir untuk memvalidasinya. AWS Sign-In memungkinkan penelepon untuk mengambil peran AWS Identity and Access Management (IAM). Ini mengembalikan token yang berisi kredensi sementara untuk peran IAM.
4. WorkSpaces masuk. WorkSpaces Klien menyajikan token ke titik akhir WorkSpaces layanan. WorkSpaces menukar token untuk token sesi dan mengembalikan token sesi ke WorkSpaces klien dengan URL login. Saat WorkSpaces klien memuat halaman login. nilai nama pengguna diisi oleh `NameId` nilai yang diteruskan dalam respons SAMP.
5. Streaming. Pengguna memasukkan kata sandi mereka dan mengautentikasi terhadap WorkSpaces direktori. Setelah otentikasi, WorkSpaces mengembalikan token ke klien. Klien mengalihkan kembali ke WorkSpaces layanan dan menyajikan token. Ini menengahi sesi streaming antara WorkSpaces klien dan WorkSpace.

> **Note**
>
> Untuk menyiapkan pengalaman masuk tunggal tanpa batas yang tidak memerlukan prompt kata sandi, lihat otentikasi berbasis Sertifikat dan Pribadi dalam dokumentasi. WorkSpaces WorkSpaces

Alat

Layanan AWS

- Amazon WorkSpaces adalah layanan infrastruktur desktop virtual (VDI) yang dikelola sepenuhnya yang menyediakan desktop berbasis cloud kepada pengguna tanpa harus mendapatkan dan menyebarkan perangkat keras atau menginstal perangkat lunak yang kompleks.
- AWS Directory Service for Microsoft Active Directorymemungkinkan beban kerja dan sumber daya yang sadar direktori Anda AWS untuk menggunakan Microsoft Active Directory di file. AWS Cloud

Alat lainnya

- Auth0 adalah platform otentikasi dan otorisasi yang membantu Anda mengelola akses ke aplikasi Anda.

Epik

Konfigurasikan Konektor LDAP Direktori Aktif di Auth0

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal konektor LDAP Active Directory di Auth0 dengan. AWS Managed Microsoft AD	1. Masuk ke dashboard Auth0 dan pilih Authentication, Enterprise, Active Directory/ LDAP. Pilih Buat koneksi. 2. Berikan nama untuk koneksi Active Directory Anda, lalu pilih Buat. 3. Pada tab Pengaturan, unduh agen untuk sistem operasi Anda. Ketika instalasi selesai, browser default Anda menampilkan prompt untuk URL tiket. 4. Masukkan URL tiket penyediaan, yang harus unik untuk string koneksi Anda, lalu pilih Lanjutkan.	Administrator cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Di kotak dialog Konfigurasi LDAP AD, untuk Nama Pengguna dan Kata Sandi, masukkan kredensi administrator Anda, lalu pilih Simpan. Ketika koneksi ke Auth0 telah dibuat, log konfigurasi ditampilkan dengan status pelaporan OK untuk semua pemeriksaan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aplikasi di Auth0 untuk menghasilkan file manifes metadata SAMP.	1. Masuk ke dashboard Auth0 dan buat aplikasi baru dengan mengikuti petunjuk dalam dokumentasi Auth0. 2. Di dasbor Auth0, pilih nama aplikasi untuk mengakses pengaturan konfigurasinya. Pada AddOnstab, pilih Aplikasi SAML2 Web. 3. Pada tab Pengaturan untuk add-on, untuk URL Panggilan Balik Aplikasi, masukkan https://signin.aws.amazon.com /saml. Di sinilah token SAMP akan mengirim P0ST permintaan. 4. Pada tab Pengaturan, di kotak Pengaturan, tempel kode konfigurasi SAMP berikut dalam format JSON: `{ "audience": "https://signin.aws.amazon.com/saml", "mappings": { "email": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/sAMAccountName", "name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" },`	Administrator cloud, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"createUpnClaim": false,
	"passthroughClaimsWithNoMapping": false,
	"mapUnknownClaimsAsIs": false,
	"mapIdentities": false,
	"nameIdentifierFormat": "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent",
	"nameIdentifierProbes": [
	"http://schemas.auth0.com/sAMAccountName"
	]
	}</pre>
	5. Simpan, lalu pilih Aktifkan.
	6. Pilih tab Penggunaan, dan unduh file manifes metadata untuk penyedia identitas. Informasi ini diperlukan untuk langkah selanjutnya.
	7. Tutup jendela Aplikasi SAML2 Web.
	8. Pada layar aplikasi, pilih Koneksi. Di bawah Enterprise, pilih Directory/ LDAP konektor Aktif yang benar dan aktifkan.	

Menyiapkan IDP, peran, dan kebijakan untuk SAMP 2.0 di IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat SAMP 2.0 iDP di IAM.	Untuk mengatur SAMP 2.0 sebagai iDP, ikuti langkah-l angkah yang diuraikan dalam Buat penyedia identitas SAMP di IAM dalam dokumentasi IAM.	Administrator awan
Buat peran dan kebijakan IAM untuk federasi SAMP 2.0.	1. Buat peran IAM untuk federasi SAMP 2.0. Untuk petunjuk, lihat langkah 2 dalam petunjuk untuk menyiapkan SAMP 2.0 untuk WorkSpaces Pribadi dalam WorkSpaces dokumentasi. 2. Buat kebijakan IAM dan kaitkan dengan peran yang Anda buat di langkah sebelumny a. Untuk petunjuk, lihat langkah 3 dalam petunjuk untuk menyiapkan SAMP 2.0 untuk WorkSpaces Pribadi dalam WorkSpaces dokumentasi.	Administrator awan

Konfigurasikan pernyataan di Auth0

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pernyataan Auth0 dan SAMP.	Anda dapat menggunak an tindakan Auth0 untuk	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengonfigurasi pernyataan dalam respons SAMP 2.0. Pernyataan SAMP adalah token terenkripsi yang menegaskan identitas pengguna. 1. Masuk ke dasbor Auth0. Pilih Actions, Library, Create Action, Build dari awal. 2. Berikan nilai-nilai berikut dan kemudian pilih Buat. Nama: Tentukan nama untuk tindakan Pemicu: Pilih Login/Post Login Runtime: Pilih Node 18 3. Pada layar berikutnya, masukkan kode berikut: `exports.onExecutePostLogin = async (event, api) => { if (event.client.name === "Workspace_Saml") { const awsRole = 'arn:aws:iam::030784294031:role/Workspace_Auth0,arn:aws:iam::030784294031:saml-provider/Auth0';`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> const awsRoleSession = event.user.sAMAccountName;
	     const email = event.user.emails[0];
	     api.samlResponse.setDestination('https://signin.aws.amazon.com/saml');
	     api.samlResponse.setAttribute('https://aws.amazon.com/SAML/Attributes/Role', awsRole)
	     api.samlResponse.setAttribute('https://aws.amazon.com/SAML/Attributes/RoleSessionName', awsRoleSession)
	     api.samlResponse.setAttribute('https://aws.amazon.com/SAML/Attributes/PrincipalTag:Email', email)
	   } return;
	};</pre>
	4. Pilih Deploy.
	
	Ini melengkapi pengaturan otentikasi SAMP 2.0 untuk WorkSpaces desktop Pribadi. Bagian Arsitektur menggamba	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	rkan proses otentikasi setelah penyiapan.	

Pemecahan masalah

Isu	Solusi
Masalah otentikasi SAMP 2.0 di WorkSpaces	Jika Anda mengalami masalah saat menerapkan otentikasi SAMP 2.0 untuk WorkSpaces Personal, ikuti langkah-langkah dan tautan yang diuraikan dalam artikel AWS RE:Post tentang pemecahan masalah otentikasi SAMP 2.0. Untuk informasi tambahan tentang menyelidiki kesalahan SAMP 2.0 saat mengakses WorkSpaces, lihat: • Bagaimana cara membuat file HAR dan log Konsol dari browser saya untuk suatu AWS Dukungan kasus? (AWS Re:posting) • Lihat respons SAMP di browser Anda (dokumentasi IAM)

Sumber daya terkait

- Mengatur SAMP 2.0 untuk WorkSpaces Pribadi (WorkSpaces dokumentasi)
- Dokumentasi Auth0

Lebih banyak pola

- Otomatiskan pembuatan sumber daya WorkSpaces Aplikasi Amazon menggunakan AWS CloudFormation
- Tingkatkan kualitas panggilan di workstation agen di pusat kontak Amazon Connect
- Jalankan tugas AWS Systems Manager Automation secara sinkron dari AWS Step Functions

Komputasi performa tinggi

Topik

- Menerapkan sistem file Lustre untuk pemrosesan data berkinerja tinggi dengan menggunakan Terraform dan DRA
- Siapkan dasbor pemantauan Grafana untuk AWS ParallelCluster
- Lebih banyak pola

Menerapkan sistem file Lustre untuk pemrosesan data berkinerja tinggi dengan menggunakan Terraform dan DRA

Arun Bagal dan Ishwar Chathaiwale, Amazon Web Services

Ringkasan

Pola ini secara otomatis menerapkan sistem file Lustre AWS dan mengintegrasikannya dengan Amazon Elastic Compute Cloud (Amazon EC2) dan Amazon Simple Storage Service (Amazon S3).

Solusi ini membantu Anda mengatur lingkungan komputasi kinerja tinggi (HPC) dengan cepat dengan penyimpanan terintegrasi, sumber daya komputasi, dan akses data Amazon S3. Ini menggabungkan kemampuan penyimpanan Lustre dengan opsi komputasi fleksibel yang disediakan oleh Amazon EC2 dan penyimpanan objek yang dapat diskalakan di Amazon S3, sehingga Anda dapat mengatasi beban kerja intensif data dalam pembelajaran mesin, HPC, dan analitik data besar.

Pola menggunakan modul HashiCorp Terraform dan Amazon FSx for Lustre untuk merampingkan proses berikut:

- Penyediaan sistem file Lustre
- Membuat asosiasi repositori data (DRA) antara Lustre dan bucket S3 FSx untuk menautkan sistem file Lustre dengan objek Amazon S3
- Membuat sebuah EC2 instance
- Memasang sistem file Lustre dengan DRA terkait Amazon S3 pada instance EC2

Manfaat dari solusi ini meliputi:

- Desain modular. Anda dapat dengan mudah memelihara dan memperbarui komponen individual dari solusi ini.
- Skalabilitas. Anda dapat dengan cepat menerapkan lingkungan yang konsisten di seluruh Akun AWS atau Wilayah.
- Fleksibilitas. Anda dapat menyesuaikan penerapan agar sesuai dengan kebutuhan spesifik Anda.
- Praktik terbaik. Pola ini menggunakan modul yang telah dikonfigurasi sebelumnya yang mengikuti praktik AWS terbaik.

Untuk informasi lebih lanjut tentang sistem file Lustre, lihat situs web Lustre.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Kebijakan hak istimewa terkecil AWS Identity and Access Management (IAM) (lihat instruksi)

Batasan

FSx untuk Lustre membatasi sistem file Lustre ke satu Availability Zone, yang bisa menjadi perhatian jika Anda memiliki persyaratan ketersediaan yang tinggi. Jika Availability Zone yang berisi sistem file gagal, akses ke sistem file hilang sampai pemulihan. Untuk mencapai ketersediaan tinggi, Anda dapat menggunakan DRA untuk menautkan sistem file Lustre dengan Amazon S3, dan mentransfer data antara Availability Zones.

Versi produk

- Terraform versi 1.9.3 atau yang lebih baru
- HashiCorp AWS Provider versi 4.0.0 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan arsitektur FSx untuk Lustre dan komplementer Layanan AWS dalam. AWS Cloud

Arsitekturnya meliputi:

- Bucket S3 digunakan sebagai lokasi penyimpanan data yang tahan lama, terukur, dan hemat biaya. Integrasi antara FSx untuk Lustre dan Amazon S3 menyediakan sistem file berkinerja tinggi yang terhubung secara mulus dengan Amazon S3.
- FSx untuk Lustre menjalankan dan mengelola sistem file Lustre.
- Amazon CloudWatch Logs mengumpulkan dan memantau data log dari sistem file. Log ini memberikan wawasan tentang kinerja, kesehatan, dan aktivitas sistem file Lustre Anda.
- Amazon EC2 digunakan untuk mengakses sistem file Lustre dengan menggunakan klien Lustre open source. EC2 instance dapat mengakses sistem file dari Availability Zone lain dalam virtual private cloud (VPC) yang sama. Konfigurasi jaringan memungkinkan akses di seluruh subnet

dalam VPC. Setelah sistem file Lustre dipasang pada instance, Anda dapat bekerja dengan file dan direktorinya sama seperti Anda akan menggunakan sistem file lokal.

- AWS Key Management Service (AWS KMS) meningkatkan keamanan sistem file dengan menyediakan enkripsi untuk data saat istirahat.

Otomatisasi dan skala

Terraform memudahkan penerapan, pengelolaan, dan skala sistem file Lustre Anda di berbagai lingkungan. FSx Untuk Lustre, satu sistem file memiliki batasan ukuran, jadi Anda mungkin perlu menskalakan secara horizontal dengan membuat beberapa sistem file. Anda dapat menggunakan Terraform untuk menyediakan beberapa sistem file Lustre berdasarkan kebutuhan beban kerja Anda.

Alat

Layanan AWS

- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon FSx for Lustre memudahkan dan hemat biaya untuk meluncurkan, menjalankan, dan menskalakan sistem file Lustre berkinerja tinggi.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Kode untuk pola ini tersedia di GitHub Provision FSx for Lustre Filesystem menggunakan repositori Terraform.

Praktik terbaik

- Variabel berikut mendefinisikan sistem file Lustre. Pastikan untuk mengonfigurasi ini dengan benar berdasarkan lingkungan Anda, seperti yang diinstruksikan di bagian Epik.

- `storage_capacity`— Kapasitas penyimpanan sistem file Lustre, di. GiBs Pengaturan minimum dan default adalah 1200 GiB.
- `deployment_type`— Jenis penyebaran untuk sistem file Lustre. Untuk penjelasan tentang dua opsi, `PERSISTENT_1` dan `PERSISTENT_2` (default), lihat dokumentasi FSx untuk Lustre.
- `per_unit_storage_throughput`— Throughput baca dan tulis, dalam MBs per detik per TiB.

- `subnet_id`— ID subnet pribadi tempat Anda ingin menyebarkan FSx untuk Lustre.
- `vpc_id`— ID cloud pribadi virtual Anda di AWS tempat yang ingin Anda gunakan FSx untuk Lustre.
- `data_repository_path`— Jalur ke bucket S3 yang akan ditautkan ke sistem file Lustre.
- `iam_instance_profile`— Profil instans IAM yang digunakan untuk meluncurkan EC2 instance.
- `kms_key_id`— Nama Sumber Daya Amazon (ARN) dari AWS KMS kunci yang akan digunakan untuk enkripsi data.

- Pastikan akses dan penempatan jaringan yang tepat di dalam VPC dengan menggunakan variabel `security_group` dan`vpc_id`.
- Jalankan `terraform plan` perintah seperti yang dijelaskan di bagian Epik untuk melihat pratinjau dan memverifikasi perubahan sebelum menerapkannya. Ini membantu menangkap potensi masalah dan memastikan bahwa Anda mengetahui apa yang akan digunakan.
- Gunakan `terraform validate` perintah seperti yang dijelaskan di bagian Epics untuk memeriksa kesalahan sintaks dan untuk mengonfirmasi bahwa konfigurasi Anda benar.

Epik

Siapkan lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Terraform.	Untuk menginstal Terraform di mesin lokal Anda, ikuti instruksi dalam dokumentasi Terraform.	AWS DevOps, DevOps insinyur
Siapkan AWS kredensil.	Untuk mengatur profil AWS Command Line Interface	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(AWS CLI) untuk akun, ikuti instruksi dalam AWS dokumentasi.	
Kloning GitHub repositori.	Untuk mengkloning GitHub repositori, jalankan perintah: `git clone https://github.com/aws-samples/provision-fsx-lustre-with-terraform.git`	AWS DevOps, DevOps insinyur

Konfigurasikan dan terapkan FSx untuk Lustre

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui konfigurasi penerapan.	1. Di repositori kloning di mesin lokal Anda, navigasikan ke direktori: `fsx_deployment` `cd fsx_deployment` 2. Buka `terraform.tfvars` file, dan perbarui nilai variabel berikut: • `vpc_id` • `subnet_id` • `data_repository_path` • `iam_instance_profile`	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `kms_key_id` Untuk deskripsi variabel-variabel ini, lihat bagian Praktik terbaik. 3. Di direktori yang sama, buka `locals.tf` file dan perbarui rentang CIDR untuk variabel grup `fsx_inress` dan `fsx_egress` keamanan. 4. Jika diperlukan, buka `variables.tf` file dan perbarui nilai default dari variabel-variabel ini: • `storage_capacity` • `deployment_type` • `per_unit_storage_throughput` Untuk deskripsi variabel-variabel ini, lihat bagian Praktik terbaik.	
Inisialisasi lingkungan Terraform.	Untuk menginisialisasi lingkungan Anda untuk menjalankan `fsx_deployment` modul Terraform, jalankan: `terraform init`	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi sintaks Terraform.	Untuk memeriksa kesalahan sintaks dan untuk mengonfir masi bahwa konfigurasi Anda benar, jalankan: `terraform validate`	AWS DevOps, DevOps insinyur
Validasi konfigurasi Terraform.	Untuk membuat rencana eksekusi Terraform dan melihat pratinjau penerapan, jalankan: `terraform plan -var-file terraform.tfvars`	AWS DevOps, DevOps insinyur
Terapkan modul Terraform.	Untuk menerapkan sumber daya FSx Lustre, jalankan: `terraform apply -var-file terraform.tfvars`	AWS DevOps, DevOps insinyur

Bersihkan AWS sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus AWS sumber daya.	Setelah Anda selesai menggunakan lingkungan FSx for Lustre, Anda dapat menghapus AWS sumber daya yang digunakan oleh Terraform untuk menghindari biaya yang tidak perlu. Modul Terraform yang disediaka n dalam repositori kode	AWS DevOps, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	mengotomatiskan pembersih an ini.
	
	1. Di repositori lokal Anda, arahkan ke direktori: `fsx_deployment`
	
	`cd fsx_deployment`
	
	2. Jalankan perintah :
	
	`terraform destroy -var-file terraform.tfvars`	

Pemecahan masalah

Isu	Solusi
FSx untuk Lustre mengembalikan kesalahan.	FSx Untuk bantuan terkait masalah Lustre, lihat Memecahkan Masalah FSx Amazon for Lustre di dokumentasi for Lustre. FSx

Sumber daya terkait

- Membangun Amazon FSx untuk Lustre dengan menggunakan Terraform (Referensi AWS penyedia dalam dokumentasi Terraform)
- Memulai Amazon FSx untuk Lustre (FSx untuk dokumentasi Lustre)
- AWS posting blog tentang Amazon FSx untuk Lustre

Siapkan dasbor pemantauan Grafana untuk AWS ParallelCluster

Dario La Porta dan William Lu, Amazon Web Services

Ringkasan

AWS ParallelCluster membantu Anda menerapkan dan mengelola klaster komputasi kinerja tinggi (HPC). Ini mendukung penjadwal pekerjaan sumber terbuka AWS Batch dan Slurm. Meskipun AWS ParallelCluster terintegrasi dengan Amazon CloudWatch untuk pencatatan dan metrik, AWS tidak menyediakan dasbor pemantauan untuk beban kerja.

Dasbor Grafana untuk AWS ParallelCluster (GitHub) adalah dasbor pemantauan untuk AWS. ParallelCluster Ini memberikan wawasan penjadwal pekerjaan dan metrik pemantauan terperinci di tingkat sistem operasi (OS). Untuk informasi selengkapnya tentang dasbor yang disertakan dalam solusi ini, lihat Contoh Dasbor di repositori. GitHub Metrik ini membantu Anda lebih memahami beban kerja HPC dan kinerjanya. Namun, kode dasbor tidak diperbarui untuk AWS versi terbaru ParallelCluster atau paket open source yang digunakan dalam solusi. Pola ini meningkatkan solusi untuk memberikan manfaat berikut:

- Mendukung AWS ParallelCluster v3
- Menggunakan versi terbaru dari paket open source, termasuk Prometheus, Grafana, Prometheus Slurm Exporter, dan NVIDIA DCGM-Exporter
- Meningkatkan jumlah inti CPU dan GPUs yang digunakan pekerjaan Slurm
- Menambahkan dasbor pemantauan pekerjaan
- Meningkatkan dasbor pemantauan node GPU untuk node dengan 4 atau 8 unit pemrosesan grafis () GPUs

Versi solusi yang disempurnakan ini telah diimplementasikan dan diverifikasi di lingkungan produksi HPC pelanggan AWS.

Prasyarat dan batasan

Prasyarat

- AWS ParallelCluster CLI, diinstal dan dikonfigurasi.
- Konfigurasi jaringan yang didukung untuk AWS ParallelCluster. Pola ini menggunakan AWS ParallelCluster menggunakan dua konfigurasi subnet, yang memerlukan subnet publik, subnet pribadi, gateway internet, dan gateway NAT.

- Semua node ParallelCluster kluster AWS harus memiliki akses internet. Ini diperlukan agar skrip instalasi dapat mengunduh perangkat lunak open source dan gambar Docker.
- Sebuah key pair di Amazon Elastic Compute Cloud (Amazon EC2). Sumber daya yang memiliki key pair ini memiliki akses Secure Shell (SSH) ke head node.

Batasan

- Pola ini dirancang untuk mendukung Ubuntu 20.04 LTS. Jika Anda menggunakan versi Ubuntu yang berbeda atau jika Anda menggunakan Amazon Linux atau CentOS, maka Anda perlu memodifikasi skrip yang disediakan dengan solusi ini. Modifikasi ini tidak termasuk dalam pola ini.

Versi produk

- Ubuntu 20.04 LTS
- ParallelCluster 3.X

Pertimbangan penagihan dan biaya

- Solusi yang diterapkan dalam pola ini tidak tercakup oleh tingkat gratis. Biaya berlaku untuk Amazon EC2, Amazon FSx untuk Lustre, gateway NAT di Amazon VPC, dan Amazon Route 53.

Arsitektur

Arsitektur target

Diagram berikut menunjukkan bagaimana pengguna dapat mengakses dasbor pemantauan untuk AWS ParallelCluster di node kepala. Node kepala menjalankan NICE DCV, Prometheus, Grafana, Prometheus Slurm Exporter, Prometheus Node Exporter, dan NGINX Open Source. Node komputasi menjalankan Prometheus Node Exporter, dan mereka juga menjalankan NVIDIA DCGM-Exporter jika node berisi. GPUs Node kepala mengambil informasi dari node komputasi dan menampilkan data tersebut di dasbor Grafana.

Dalam kebanyakan kasus, node kepala tidak banyak dimuat karena penjadwal pekerjaan tidak memerlukan sejumlah besar CPU atau memori. Pengguna mengakses dasbor pada node kepala dengan menggunakan SSL pada port 443.

Semua pemirsa resmi dapat melihat dasbor pemantauan secara anonim. Hanya administrator Grafana yang dapat memodifikasi dasbor. Anda mengonfigurasi kata sandi untuk administrator Grafana dalam file. `aws-parallelcluster-monitoring/docker-compose/docker-compose.head.yml`

Alat

Layanan AWS

- NICE DCV adalah protokol tampilan jarak jauh berkinerja tinggi yang membantu Anda mengirimkan desktop jarak jauh dan streaming aplikasi dari cloud atau pusat data apa pun ke perangkat apa pun, dalam berbagai kondisi jaringan.
- AWS ParallelCluster membantu Anda menerapkan dan mengelola klaster komputasi kinerja tinggi (HPC). Ini mendukung penjadwal pekerjaan sumber terbuka AWS Batch dan Slurm.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan.

Alat-alat lainnya

- Docker adalah seperangkat produk platform as a service (PaaS) yang menggunakan virtualisasi pada tingkat sistem operasi untuk mengirimkan perangkat lunak dalam wadah.
- Grafana adalah perangkat lunak open source yang membantu Anda menanyakan, memvisualisasikan, memperingatkan, dan menjelajahi metrik, log, dan jejak.
- NGINX Open Source adalah server web open source dan reverse proxy.
- NVIDIA Data Center GPU Manager (DCGM) adalah seperangkat alat untuk mengelola dan memantau unit pemrosesan grafis pusat data NVIDIA (GPUs) di lingkungan cluster. Dalam pola ini, Anda menggunakan DCGM-Exporter, yang membantu Anda mengekspor metrik GPU dari Prometheus.
- Prometheus adalah toolkit pemantauan sistem open source yang mengumpulkan dan menyimpan metriknya sebagai data deret waktu dengan pasangan nilai kunci terkait, yang disebut label. Dalam pola ini, Anda juga menggunakan Prometheus Slurm Exporter untuk mengumpulkan dan mengekspor metrik, dan Anda menggunakan Prometheus Node Exporter untuk mengekspor metrik dari node komputasi.

- Ubuntu adalah sistem operasi berbasis Linux open source yang dirancang untuk server perusahaan, desktop, lingkungan cloud, dan IoT.

Repositori kode

Kode untuk pola ini tersedia di GitHub pcluster-monitoring-dashboardrepositori.

Epik

Buat sumber daya yang dibutuhkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	Buat bucket Amazon S3. Anda menggunakan bucket ini untuk menyimpan skrip konfigura si. Untuk petunjuknya, lihat Membuat bucket di dokumenta si Amazon S3.	AWS Umum
Kloning repositori.	Kloning GitHub pcluster-monitoring-dashboardrepo dengan menjalankan perintah berikut. `git clone https://g ithub.com/aws-samp les/parallelcluste r-monitoring-dashb oard.git`	DevOps insinyur
Buat kata sandi admin.	1. Pilih folder, pilih `aws-parallelcluster-monitoring` folder, lalu buka `docker-compose` file docker-compose.hea d.yml.	Skrip Shell Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Dalam `GF_SECURITY_ADMIN_PASSWORD` variabel, ganti `Grafana4PC!` dengan kata sandi pilihan Anda. Ini adalah kata sandi administratif yang Anda gunakan untuk mengelola akun Grafana. 3. Simpan dan tutup file docker-compose.head.yml.	
Salin file yang diperlukan ke dalam ember S3.	Salin skrip post_install.sh dan aws-parallelcluster-monitoringfolder ke bucket S3 yang Anda buat. Untuk petunjuk, lihat Mengunggah objek dalam dokumentasi Amazon S3.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan grup keamanan tambahan untuk node kepala.	1. Buat grup keamanan untuk node kepala. Grup keamanan ini akan memungkinkan lalu lintas masuk ke dasbor pemantauan di node kepala. Untuk petunjukn ya, lihat Membuat grup keamanan di dokumentasi Amazon VPC. 2. Tambahkan aturan masuk ke grup keamanan. Untuk petunjuknya, lihat Menambahkan aturan ke grup keamanan di dokumentasi Amazon VPC. Gunakan parameter berikut untuk aturan: • Jenis - HTTPS • Protokol — TCP • Rentang pelabuhan - 443 • Sumber - Masukkan alamat IP Anda • Deskripsi - Memungkin kan pengguna mengakses dasbor pemantauan	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan kebijakan IAM untuk node kepala.	Buat kebijakan berbasis identitas untuk node kepala. Kebijakan ini memungkinkan node untuk mengambil data metrik dari Amazon CloudWatch. GitHub Repo berisi contoh kebijakan. Untuk petunjuk, lihat Membuat kebijakan IAM dalam dokumentasi AWS Identity and Access Management (IAM).	Administrator AWS
Konfigurasikan kebijakan IAM untuk node komputasi.	Buat kebijakan berbasis identitas untuk node komputasi. Kebijakan ini memungkinkan node untuk membuat tag yang berisi ID pekerjaan dan pemilik pekerjaan. GitHub Repo berisi contoh kebijakan . Untuk petunjuk, lihat Membuat kebijakan IAM dalam dokumentasi IAM. Jika Anda menggunakan file contoh yang disediakan, ganti nilai berikut: • <REGION>— Wilayah AWS tempat cluster di-host • <ACCOUNT_ID>— ID akun AWS	Administrator AWS

Buat cluster

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah file template cluster yang disediakan.	Buat ParallelCluster klaster AWS. Gunakan file template CloudFormation AWS cluster.yaml yang disediakan sebagai titik awal untuk membuat cluster. Ganti nilai-nilai berikut dalam template yang disediakan: • <REGION>— Wilayah AWS tempat cluster di-host. • <HEADNODE_SUBNET>— Subnet publik dari VPC. • <ADDITIONAL_HEAD_NODE_SG>— Nama grup keamanan yang Anda buat untuk node kepala. • <KEY_NAME>— Masukkan nama EC2 key pair Amazon yang ada. Sumber daya yang memiliki key pair ini memiliki akses Secure Shell (SSH) ke head node. • <ALLOWED_IPS>-—Masukkan rentang alamat IP berformat CIDR yang diizinkan untuk membuat koneksi SSH ke node kepala. • <ADDITIONAL_HEAD_NODE_POLICY>— Masukkan nama kebijakan	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	IAM yang Anda buat untuk node kepala. • <BUCKET_NAME>— Masukkan nama bucket S3 yang Anda buat. • <COMPUTE_SUBNET> — Masukkan nama subnet pribadi di VPC. • <ADDITIONAL_COMPUTE_NODE_POLICY>— Masukkan nama kebijakan IAM yang Anda buat untuk node komputasi.	
Buat cluster.	Di AWS ParallelCluster CLI, masukkan perintah berikut. Ini menyebarkan CloudFormation template dan membuat cluster. Untuk informasi selengkapnya tentang perintah ini, lihat pcluster create-cluster di dokumentasi AWS. ParallelCluster `pcluster create-cluster -n <cluster_name> -c cluster.yaml`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pantau pembuatan cluster.	Masukkan perintah berikut untuk memantau pembuatan cluster. Untuk informasi selengkapnya tentang perintah ini, lihat pcluster describe-cluster dalam dokumentasi AWS. ParallelCluster `pcluster describe-cluster -n <cluster_name>`	Administrator AWS

Menggunakan dasbor Grafana

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Akses ke portal Grafana.	1. Masukkan perintah berikut untuk mengambil alamat IP publik dari node kepala. `pcluster describe-cluster -n <cluster_name> --query headNode.publicIpAddress` 2. Di browser web, arahkan ke URL berikut untuk mengakses dasbor Grafana. https://<head_node_public_ip_address> 3. Di halaman depan Grafana, pilih ikon Dasbor 4-persegi	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di menu sebelah kiri, lalu pilih Umum. Ini menunjukkan daftar dasbor yang dikonfigurasi. Dasbor berikut tersedia di Grafana: • Biaya Cluster - Berisi informasi tentang biaya cluster • Cluster Logs - Berisi informasi tentang log cluster • Detail Node Komputasi - Berisi informasi tentang statistik penggunaan node komputasi • Compute Node List - Berisi daftar node komputasi cluster • Node GPU - Berisi informasi tentang statistik penggunaan node GPU • Rincian Pekerjaan - Berisi informasi tentang pemanfaatan sumber daya pekerjaan • Detail Node Kepala - Berisi informasi tentang statistik penggunaan node kepala • ParallelCluster Ringkasan - Berisi informasi tentang penggunaan cluster	

Bersihkan solusi untuk berhenti mengeluarkan biaya terkait

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus klaster .	Masukkan perintah berikut untuk menghapus cluster. Untuk informasi selengkap nya tentang perintah ini, lihat pcluster delete-cluster di dokumentasi AWS. ParallelC luster `pcluster delete-cluster -n <cluster_name>`	Administrator AWS
Hapus kebijakan IAM.	Hapus kebijakan yang Anda buat untuk node kepala dan node komputasi. Untuk informasi selengkapnya tentang menghapus kebijakan , lihat Menghapus kebijakan IAM di dokumentasi IAM.	Administrator AWS
Hapus grup dan aturan keamanan.	Hapus grup keamanan yang Anda buat untuk node kepala. Untuk informasi selengkapnya, lihat Menghapus aturan grup keamanan dan Menghapus grup keamanan di dokumenta si Amazon VPC.	Administrator AWS
Hapus bucket S3.	Hapus bucket S3 yang Anda buat untuk menyimpan skrip konfigurasi. Untuk informasi selengkapnya, lihat Menghapus bucket di dokumentasi Amazon S3.	AWS Umum

Pemecahan masalah

Isu	Solusi
Node kepala tidak dapat diakses di browser.	Periksa grup keamanan dan konfirmasikan bahwa port masuk 443 terbuka.
Grafana tidak terbuka.	Pada node kepala, periksa log kontainer untuk`docker logs Grafana`.
Beberapa metrik tidak memiliki data.	Pada node kepala, periksa log kontainer dari semua kontainer.

Sumber daya terkait

Dokumentasi AWS

- Kebijakan IAM untuk Amazon EC2

Sumber daya AWS lainnya

- AWS ParallelCluster
- Dasbor pemantauan untuk AWS ParallelCluster (postingan blog AWS)

Sumber daya lainnya

- Sistem pemantauan Prometheus
- Grafana

Lebih banyak pola

- Menerapkan diagnostik Kubernetes yang didukung AI dan pemecahan masalah dengan integrasi K8SGPT dan Amazon Bedrock

Awan hibrida

Topik

- Siapkan CI/CD pipeline untuk beban kerja hybrid di Amazon ECS Anywhere dengan menggunakan AWS CDK dan GitLab
- Lebih banyak pola

Siapkan CI/CD pipeline untuk beban kerja hybrid di Amazon ECS Anywhere dengan menggunakan AWS CDK dan GitLab

Rafael Ortiz, Amazon Web Services

Ringkasan

Amazon ECS Anywhere adalah perpanjangan dari Amazon Elastic Container Service (Amazon ECS). Ini memberikan dukungan untuk mendaftarkan instans eksternal, seperti server lokal atau mesin virtual (VM), ke kluster Amazon ECS Anda. adalah fitur yang membantu mengurangi biaya dan mengurangi orkestrasi dan operasi kontainer lokal yang kompleks. Anda dapat menggunakan ECS Anywhere untuk menyebarkan dan menjalankan aplikasi kontainer di lingkungan lokal dan cloud. Ini menghilangkan kebutuhan tim Anda untuk mempelajari beberapa domain dan keahlian, atau untuk mengelola perangkat lunak yang kompleks sendiri.

Pola ini menjelaskan step-by-step pendekatan untuk menyediakan cluster Amazon ECS dengan instans Amazon ECS Anywhere dengan menggunakan tumpukan Amazon Web Services (AWS) Cloud Development Kit (AWS CDK). Anda kemudian menggunakan AWS CodePipeline untuk menyiapkan pipeline integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD). Kemudian, Anda mereplikasi repositori GitLab kode Anda ke AWS CodeCommit dan menerapkan aplikasi kontainer Anda di cluster Amazon ECS.

Pola ini dirancang untuk membantu mereka yang menggunakan infrastruktur lokal untuk menjalankan aplikasi kontainer dan menggunakannya GitLab untuk mengelola basis kode aplikasi. Anda dapat mengelola beban kerja tersebut dengan menggunakan layanan AWS Cloud, tanpa mengganggu infrastruktur lokal yang ada.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Aplikasi kontainer yang berjalan pada infrastruktur lokal.
- GitLab Repositori tempat Anda mengelola basis kode aplikasi Anda. Untuk informasi selengkapnya, lihat Repository ()GitLab.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui versi terbaru AWS CLI (dokumentasi AWS CLI).

- AWS CDK Toolkit, diinstal dan dikonfigurasi secara global. Untuk informasi selengkapnya, lihat Menginstal AWS CDK (dokumentasi AWS CDK).
- npm, diinstal dan dikonfigurasi untuk AWS CDK di. TypeScript Untuk informasi selengkapnya, lihat Mengunduh dan menginstal Node.js dan npm (dokumentasi npm).

Batasan

- Untuk batasan dan pertimbangan, lihat Instans eksternal (Amazon ECS Anywhere) di dokumentasi Amazon ECS.

Versi produk

- AWS CDK Toolkit versi 2.27.0 atau yang lebih baru
- npm versi 7.20.3 atau yang lebih baru
- Node.js versi 16.6.1 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- AWS CDK
- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon ECS Anywhere
- Amazon Elastic Container Registry (Amazon ECR)
- AWS Identity and Access Management (IAM)
- Manajer Sistem AWS
- GitLab repositori

Arsitektur target

Diagram ini mewakili dua alur kerja utama yang dijelaskan dalam pola ini, menyediakan kluster Amazon ECS dan menyiapkan CI/CD pipeline yang menyiapkan dan menyebarkan pipeline, sebagai CI/CD berikut:

1. Penyediaan cluster Amazon ECS
    a. Saat Anda menerapkan tumpukan AWS CDK pertama, itu membuat CloudFormation tumpukan di AWS.
    b. CloudFormation Tumpukan ini menyediakan kluster Amazon ECS dan sumber daya AWS terkait.
    c. Untuk mendaftarkan instans eksternal dengan kluster Amazon ECS, Anda harus menginstal AWS Systems Manager Agent (Agen SSM) di VM Anda dan mendaftarkan VM sebagai instans yang dikelola AWS Systems Manager.
    d. Anda juga harus menginstal agen penampung Amazon ECS dan Docker di VM Anda untuk mendaftarkannya sebagai instance eksternal dengan cluster Amazon ECS.
    e. Ketika instans eksternal terdaftar dan dikonfigurasi dengan cluster Amazon ECS, instans dapat menjalankan beberapa kontainer di VM Anda, yang terdaftar sebagai instance eksternal.
    f. Cluster Amazon ECS aktif dan dapat menjalankan beban kerja aplikasi melalui kontainer. Instans container Amazon ECS Anywhere berjalan di lingkungan lokal tetapi dikaitkan dengan cluster Amazon ECS di cloud.
2. Menyiapkan dan menyebarkan pipa CI/CD
    a. Saat Anda menerapkan tumpukan AWS CDK kedua, itu membuat CloudFormation tumpukan lain di AWS.
    b. CloudFormation Tumpukan ini menyediakan pipeline CodePipeline dan sumber daya AWS terkait.
    c. Anda mendorong dan menggabungkan perubahan kode aplikasi ke repositori lokal GitLab .
    d. GitLab Repositori secara otomatis direplikasi ke repositori. CodeCommit
    e. Pembaruan CodeCommit repo secara otomatis dimulai CodePipeline.
    f. CodePipeline menyalin kode dari CodeCommit dan membuat build aplikasi yang dapat diterapkan. CodeBuild
    g. CodePipeline membuat image Docker dari lingkungan CodeBuild build dan mendorongnya ke repo Amazon ECR.
    h. CodePipeline memulai CodeDeploy tindakan yang menarik image container dari repo Amazon ECR.
    i. CodePipeline menyebarkan gambar kontainer di kluster Amazon ECS.

Otomatisasi dan skala

Pola ini menggunakan AWS CDK sebagai alat infrastruktur sebagai kode (IAc) untuk mengonfigurasi dan menerapkan arsitektur ini. AWS CDK membantu Anda mengatur sumber daya AWS dan menyiapkan Amazon ECS Anywhere dan pipeline. CI/CD

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK) adalah kerangka kerja pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan infrastruktur AWS Cloud dalam kode.
- AWS CodeCommit adalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodePipeline membantu Anda memodelkan dan mengonfigurasi berbagai tahapan rilis perangkat lunak dengan cepat dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri gambar kontainer terkelola yang aman, terukur, dan andal.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster. Pola ini juga menggunakan Amazon ECS Anywhere, yang menyediakan dukungan untuk mendaftarkan server lokal atau VM ke cluster Amazon ECS Anda.

Alat lainnya

- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.
- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.
- Vagrant adalah utilitas open-source untuk membangun dan memelihara lingkungan pengembangan perangkat lunak virtual portabel. Untuk tujuan demonstrasi, pola ini menggunakan Vagrant untuk membuat VM lokal.

Repositori kode

Kode untuk pola ini tersedia di pipeline GitHub CI/CD untuk Amazon ECS Anywhere menggunakan repositori AWS CDK.

Praktik terbaik

Pertimbangkan praktik terbaik berikut saat menerapkan pola ini:

- Praktik terbaik untuk mengembangkan dan menerapkan infrastruktur cloud dengan AWS CDK
- Praktik terbaik untuk mengembangkan aplikasi cloud dengan AWS CDK (postingan blog AWS)

Epik

Verifikasi konfigurasi AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi versi AWS CDK.	Verifikasi versi AWS CDK Toolkit dengan memasukkan perintah berikut. `cdk --version` Pola ini membutuhkan versi 2.27.0 atau yang lebih baru. Jika Anda memiliki versi sebelumnya, ikuti petunjuk dalam dokumentasi AWS CDK untuk memperbaruinya.	DevOps insinyur
Verifikasi versi npm.	Verifikasi versi npm dengan memasukkan perintah berikut. `npm --version` Pola ini membutuhkan versi 7.20.3 atau yang lebih baru.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Jika Anda memiliki versi sebelumnya, ikuti petunjuk dalam dokumentasi npm untuk memperbaruinya.	

Siapkan kredensi AWS.	Siapkan kredensi AWS dengan memasukkan `aws configure` perintah dan mengikuti petunjuknya.
	`$aws configure`
	`AWS Access Key ID [None]: <your-access-key-ID>`
	`AWS Secret Access Key [None]: <your-secret-access-key>`
	`Default region name [None]: <your-Region-name>`
	`Default output format [None]:`
		DevOps insinyur

Bootstrap lingkungan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode AWS CDK.	1. Kloning pipeline CI/CD untuk Amazon ECS Anywhere menggunakan repositori AWS CDK untuk pola ini dengan memasukkan perintah berikut. `git clone https://github.com/aws-samples/amazon-ecs-any`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`where-cicd-pipelin e-cdk-sample.git`
	2. Arahkan ke direktori kloning dengan memasukkan perintah berikut.
	`cd amazon-ecs-anywher e-cicd-pipeline-cd k-sample`	

Bootstrap lingkungan.	Menerapkan CloudForm ation template ke akun dan Wilayah AWS yang ingin Anda gunakan dengan memasukkan perintah berikut. `cdk bootstrap <account-number>/<Region>` Untuk informasi selengkap nya, lihat Bootstrapping di dokumentasi AWS CDK.	DevOps insinyur

Membangun dan menerapkan infrastruktur untuk Amazon ECS Anywhere

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal dependensi paket dan kompilasi file. TypeScript	Instal dependensi paket dan kompilasi TypeScript file dengan memasukkan perintah berikut.
	`$cd EcsAnywhereCdk`
	`$npm install`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`$npm fund` Perintah ini menginstal semua paket dari repositori sampel. Untuk informasi selengkapnya, lihat npm ci dan npm install di dokumentasi npm. Jika Anda mendapatkan kesalahan tentang paket yang hilang saat Anda memasukkan perintah ini, lihat bagian Pemecahan Masalah dari pola ini.	
Bangun proyek.	Untuk membangun kode proyek, masukkan perintah berikut. `npm run build` Untuk informasi selengkap nya tentang membangun dan menerapkan project, lihat aplikasi AWS CDK pertama Anda di dokumentasi AWS CDK.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Terapkan tumpukan infrastruktur Amazon ECS Anywhere.	1. Buat daftar tumpukan dengan memasukkan perintah berikut.
	`$cdk list`
	2. Konfirmasikan bahwa output mengembalikan `EcsAnywhereInfraStack` dan `ECSAnywherePipelineStack` tumpukan.
	3. Menyebarkan `EcsAnywhereInfraStack` tumpukan dengan memasukkan perintah berikut.
	`$cdk deploy EcsAnywhereInfraStack`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi pembuatan dan output tumpukan.	1. Masuk ke AWS Management Console dan buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation/. 2. Pada halaman Stacks, pilih `EcsAnywhereInfraStack` tumpukan. 3. Konfirmasikan bahwa status tumpukan adalah salah satu `CREATE_IN_PROGRESS` atau`CREATE_COMPLETE` . Menyiapkan cluster Amazon ECS dapat memakan waktu. Jangan lanjutkan sampai pembuatan tumpukan selesai.	DevOps insinyur

Mengatur VM lokal

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan VM Anda.	Buat VM Vagrant dengan memasukkan `vagrant up` perintah dari direktori root tempat Vagrantfile berada. Untuk informasi lebih lanjut, lihat dokumentasi Vagrant.	DevOps insinyur
Daftarkan VM Anda sebagai instance eksternal.	1. Masuk ke Vagrant VM dengan menggunakan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	perintah. `vagrant ssh` Untuk informasi lebih lanjut, lihat dokumentasi Vagrant. 2. Instal AWS CLI di VM dengan mengikuti petunjuk penginstalan AWS CLI dan memasukkan perintah berikut. `$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" \ > -o "awscliv2.zip" $sudo apt install unzip $unzip awscliv2.zip $sudo ./aws/install $aws configure AWS Access Key ID [None]: <your-access-key-ID> AWS Secret Access Key [None]: <your-secret-access-key> Default region name [None]: <your-Region-name> Default output format [None]:` 1. Buat kode aktivasi dan ID yang dapat Anda gunakan untuk mendaftarkan VM Anda dengan AWS Systems Manager dan untuk mengaktifkan instans	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	eksternal Anda. Output dari perintah ini mencakup ID aktivasi dan nilai kode aktivasi.
	
	`aws ssm create-activation \ > --iam-role EcsAnywhereInstanceRole \ > | tee ssm-activation.json`
	
	Jika Anda menerima kesalahan saat menjalankan perintah ini, lihat bagian Pemecahan Masalah.
	2. Ekspor ID aktivasi dan nilai kode.
	
	`export ACTIVATION_ID=<activation-ID> export ACTIVATION_CODE=<activation-code>`
	
	3. Unduh skrip instalasi ke VM Anda.
	
	`curl --proto "https" -o "ecs-anywhere-install.sh" \ > "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh"`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Jalankan skrip instalasi di VM Anda. `sudo bash ecs-anywhere-install.sh \ --cluster EcsAnywhereCluster \ --activation-id $ACTIVATION_ID \ --activation-code $ACTIVATION_CODE \ --region <region-name>` Ini mengatur VM Anda sebagai instans eksternal Amazon ECS Anywhere dan mendaftarkan instans di cluster Amazon ECS. Untuk informasi selengkapnya, lihat Mendaftarkan instance eksternal ke klaster di dokumentasi Amazon ECS. Jika Anda mengalami masalah apa pun, lihat bagian Pemecahan Masalah.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi status Amazon ECS Anywhere dan VM eksternal.	Untuk memverifikasi apakah VM Anda terhubung ke bidang kontrol Amazon ECS dan berjalan, gunakan perintah berikut.
	`$aws ssm describe-instance-information`
	`$aws ecs list-container-instances --cluster $CLUSTER_NAME`
		DevOps insinyur

Menyebarkan pipa CI/CD

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cabang di CodeCommit repo.	Buat cabang bernama `main` dalam CodeCommit repo dengan membuat komit pertama untuk repositori. Anda dapat mengikuti dokumentasi AWS untuk Membuat komit CodeCommit. Berikut adalah contoh perintah tersebut. `aws codecommit put-file \ --repository-name EcsAnywhereRepo \ --branch-name main \ --file-path README.md \ --file-content "Test" \ --name "Dev Ops" \`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--email "devops@example.com" \ --commit-message "Adding README."`	
Siapkan pencerminan repo.	Anda dapat mencerminkan GitLab repositori ke dan dari sumber eksternal. Anda dapat memilih repositori mana yang berfungsi sebagai sumber. Cabang, tag, dan komit disinkronkan secara otomatis. Siapkan cermin dorong antara GitLab repositori yang menghosting aplikasi Anda dan repositori. CodeCommit Untuk petunjuk, lihat Mengatur cermin dorong dari GitLab ke CodeCommit (GitLab dokumentasi). **Note** Secara default, mirroring secara otomatis menyinkronkan repositori. Jika Anda ingin memperbarui repositori secara manual, lihat Memperbarui cermin (GitLab dokumentasi).	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan tumpukan CI/CD pipa.	Menyebarkan `EcsAnywhe rePipelineStack` tumpukan dengan memasukkan perintah berikut. `$cdk  deploy EcsAnywhe rePipelineStack`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji CI/CD pipa.	1. Buat perubahan kode aplikasi dan dorong ke sumber, GitLab repo lokal. Misalnya, edit `../application/index.html` file untuk memperbarui nilai versi aplikasi. 2. Ketika kode direplikasi ke CodeCommit repo, ini memulai pipeline. CI/CD Lakukan salah satu tindakan berikut: • Jika Anda menggunakan pencerminan otomatis untuk menyinkronkan GitLab repo dengan CodeCommit repo, lanjutkan ke langkah berikutnya. • Jika Anda menggunakan pencerminan manual, dorong perubahan kode aplikasi ke CodeCommit repo dengan mengikuti instruksi di Perbarui cermin (GitLab dokumentasi). 3. Di mesin lokal Anda, di browser web, masukkan http://localhost:80. Ini membuka halaman web NGINX karena port 80 diteruskan ke localhost di Vagrantfile. Konfirmasikan	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bahwa Anda dapat melihat nilai versi aplikasi yang diperbarui. Ini memvalida si pipeline dan penerapan gambar. 4. (Opsional) Jika Anda ingin memverifikasi penerapan di AWS Managemen t Console, lakukan hal berikut: a. Buka konsol Amazon ECS di https://console.a ws.amazon.com/ecs/. b. Dari bilah navigasi, pilih Wilayah untuk digunakan . c. Di panel navigasi, pilih Klaster. d. Pada halaman Clusters, pilih EcsAnywhe reClustercluster. e. Pilih Definisi Tugas. f. Konfirmasikan wadah sedang berjalan.	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan dan hapus sumber daya.	Setelah Anda berjalan melalui pola ini, Anda harus menghapus proof-of-concept sumber daya yang Anda	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	buat. Untuk membersihkan, masukkan perintah berikut.
	`$cdk destroy EcsAnywhe rePipelineStack`
	`$cdk destroy EcsAnywhe reInfraStack`	

Pemecahan masalah

Isu	Solusi

Kesalahan tentang paket yang hilang saat menginstal dependensi paket.	Masukkan salah satu perintah berikut untuk menyelesaikan paket yang hilang.
	`$npm ci`
	atau
	`$npm install -g @aws-cdk/<package_ name>`

Ketika Anda menjalankan `aws ssm create-activation` perintah pada VM, Anda menerima kesalahan berikut. `An error occurred (Validati onException) when calling the CreateActivation operation: Nonexistent role or missing ssm service principal in trust policy: arn:aws:iam::000000000000:role/ EcsAnywhereInstanceRole`	`EcsAnywhereInfraStack` Tumpukan belum sepenuhnya digunakan, dan peran IAM yang diperlukan untuk menjalankan perintah ini belum dibuat. Periksa status tumpukan di CloudFormation konsol. Coba lagi perintah setelah status berubah menjadi`CREATE_CO MPLETE` .

Isu	Solusi

Pemeriksaan kesehatan Amazon ECS kembali`UNHEALTHY` , dan Anda melihat kesalahan berikut di bagian Layanan klaster di konsol Amazon ECS.
 	
`service EcsAnywhereService was unable to place a task because no container instance met all of its requirements. Reason: No Container Instances were found in your cluster.`
	Mulai ulang agen Amazon ECS di Vagrant VM Anda dengan memasukkan perintah berikut.
	
	`$vagrant ssh`
	`$sudo systemctl restart ecs`
	`$sudo systemctl status ecs`

Sumber daya terkait

- Halaman pemasaran Amazon ECS Anywhere
- Dokumentasi Amazon ECS Anywhere
- Amazon ECS Anywhere demo (video)
- Sampel GitHublokakarya Amazon ECS Anywhere ()
- Pencerminan repositori (dokumentasi) GitLab

Lebih banyak pola

- Mengotomatiskan pengaturan peering Antar wilayah dengan AWS Transit Gateway
- Menerapkan aplikasi kontainer saat AWS IoT Greengrass V2 berjalan sebagai wadah Docker
- Mengelola aplikasi kontainer lokal dengan menyiapkan Amazon ECS Anywhere dengan AWS CDK
- Migrasi data Hadoop ke Amazon S3 menggunakan Migrator WANdisco LiveData
- Ubah header HTTP saat Anda bermigrasi dari F5 ke Application Load Balancer di AWS
- Menghosting ulang beban kerja lokal di AWS Cloud: daftar periksa migrasi
- Gunakan kueri BMC Discovery untuk mengekstrak data migrasi untuk perencanaan migrasi

Manajemen & tata kelola

Topik

- Identifikasi dan beri tahu kapan sumber daya Amazon Data Firehose tidak dienkripsi dengan kunci AWS KMS
- Otomatiskan alokasi Amazon VPC IPv4 IPAM CIDR untuk yang baru dengan menggunakan AFT Akun AWS
- Otomatis menambahkan atau memperbarui entri registri Windows menggunakan AWS Systems Manager
- Secara otomatis membuat RFC di AMS menggunakan Python
- Secara otomatis berhenti dan memulai instans Amazon RDS DB menggunakan Windows AWS Systems Manager Pemeliharaan
- Memusatkan distribusi paket perangkat lunak di AWS Organizations dengan menggunakan Terraform
- Konfigurasikan logging untuk aplikasi.NET di Amazon CloudWatch Logs dengan menggunakan NLog
- Salin produk AWS Service Catalog di berbagai akun AWS dan Wilayah AWS
- Buat matriks RACI atau RASCI untuk model operasi cloud
- Buat alarm untuk metrik kustom menggunakan deteksi anomali Amazon CloudWatch
- Buat AWS Cloud9 IDE yang menggunakan volume Amazon EBS dengan enkripsi default
- Buat CloudWatch dasbor Amazon berbasis tag secara otomatis
- Dokumentasikan desain AWS landing zone
- Tingkatkan kinerja operasional dengan mengaktifkan Amazon DevOps Guru di beberapa Wilayah AWS, akun, dan OUs dengan AWS CDK
- Mengatur set izin untuk beberapa akun dengan menggunakan Account Factory untuk Terraform
- Implementasikan Account Factory for Terraform (AFT) dengan menggunakan pipeline bootstrap
- Kelola produk AWS Service Catalog di beberapa akun AWS dan Wilayah AWS
- Pantau cluster SAP RHEL Pacemaker dengan menggunakan layanan AWS
- Memantau aktivitas aplikasi dengan menggunakan Wawasan CloudWatch Log
- Pantau penggunaan Gambar Mesin Amazon bersama di beberapa Akun AWS
- Lihat detail snapshot EBS untuk akun atau organisasi AWS
- Lebih banyak pola

Identifikasi dan beri tahu kapan sumber daya Amazon Data Firehose tidak dienkripsi dengan kunci AWS KMS

Ram Kandaswamy, Amazon Web Services

Ringkasan

Untuk kepatuhan, beberapa organisasi harus mengaktifkan enkripsi pada sumber daya pengiriman data seperti Amazon Data Firehose. Pola ini menunjukkan cara untuk memantau, mendeteksi, dan memberi tahu ketika sumber daya tidak sesuai.

Untuk mempertahankan persyaratan enkripsi, pola ini dapat digunakan AWS untuk menyediakan pemantauan dan deteksi otomatis sumber daya pengiriman Amazon Data Firehose yang tidak dienkripsi dengan kunci (). AWS Key Management Service AWS KMS Solusinya mengirimkan pemberitahuan peringatan, dan dapat diperluas untuk melakukan remediasi otomatis. Solusi ini dapat diterapkan ke akun individu atau lingkungan multi-akun, seperti lingkungan yang menggunakan AWS landing zone atau. AWS Control Tower

Prasyarat dan batasan

Prasyarat

- Aliran pengiriman Amazon Data Firehose
- Izin dan keakraban yang memadai CloudFormation, yang digunakan dalam otomatisasi infrastruktur ini

Batasan

- Solusinya tidak real time karena menggunakan AWS CloudTrail peristiwa untuk deteksi, dan ada penundaan antara waktu sumber daya yang tidak terenkripsi dibuat dan pemberitahuan dikirim.

Arsitektur

Tumpukan teknologi target

Solusinya menggunakan teknologi tanpa server dan layanan berikut:

- AWS CloudTrail

- Amazon CloudWatch
- AWS Command Line Interface (AWS CLI)
- AWS Identity and Access Management (IAM)
- Amazon Data Firehose
- AWS Lambda
- Amazon Simple Notification Service (Amazon SNS)

Arsitektur target

Diagram menggambarkan langkah-langkah ini:

1. Pengguna membuat atau memodifikasi Amazon Data Firehose.
2. Sebuah CloudTrail peristiwa terdeteksi dan dicocokkan.
3. Lambda dipanggil.
4. Sumber daya yang tidak sesuai diidentifikasi.
5. Pemberitahuan email dikirim.

Otomatisasi dan skala

Anda dapat menggunakan CloudFormation StackSets untuk menerapkan solusi ini ke beberapa Wilayah AWS atau akun dengan satu perintah.

Alat

- AWS CloudTrailadalah sebuah Layanan AWS yang membantu Anda mengaktifkan tata kelola, kepatuhan, dan audit operasional dan risiko Anda. Akun AWS Tindakan yang diambil oleh pengguna, peran, atau Layanan AWS direkam sebagai peristiwa di CloudTrail. Peristiwa mencakup tindakan yang diambil dalam operasi Konsol Manajemen AWS AWS CLI, AWS SDKs,, dan API.
- Amazon CloudWatch Events memberikan aliran peristiwa sistem yang mendekati waktu nyata yang menjelaskan perubahan AWS sumber daya.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang memungkinkan Anda untuk berinteraksi Layanan AWS dengan menggunakan perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke AWS sumber daya dengan aman. Anda menggunakan IAM untuk mengontrol

siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan sumber daya.

- Amazon Data Firehose adalah layanan yang dikelola sepenuhnya untuk mengirimkan data streaming waktu nyata. Dengan Firehose, Anda tidak perlu menulis aplikasi atau mengelola sumber daya. Anda mengonfigurasi produsen data Anda untuk mengirim data ke Firehose, dan secara otomatis mengirimkan data ke tujuan yang Anda tentukan.
- AWS Lambdaadalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda konsumsi—tidak ada biaya ketika kode Anda tidak berjalan.
- Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan (juga dikenal sebagai produsen dan konsumen).

Epik

Menegakkan enkripsi untuk kepatuhan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation StackSets.	Dalam AWS CLI, gunakan `firehose-encryption-checker.yaml` template (terlampir) untuk membuat stack set dengan menjalank an perintah berikut. Berikan topik Amazon SNS yang valid Nama Sumber Daya Amazon (ARN) untuk parameter tersebut. Penerapan harus berhasil membuat aturan CloudWatch Acara, fungsi Lambda, dan peran IAM dengan izin yang diperlukan, seperti yang dijelaskan dalam template.	Arsitek awan, Administrator sistem

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws cloudformation create-stack-set --stack-set-name my-stack-set --template-body file://firehose-encryption-checker.yaml`	
Buat instance tumpukan.	Tumpukan dapat dibuat sesuai pilihan Anda dan juga di satu atau lebih akun. Wilayah AWS Untuk membuat instance tumpukan, jalankan perintah berikut. Ganti nama tumpukan, nomor akun, dan Wilayah dengan nama Anda sendiri. `aws cloudformation create-stack-instances --stack-set-name my-stack-set --accounts 123456789012 223456789012 --regions us-east-1 us-east-2 us-west-1 us-west-2 --operation-preferences FailureToleranceCount=1`	Arsitek awan, Administrator sistem

Sumber daya terkait

- Bekerja dengan CloudFormation StackSets
- Apa itu CloudWatch Acara Amazon?

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Otomatiskan alokasi Amazon VPC IPv4 IPAM CIDR untuk yang baru dengan menggunakan AFT Akun AWS

Kien Pham dan Alex Pazik, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara mengotomatiskan alokasi CIDR Amazon VPC IP Address Manager (IPAM IPv4) untuk yang baru dengan Akun AWS menggunakan Account AWS Control Tower Factory for Terraform (AFT). Ini dilakukan dengan menggunakan kustomisasi tingkat akun yang mengalokasikan blok IPv4 CIDR dari IPAM ke virtual private cloud (VPC) baru menggunakan modul. `aft-account-customizations`

Dengan IPAM, Anda dapat mengatur, menetapkan, memantau, dan mengaudit alamat IP dalam skala besar, memungkinkan Anda untuk dengan mudah merencanakan, melacak, dan memantau alamat IP untuk beban kerja Anda AWS . Anda dapat membuat kolam IPAM dan IPAM untuk digunakan untuk mengalokasikan blok IPv4 CIDR ke VPC baru selama proses penjual akun.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan AWS Control Tower diaktifkan di didukung Wilayah AWSdan AFT dikerahkan
- Penyedia sistem kontrol versi (VCS) yang didukung seperti BitBucket, GitHub, dan Enterprise GitHub
- Antarmuka Baris Perintah Terraform (CLI) diinstal
- Lingkungan runtime tempat Anda dapat menjalankan modul Terraform yang menginstal AFT
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Versi produk

- AWS Control Tower landing zone versi 3.0 atau lebih baru, lebih awal dari versi 4.0
- AFT versi 1.13.0 atau yang lebih baru, lebih awal dari versi 2.0.0
- Terraform OSS versi 1.2.0 atau yang lebih baru, lebih awal dari versi 2.0.0
- Terraform AWS Provider (`terraform-provider-aws`) versi 5.11.0 atau yang lebih baru, lebih awal dari versi 6.0.0
- Modul Terraform untuk IPAM (`aws-ia/ipam/aws`) versi 2.1.0 atau yang lebih baru

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen pola ini.

Alur kerja terdiri dari tugas-tugas utama berikut:

1. Memicu perubahan - Perubahan pada kustomisasi Terraform dan IPAM berkomitmen ke GitHub repositori dan didorong. Tugas ini memicu AWS CodeBuild pipeline secara otomatis.
2. Otomatiskan build — Di dalam CodeBuild, beberapa proyek build dipicu AWS Step Functions.
3. Terapkan kustomisasi — Step Functions berkoordinasi CodeBuild dengan merencanakan dan menerapkan perubahan Terraform. Tugas ini menggunakan modul AFT Terraform untuk mengoordinasikan penetapan IP kumpulan IPAM ke akun penjual otomatis. AWS

Alat

Layanan AWS

- AWS CodeBuildadalah layanan build yang dikelola sepenuhnya yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS Control Towermengatur kemampuan beberapa lainnya, termasuk Layanan AWS AWS Organizations,, dan. AWS Service Catalog AWS IAM Identity Center Ini dapat membantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.

- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS
- AWS Service Catalogmembantu Anda mengelola katalog layanan TI secara terpusat yang disetujui. AWS Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS Amazon VPC IP Address Manager (IPAM) adalah fitur VPC yang memudahkan Anda merencanakan, melacak, dan memantau alamat IP untuk beban kerja Anda. AWS

Alat-alat lainnya

- GitHubadalah platform pengembang yang dapat digunakan pengembang untuk membuat, menyimpan, mengelola, dan membagikan kode mereka.
- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal. Ini termasuk komponen tingkat rendah seperti instance komputasi, penyimpanan, dan jaringan dan komponen tingkat tinggi seperti entri DNS dan fitur perangkat lunak layanan (SaaS).
- Python adalah bahasa pemrograman komputer tujuan umum. Anda dapat menggunakannya untuk membangun aplikasi, mengotomatiskan tugas, dan mengembangkan layanan di AWS Cloudaplikasi.

Repositori kode

- Kode untuk pola ini tersedia di GitHub AWS Control Tower Account Factory for Terraform repositori.

Praktik terbaik

Saat Anda menerapkan AFT, kami menyarankan Anda mengikuti praktik terbaik untuk membantu memastikan implementasi yang aman, efisien, dan berhasil. Pedoman dan rekomendasi utama untuk mengimplementasikan dan mengoperasikan AFT meliputi:

- Tinjauan masukan secara menyeluruh — Tinjau dan pahami setiap masukan dengan cermat. Konfigurasi input yang benar sangat penting untuk pengaturan dan fungsi AFT.
- Pembaruan template reguler - Simpan template diperbarui dengan AWS fitur terbaru dan versi Terraform. Pembaruan rutin membantu Anda memanfaatkan fungsionalitas baru dan menjaga keamanan.
- Versioning — Sematkan versi modul AFT Anda dan gunakan penerapan AFT terpisah untuk pengujian jika memungkinkan.
- Cakupan — Gunakan AFT hanya untuk menyebarkan pagar pembatas infrastruktur dan penyesuaian. Jangan menggunakannya untuk menyebarkan aplikasi Anda.
- Linting dan validasi — Pipeline AFT memerlukan konfigurasi Terraform yang dilinted dan divalidasi. Jalankan lint, validasi, dan uji sebelum mendorong konfigurasi ke repositori AFT.
- Modul Terraform — Buat kode Terraform yang dapat digunakan kembali sebagai modul, dan selalu tentukan versi Terraform dan AWS penyedia agar sesuai dengan kebutuhan organisasi Anda.

Epik

Siapkan dan konfigurasikan AWS lingkungan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan AWS Control Tower.	Siapkan dan konfigurasikan AWS Control Tower di AWS lingkungan Anda untuk memastikan manajemen dan tata kelola terpusat Anda. Akun AWS Untuk informasi selengkapnya, lihat Memulai AWS Control Tower dalam AWS Control Tower dokumentasi.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan AWS Control Tower Account Factory untuk Terraform (AFT).	Siapkan AFT di akun manajemen AFT khusus yang baru. Untuk informasi selengkapnya, lihat C onfigure dan luncurkan AWS Control Tower Account Factory for Terraform Anda di dokumenta si. AWS Control Tower	Administrator awan
Selesaikan pasca-penyebaran AFT.	Setelah penerapan infrastru ktur AFT selesai, selesaika n langkah-langkah dalam langkah-langkah pasca-pen erapan dalam dokumentasi. AWS Control Tower	Administrator awan

Buat IPAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Delegasikan administrator IPAM.	Untuk mendelegasikan akun administrator IPAM di AWS organisasi Anda, gunakan langkah-langkah berikut: 1. Menggunakan akun AWS Organizations manajemen, buka konsol IPAM di https:// console.aws.amazon.com/ ipam/. 2. Di Konsol Manajemen AWS, pilih Wilayah AWS di mana Anda ingin bekerja dengan IPAM.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Di panel navigasi, pilih Pengaturan organisasi.
	4. Pilih Delegasikan. Opsi Delegasi hanya tersedia jika Anda masuk ke konsol sebagai akun AWS Organizations manajemen.
	5. Masukkan Akun AWS ID untuk akun IPAM. Administrator IPAM harus berada di akun AWS Organizations anggota.
	6. Pilih Simpan perubahan.
	
	Atau, Anda dapat menggunakan AWS CLI dan menjalankan perintah berikut:
	
	`aws ec2 enable-ipam-organization-admin-account \`
	`--delegated-admin-account-id 012345678901`
	
	Untuk informasi selengkapnya, lihat Mengintegrasikan IPAM dengan akun di AWS organisasi dalam dokumentasi VPC Amazon enable-ipam-organization-admindan - account di AWS CLI Referensi Perintah.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important Untuk terus menggunakan IPAM, Anda harus masuk ke akun administr ator yang didelegas ikan. Profil SSO atau variabel AWS lingkungan yang ditentukan pada langkah berikutnya harus memungkin kan Anda masuk ke akun tersebut dan memberikan izin untuk membuat kumpulan tingkat atas dan regional IPAM.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat kolam tingkat atas dan regional IPAM.	GitHub Repositori pola ini berisi template Terraform yang dapat Anda gunakan untuk membuat kumpulan tingkat atas IPAM dan kumpulan regional Anda. Kemudian Anda dapat berbagi pool dengan organisasi, unit organisasi (OU) Akun AWS, atau sumber daya lainnya dengan menggunakan AWS Resource Access Manager (AWS RAM).
	
	Gunakan langkah-langkah berikut:
	
	1. Jalankan perintah berikut:
	
	`# Navigate to the IPAM module`
	`cd ipam/terraform`
	
	`# Initiate the IPAM module`
	`terraform init`
	
	2. Buka `main.tf` file, ubah kode seperti yang dijelaskan dalam komentar berikut, dan simpan file.
	
	`terraform { required_providers { aws = { source = "hashicorp/aws"`
		Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> version = ">= 5.11.0, < 6.0.0"
	    }
	  }
	}
	
	provider "aws" {
	  region = "us-east-1" # Replace with the region to create your top-level pool
	}
	
	module "ipam" {
	  source = "aws-ia/ipam/aws"
	  version = ">= 2.1.0"
	
	  top_name = "my-top-level-pool" # Replace with your desired top-level pool name
	  top_description = "top-level pool" # Replace with your desired top-level level pool description
	  top_cidr = ["10.0.0.0/16"] # Replace with your desired top-level pool CIDR
	
	  pool_configurations = {
	    my-regional-pool = { # (Optional) Replace with a different resource name
	      name = "my-regional-pool"</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> # Replace with your
	  desired pool name
	       description
	       = "regional
	  pool" # Replace with
	  your desired pool
	  description
	       cidr
	    = ["10.0.0.0/23"]
	  # Replace with your
	  desired pool CIDR
	       locale
	      = "us-east-1"
	  # Replace with your
	  desired pool locale
	       ram_share
	_principals =
	  ["arn:aws:organiza
	tions::01234567890
	1:ou/ou-ab1c2de345
	/ou-ab1c2de345"]
	  # Replace with your
	  desired principal
	  ARN to share with
	  via Resource Access
	  Manager (RAM)
	     }
	   }
	}
	
	output "my_regio
	nal_pool_id" {
	   description = "The
	  ID of the regional
	  pool"
	   value =
	  module.ipam.pools_
	level_1["my-region
	al-pool"].id #
	  Replace with your
	  desired resource name
	  if changed above</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>}
	
	# Create the IPAM
	 top-level and
	 resource pool
	terraform apply</pre>
	Catat ID kumpulan sumber daya yang dihasilkan setelah pembuatan. Anda akan memerlukan ID saat mengirimkan permintaa n akun. Jika Anda lupa ID kumpulan sumber daya, Anda bisa mendapatkannya nanti dari file Konsol Manajemen AWS.
	
	Important
	Pastikan bahwa kolam yang dibuat CIDRs tidak tumpang tindih dengan kolam lain di Wilayah kerja Anda. Anda dapat membuat pool tanpa CIDR, tetapi Anda tidak akan dapat menggunak an pool untuk alokasi sampai Anda telah menyediakan CIDR untuk itu. Anda dapat menambahkan CIDRs ke kolam kapan saja	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan mengedit kolam.	

Integrasikan IPAM dengan AFT

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Mulailah membuat kustomisasi akun.	Untuk memulai kustomisasi akun baru, jalankan perintah berikut dari terminal Anda:
	`# Default name for customization repo`
	`cd aft-account-customizations # Replace with your actual repo name if different than the default`
	`mkdir -p APG-AFT-IPAM/terraform # Replace APG-AFT-IPAM with your desired customization name`
	`cd APG-AFT-IPAM/terraform`
		DevOps insinyur

Buat `aft-providers.jinja` file.	Tambahkan kode dinamis ke `aft-providers.jinja` file yang menentukan backend dan penyedia Terraform untuk digunakan. Gunakan langkah-langkah berikut:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Buat `aft-providers.jinja` file baru di `terraform` direktori. 2. Buka file, tambahkan kode berikut, dan simpan file. `provider "aws" { region = "{{ provider_region }}" assume_role { role_arn = "{{ target_admin_role_arn }}" } default_tags { tags = { managed_by = "AFT" } } }`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat `backend.jinja` file.	Tambahkan kode dinamis ke `backend.jinja` file yang menentukan backend dan penyedia Terraform untuk digunakan.
	
	Gunakan langkah-langkah berikut:
	
	1. Buat `backend.jinja` file baru di `terraform` direktori.
	2. Buka file, tambahkan kode berikut, dan simpan file.
	
	``
	## Auto generated
	 backend.tf ##
	## Updated on:
	 {{ timestamp }} ##
	
	{% if tf_distri
	bution_type == "oss"
	 -%}
	terraform {
	  required_version
	 = ">= {{ tf_versio
	n }}"
	  backend "s3" {
	  region =
	 "{{ region }}"
	  bucket =
	 "{{ bucket }}"
	  key =
	 "{{ key }}"
	  dynamodb_table
	 = "{{ dynamodb_
	table }}"
	  encrypt =
	 "true"
	``
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`kms_key_id = "{{ kms_key_id }}" role_arn = "{{ aft_admin_role_arn }}" } } {% else -%} terraform { backend "remote" { organization = "{{ terraform_org_name }}" workspaces { name = "{{ terraform_workspace_name }}" } } } {% endif %}`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat `main.tf` file.	Buat `main.tf` file baru dan tambahkan kode yang mendefinisikan dua sumber data yang mengambil dua nilai dari AWS Systems Manager (`aws_ssm`) dan membuat VPC.
	
	Gunakan langkah-langkah berikut:
	
	1. Buat `main.tf` file baru di `terraform` direktori.
	2. Buka file, tambahkan kode berikut, dan simpan file.
	
	`# Define data sources`
	`data "aws_ssm_`
	`parameter" "vpc_ipam`
	`_id" {`
	`  name = "/aft/acc`
	`ount-request/custo`
	`m-fields/vpc-`
	`ipam-id" # Value`
	` is defined in the`
	` account-request.tf`
	` file`
	`}`
	
	`data "aws_ssm_`
	`parameter" "vpc_netm`
	`ask" {`
	`  name = "/aft/acc`
	`ount-request/custo`
	`m-fields/vpc-ipam-`
	`netmask" # Value`
	` is defined in the`
	` account-request.tf`
	` file`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`}`
	
	`# Create new VPC`
	`resource "aws_vpc"`
	`"vpc1" {`
	
	`ipv4_ipam_pool_id`
	`= data.aws_ssm_param`
	`eter.vpc_ipam_id.v`
	`alue # Retrieved from`
	`SSM - this is how we`
	`integrate with IPAM`
	`ipv4_netmask_lengt`
	`h = data.aws_`
	`ssm_parameter.vpc_`
	`netmask.value #`
	`Retrieved from SSM`
	
	`assign_generated_i`
	`pv6_cidr_block`
	`= var.enable_ipv6 ?`
	`true : null`
	`ipv6_cidr_block`
	`=`
	`var.ipv6_cidr`
	`ipv6_ipam_pool_id`
	`=`
	`var.ipv6_ipam_pool`
	`_id`
	`ipv6_netmask_lengt`
	`h =`
	`var.ipv6_netmask_l`
	`ength`
	`ipv6_cidr_block_ne`
	`twork_border_group =`
	`var.ipv6_cidr_bloc`
	`k_network_border_g`
	`roup`
	
	`instance_tenancy`
	`=`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> var.instance_tenancy
	  enable_dns_hostnames = var.enable_dns_hostnames
	  enable_dns_support = var.enable_dns_support
	  enable_network_address_usage_metrics = var.enable_network_address_usage_metrics
	
	  tags = var.tags
	
	  lifecycle {
	  ignore_changes = [
	  tags, # Any changes made to VPC tags after creation will not be overwritten - remove to revert these changes during future 'terraform apply' operations
	  ]
	  }
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat `variables.tf` file.	Buat `variables.tf` file yang mendeklarasikan variabel yang digunakan oleh modul Terraform.
	
	Gunakan langkah-langkah berikut:
	
	1. Buat `variables.tf` file baru di `terraform` direktori.
	2. Buka file, tambahkan kode berikut, dan simpan file.
	
	`# Copied from AWS VPC module`
	`# https://github.com/terraform-aws-modules/terraform-aws-vpc/blob/master/variables.tf`
	
	`variable "name" {`
	`  description = "Name to be used on all the resources as identifier"`
	`  type        = string`
	`  default     = ""`
	`}`
	
	`variable "enable_ipv6" {`
	`  description = "Requests an Amazon-provided IPv6 CIDR block with a /56 prefix length for the VPC. You cannot`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>specify the range of IP addresses, or the size of the CIDR block"
	type = bool
	default = false
	}
	
	variable "ipv6_cidr" {
	description = "(Optional) IPv6 CIDR block to request from an IPAM Pool. Can be set explicitly or derived from IPAM using `ipv6_netmask_length`"
	type = string
	default = null
	}
	
	variable "ipv6_ipam_pool_id" {
	description = "(Optional) IPAM Pool ID for a IPv6 pool. Conflicts with `assign_generated_ipv6_cidr_block`"
	type = string
	default = null
	}
	
	variable "ipv6_netmask_length" {
	description = "(Optional) Netmask length to request from IPAM Pool. Conflicts with `ipv6_cidr_block`.</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>This can be omitted if IPAM pool as a `allocation_default_netmask_length` set. Valid values: `56`"
	  type = number
	  default = null
	}
	
	variable "ipv6_cidr_block_network_border_group" {
	  description = "By default when an IPv6 CIDR is assigned to a VPC a default ipv6_cidr_block_network_border_group will be set to the region of the VPC. This can be changed to restrict advertisement of public addresses to specific Network Border Groups such as LocalZones"
	  type = string
	  default = null
	}
	
	variable "instance_tenancy" {
	  description = "A tenancy option for instances launched into the VPC"
	  type = string
	  default = "default"
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>variable "enable_dns_hostnames" {
	  description = "Should be true to enable DNS hostnames in the VPC"
	  type = bool
	  default = true
	}
	
	variable "enable_dns_support" {
	  description = "Should be true to enable DNS support in the VPC"
	  type = bool
	  default = true
	}
	
	variable "enable_network_address_usage_metrics" {
	  description = "Determines whether network address usage metrics are enabled for the VPC"
	  type = bool
	  default = null
	}
	
	variable "tags" {
	  description = "A map of tags to add to all resources"
	  type = map(string)
	  default = {}
	}</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat `terraform.tfvars` file.	Buat `terraform.tfvars` file yang mendefinisikan nilai-nilai variabel yang diteruskan ke `main.tf` file.
	
	Gunakan langkah-langkah berikut:
	
	1. Buat `terraform.tfvars` file baru di `terraform` direktori.
	2. Buka file, tambahkan kode berikut, dan simpan file.
	
	`name = "demo-ipam"`
	`enable_ipv6 = false`
	`enable_dns_hostnames = true`
	`enable_dns_support = true`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat `outputs.tf` file.	Buat `outputs.tf` file baru yang mengekspos beberapa nilai di CodeBuild. Gunakan langkah-langkah berikut: 1. Buat `outputs.tf` file baru di `terraform` direktori. 2. Buka file, tambahkan kode berikut, dan simpan file. `# Output VPC ID and ARN in CodeBuild output "vpc1" { description = "VPC 1 information" value = { id = try(aws_vpc.vpc1.id, null) arn = try(aws_vpc.vpc1.arn, null) } }`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Komit kustomisasi.	Untuk melakukan penyesuaian baru ke repositori kustomisasi akun, jalankan perintah berikut:
	`# Assumes you are still in the /terraform directory`
	`cd .. # Skip if you are in the account customization root directory (APG-AFT-IPAM)`
	`git add .`
	`git commit -m "APG customization"`
	`git push origin`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan kustomisasi.	Tambahkan kode ke `account-requests.tf` file yang meminta akun baru dengan kustomisasi akun yang baru dibuat. Bidang kustom membuat parameter Systems Manager di akun vended yang diperlukan untuk membuat VPC dengan CIDR yang dialokasikan IPv4 IPAM yang benar. Gunakan langkah-langkah berikut: 1. Dari terminal Anda, navigasikan ke `aft-account-request/terraform` direktori. 2. Buka `account-requests.tf` file, tambahkan dan modifikasi kode berikut dengan nilai Anda, dan simpan file. `module "sandbox_account_01" { source = "./modules/aft-account-request" control_tower_parameters = { AccountEmail = "john.doe@example.com"`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> AccountName =
	 "sandbox-account-0
	1"
	    # Syntax for top-
	level OU
	    ManagedOr
	ganizationalUnit =
	 "Sandbox"
	    # Syntax for
	 nested OU
	    # ManagedOr
	ganizationalUnit =
	 "Sandbox (ou-xfe5-
	a8hb8ml8)"
	    SSOUserEmail
	 = "john.doe@example.
	com"
	    SSOUserFirstName
	 = "John"
	    SSOUserLastName
	 = "Doe"
	  }
	
	  account_tags = {
	    "ABC:Owner"
	 = "john.doe@example.
	com"
	    "ABC:Division"
	 = "ENT"
	    "ABC:Environment"
	 = "Dev"
	    "ABC:CostCenter"
	 = "123456"
	    "ABC:Vended"
	 = "true"
	    "ABC:DivCode"
	 = "102"
	    "ABC:BUCode"
	 = "ABC003"
	    "ABC:Project"
	 = "123456"
	  }</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

```
  change_management_
parameters = {
    change_re
quested_by = "John
 Doe"
    change_reason
       = "testing
 the account vending
 process"
  }

  custom_fields = {
    vpc-ipam-id =
 "ipam-pool-123456a
b789041cd2"
    vpc-ipam-netmask
 = "24"

  }

  account_customizat
ions_name = "APG-AFT-
IPAM" # Do not change
 this value
}
```

> **Important**
>
> Pastikan `vpc-ipam-netmask` nilainya kurang dari atau sama dengan alokasi kolam IPAM`netmask`, dan ada cukup IPs tersedia di kolam IPAM untuk

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dialokasikan ke VPC.
	
	3. Untuk memasukkan akun baru ke repositori permintaan akun, gunakan perintah berikut:
	
	`git add .`
	`git commit -m 'requested APG account'`
	`git push origin`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi kustomisasi.	Masuk ke akun yang baru dijual dan verifikasi bahwa kustomisasi berhasil diterapkan. Gunakan langkah-langkah berikut: 1. Buka halaman AWS IAM Identity Center login Anda dan autentikasi sebagai pengguna yang Anda tentukan dalam file. `account-requests.tf` 2. Dari daftar akun Pusat Identitas IAM, pilih akun yang Anda jual sebelumnya dan masuk menggunakan peran tersebut AWS `AdministratorAccess` . 3. Arahkan ke konsol VPC, dan pilih Your. VPCs 4. Pilih ID VPC dari VPC yang baru saja Anda buat. 5. Pilih CIDRstab. Anda akan melihat CIDR yang dialokasikan IPv4 IPAM. 6. (Opsional) Masuk ke akun administrator yang didelegasikan IPAM Anda dan verifikasi bahwa alokasi ditampilkan di konsol IPAM.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Anda mengalami kegagalan dalam pembuatan atau manajemen sumber daya yang disebabkan oleh izin yang tidak memadai.	Tinjau peran dan kebijakan AWS Identity and Access Management (IAM) yang dilampirkan ke Step Functions CodeBuild, dan layanan lain yang terlibat dalam penerapan. Konfirmasikan bahwa mereka memiliki izin yang diperlukan. Jika ada masalah izin, sesuaikan kebijakan IAM untuk memberikan akses yang diperlukan.
Anda mencapai Layanan AWS kuota selama penerapan.	Sebelum Anda menerapkan pipeline, periksa Layanan AWS kuota untuk sumber daya seperti bucket Amazon Simple Storage Service (Amazon S3), peran IAM, dan fungsi. AWS Lambda Jika perlu, permintaan meningkat ke kuota. Untuk informasi lebih lanjut, lihat Layanan AWS kuota di Referensi AWS Umum.

Sumber daya terkait

Layanan AWS dokumentasi

- AWS Control Tower Panduan Pengguna
- Bagaimana IPAM bekerja
- Praktik terbaik keamanan di IAM
- Layanan AWS kuota

Sumber daya lainnya

- Dokumentasi Penyedia Terraform AWS

Otomatis menambahkan atau memperbarui entri registri Windows menggunakan AWS Systems Manager

Appasaheb Bagali, Amazon Web Services

Ringkasan

AWS Systems Manager adalah alat manajemen jarak jauh untuk instans Amazon Elastic Compute Cloud (Amazon EC2). Systems Manager menyediakan visibilitas dan kontrol atas infrastruktur Anda di Amazon Web Services. Alat serbaguna ini dapat digunakan untuk memperbaiki perubahan registri Windows yang diidentifikasi sebagai kerentanan oleh laporan pemindaian kerentanan keamanan.

Pola ini mencakup langkah-langkah untuk menjaga EC2 instance Anda yang menjalankan sistem operasi Windows aman dengan mengotomatiskan perubahan registri yang direkomendasikan untuk keamanan lingkungan Anda. Pola menggunakan perintah Run untuk menjalankan dokumen Command. Kode dilampirkan, dan sebagian dari itu termasuk dalam bagian Kode.

Prasyarat dan batasan

- Akun AWS yang aktif
- Izin untuk mengakses EC2 instans dan Systems Manager

Arsitektur

Tumpukan teknologi target

- Virtual Private Cloud (VPC), dengan dua subnet dan gateway Network Address Translation (NAT)
- Dokumen Systems Manager Command untuk menambah atau memperbarui nama dan nilai registri
- Systems Manager Run Command untuk menjalankan dokumen Command pada EC2 instance yang ditentukan

Arsitektur target

Alat

Alat

- Kebijakan dan peran IAM — AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke sumber daya AWS dengan aman. Anda menggunakan IAM untuk mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan sumber daya.
- Amazon Simple Storage Service — Amazon Simple Storage Service (Amazon S3) adalah penyimpanan untuk internet. Ini dirancang untuk membuat komputasi skala web lebih mudah bagi pengembang. Dalam pola ini, bucket S3 digunakan untuk menyimpan log Systems Manager.
- AWS Systems Manager — AWS Systems Manager adalah layanan AWS yang dapat Anda gunakan untuk melihat dan mengontrol infrastruktur di AWS. Systems Manager membantu Anda menjaga keamanan dan kepatuhan dengan memindai instans terkelola dan melaporkan (atau mengambil tindakan korektif terhadap) setiap pelanggaran kebijakan yang dideteksi.
- Dokumen AWS Systems Manager Command — Dokumen AWS Systems Manager Command digunakan oleh Run Command. Sebagian besar dokumen Command didukung pada semua sistem operasi Linux dan Windows Server yang didukung oleh Systems Manager.
- AWS Systems Manager Run Command — AWS Systems Manager Run Command memberi Anda cara untuk mengelola konfigurasi instans terkelola Anda dari jarak jauh dan aman. Menggunakan Run Command, Anda dapat mengotomatiskan tugas administratif umum dan melakukan perubahan konfigurasi satu kali dalam skala besar.

Kode

Anda dapat menggunakan kode contoh berikut untuk menambahkan atau memperbarui nama registri Microsoft Windows ke`Version`, jalur registri ke`HKCU:\Software\ScriptingGuys\Scripts`, dan nilai ke2.

```
#Windows registry path which needs to add/update
$registryPath ='HKCU:\\Software\\ScriptingGuys\\Scripts'
#Windows registry Name  which needs to add/update
$Name = 'Version'
#Windows registry value  which needs to add/update
$value = 2
# Test-Path cmdlet to see if the registry key exists.
IF(!(Test-Path $registryPath))
        {
           New-Item -Path $registryPath -Force | Out-Null
           New-ItemProperty -Path $registryPath -Name $name -Value     $value ` -
PropertyType DWORD -                 Force | Out-        Null
        } ELSE {
```

```
                    New-ItemProperty -Path $registryPath -Name $name -Value $value `
 -PropertyType            DWORD        -Force | Out-Null
           }
echo 'Registry Path:'$registryPath
 echo 'Registry Name:'$registryPath
 echo 'Registry Value:'(Get-ItemProperty -Path $registryPath -Name $Name).version
```

Contoh kode Systems Manager Command document JavaScript Object Notation (JSON) lengkap dilampirkan.

Epik

Menyiapkan VPC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sebuah VPC.	Di AWS Management Console, buat VPC yang memiliki subnet publik dan pribadi serta gateway NAT. Untuk informasi selengkapnya, lihat dokumentasi AWS.	Administrator awan
Buat grup keamanan.	Pastikan bahwa setiap grup keamanan memungkinkan akses untuk Remote Desktop Protocol (RDP) dari alamat IP sumber.	Administrator awan

Buat kebijakan IAM dan peran IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM.	Buat kebijakan IAM yang menyediakan akses ke Amazon S3, EC2 Amazon, dan Systems Manager.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM.	Buat peran IAM, dan lampirkan kebijakan IAM yang menyediakan akses ke Amazon S3, EC2 Amazon, dan Systems Manager.	Administrator awan

Jalankan otomatisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dokumen Systems Manager Command.	Buat dokumen Command Systems Manager yang akan menyebarkan perubahan registri Microsoft Windows untuk menambah atau memperbarui.	Administrator awan
Jalankan perintah Systems Manager Run.	Jalankan Systems Manager Run Command, pilih dokumen Command dan instance target Systems Manager. Ini mendorong perubahan registri Microsoft Windows dalam dokumen Command yang dipilih ke instance target.	Administrator awan

Sumber daya terkait

- AWS Systems Manager
- Dokumen AWS Systems Manager
- AWS Systems Manager Jalankan Perintah

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Secara otomatis membuat RFC di AMS menggunakan Python

Gnanasekaran Kailasam, Amazon Web Services

Ringkasan

AWS Managed Services (AMS) membantu Anda mengoperasikan infrastruktur berbasis cloud dengan lebih efisien dan aman dengan menyediakan pengelolaan infrastruktur Amazon Web Services (AWS) yang berkelanjutan. Untuk membuat perubahan pada lingkungan terkelola, Anda perlu membuat dan mengirimkan permintaan perubahan (RFC) baru yang menyertakan ID tipe perubahan (CT) untuk operasi atau tindakan tertentu.

Namun, membuat RFC secara manual dapat memakan waktu sekitar lima menit dan tim di organisasi Anda mungkin perlu mengirimkan beberapa RFCs setiap hari. Pola ini membantu Anda mengotomatiskan proses pembuatan RFC, mengurangi waktu pembuatan untuk setiap RFC, dan menghilangkan kesalahan manual.

Pola ini menjelaskan cara menggunakan kode Python untuk secara otomatis membuat `Stop EC2 instance` RFC yang menghentikan instans Amazon Elastic Compute Cloud (Amazon EC2) di akun AMS Anda. Anda kemudian dapat menerapkan pendekatan pola ini dan otomatisasi Python ke jenis RFC lainnya.

Prasyarat dan batasan

Prasyarat

- Akun AMS Advanced. Untuk informasi selengkapnya tentang hal ini, lihat paket operasi AMS di dokumentasi AWS Managed Services.
- Setidaknya satu EC2 instance yang ada di akun AMS Anda.
- Pemahaman tentang cara membuat dan mengirimkan RFCs di AMS.
- Keakraban dengan Python

Batasan

- Anda hanya dapat menggunakan RFCs untuk perubahan di akun AMS Anda. Akun AWS Anda menggunakan proses yang berbeda untuk perubahan serupa.

Arsitektur

Tumpukan teknologi

- AMS
- Antarmuka Baris Perintah AWS (AWS CLI)
- AWS SDK untuk Python (Boto3)
- Python dan paket yang diperlukan (JSON dan Boto3)

Otomatisasi dan skala

Pola ini menyediakan kode sampel untuk mengotomatiskan `Stop EC2 instance` RFC, tetapi Anda dapat menggunakan kode sampel pola ini dan pendekatan untuk yang lain. RFCs

Alat

- AWS Managed Services — AMS membantu Anda mengoperasikan infrastruktur AWS dengan lebih efisien dan aman.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat terpadu untuk mengelola layanan AWS Anda. Di AMS, API manajemen perubahan menyediakan operasi untuk membuat dan mengelola RFCs.
- AWS SDK for Python (Boto3) — SDK untuk Python memudahkan Anda mengintegrasikan aplikasi, pustaka, atau skrip Python Anda dengan layanan AWS.

Kode

`AMS Stop EC2 Instance.zip`File (terlampir) berisi kode Python untuk membuat RFC. `Stop EC2 instance` Anda juga dapat mengonfigurasi kode ini untuk mengirimkan satu RFC untuk beberapa EC2 instance.

Epik

Opsi 1 - Mengatur lingkungan untuk macOS atau Linux

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan validasi Python.	1. Buka jendela terminal dan jalankan `brew install python3` perintah. 2. Validasi bahwa Python diinstal dengan benar dengan menjalankan `python --version` perintah. 3. Validasi `pip` yang diinstal dengan benar dengan menjalankan `pip --version` perintah.	Administrator sistem AWS
Instal AWS CLI.	Jalankan *`pip install awscli --upgrade –user`* perintah untuk menginstal AWS CLI.	Administrator sistem AWS
Instal Boto3.	Jalankan `pip install boto3` perintah untuk menginstal Boto3.	Administrator sistem AWS
Instal JSON.	Jalankan `pip install json` perintah untuk menginstal JSON.	Administrator sistem AWS
Mengatur AMS CLI.	Masuk ke AWS Management Console, buka konsol AMS, lalu pilih Dokumentasi. Unduh file.zip yang berisi AMS CLI, unzip, dan kemudian instal di mesin lokal Anda.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Setelah Anda menginstal AMS CLI, jalankan perintah. `aws amscm help` Output memberikan informasi tentang proses manajemen perubahan AMS.	

Opsi 2 - Mengatur lingkungan untuk Windows

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan validasi Python.	1. Buka rilis Python untuk halaman Windows, unduh versi terbaru, dan kemudian instal Python. 2. Validasi bahwa Python diinstal dengan benar dengan menjalankan `python --version` perintah. 3. Validasi `pip` yang diinstal dengan benar dengan menjalankan `pip --version` perintah.	Administrator sistem AWS
Instal AWS CLI.	Jalankan `pip install awscli --upgrade –user` perintah untuk menginstal AWS CLI.	Administrator sistem AWS
Instal Boto3.	Jalankan `pip install boto3` perintah untuk menginstal Boto3.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal JSON.	Jalankan `pip install json` perintah untuk menginstal JSON.	Administrator sistem AWS
Mengatur AMS CLI.	Masuk ke AWS Managemen t Console, buka konsol AMS, lalu pilih Dokumentasi. Unduh file.zip yang berisi AMS CLI, unzip, dan kemudian instal di mesin lokal Anda. Setelah Anda menginstal AMS CLI, jalankan perintah. `aws amscm help` Output memberikan informasi tentang proses manajemen perubahan AMS	Administrator sistem AWS

Ekstrak ID CT dan parameter eksekusi untuk RFC

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak CT ID, versi, dan parameter eksekusi untuk RFC.	Setiap RFC memiliki CT ID, versi, dan parameter eksekusi yang berbeda. Anda dapat mengekstrak informasi ini dengan menggunakan salah satu opsi berikut: 1. Ikuti petunjuk dari Menemukan permintaan untuk perubahan (RFC) dengan bagian CLI dalam contoh penggunaan RFC	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dari dokumentasi AWS Managed Services. 2. Buka RFC yang ada dengan tipe serupa atau buat RFC baru sebagai pengujian melalui konsol AMS. Gunakan CT ID RFC dan parameter eksekusi. Untuk informasi selengkap nya tentang hal ini, lihat Menemukan RFC dengan konsol di dokumentasi AWS Managed Services. **Note** Untuk mengadaptasi otomatisasi Python pola ini untuk yang lain RFCs, ganti jenis CT dan nilai parameter dalam file kode `ams_stop_ec2_instance` Python dari `AMS Stop EC2 Instance.zip` file (terlampir) dengan yang Anda ekstrak.	

Jalankan otomatisasi Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan otomatisasi Python.	1. Unduh `AMS Stop EC2 Instance.zip` file (terlampir) ke mesin lokal Anda dan ekstrak file. 2. Perbarui `input_instances` dengan informasi EC2 instans Anda. 3. Buka terminal dan arahkan ke jalur untuk kode yang Anda ekstrak 4. Jalankan perintah `pythonams_stop_ec2_instance.py` .	Administrator sistem AWS

Sumber daya terkait

- Apa itu jenis perubahan?
- Tutorial CLI: Tumpukan dua tingkat ketersediaan tinggi (Linux/RHEL)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Secara otomatis berhenti dan memulai instans Amazon RDS DB menggunakan Windows AWS Systems Manager Pemeliharaan

Ashita Dsilva, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menghentikan dan memulai instans DB Amazon Relational Database Service (Amazon RDS) secara otomatis pada jadwal tertentu (misalnya, mematikan instans DB di luar jam kerja untuk mengurangi biaya) dengan menggunakan Windows Pemeliharaan. AWS Systems Manager Untuk tujuan ini, Systems Manager hemat biaya untuk kasus penggunaan biasa.

AWS Systems Manager Automation menyediakan `AWS-StopRdsInstance` dan `AWS-StartRdsInstance` runbook untuk menghentikan dan memulai instans Amazon RDS DB. Ini berarti Anda tidak perlu menulis logika khusus dengan AWS Lambda fungsi atau membuat aturan Amazon CloudWatch Events.

Systems Manager menyediakan dua kemampuan untuk tugas penjadwalan: State Manager dan Maintenance Windows. State Manager menetapkan dan mempertahankan konfigurasi status yang diperlukan untuk sumber daya di akun Amazon Web Services (AWS) Anda satu kali atau pada jadwal tertentu. Pemeliharaan Windows menjalankan tugas pada sumber daya di akun Anda selama jendela waktu tertentu. Meskipun Anda dapat menggunakan pendekatan pola ini dengan State Manager atau Maintenance Windows, kami menyarankan Anda menggunakan Maintenance Windows karena dapat menjalankan satu atau lebih tugas berdasarkan prioritas yang ditetapkan dan juga dapat menjalankan AWS Lambda fungsi dan AWS Step Functions tugas. Untuk informasi selengkapnya tentang State Manager dan Maintenance Windows, lihat Memilih antara State Manager dan Maintenance Windows dalam dokumentasi Systems Manager.

Pola ini memberikan langkah-langkah terperinci untuk mengonfigurasi dua jendela pemeliharaan terpisah yang menggunakan ekspresi cron untuk berhenti dan kemudian memulai instans Amazon RDS DB.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Instans Amazon RDS DB yang ada yang ingin Anda hentikan dan mulai pada jadwal tertentu.

- Ekspresi cron untuk jadwal yang Anda butuhkan. Misalnya, ekspresi `cron(0 9 ? * MON-FRI *)` menjalankan tugas pada pukul 09:00 setiap hari Senin, Selasa, Rabu, Kamis, dan Jumat. Untuk informasi selengkapnya, lihat Cron dan ekspresi tingkat untuk jendela pemeliharaan di dokumentasi Systems Manager.
- Keakraban dengan Systems Manager
- Izin untuk memulai dan menghentikan instance RDS. Untuk informasi lebih lanjut, lihat bagian Epik.

Batasan

- Instans Amazon RDS DB dapat dihentikan hingga tujuh hari sekaligus. Setelah tujuh hari, instans DB secara otomatis restart untuk memastikan bahwa ia menerima pembaruan pemeliharaan yang diperlukan.
- Anda tidak dapat menghentikan instance DB yang merupakan replika baca atau yang memiliki replika baca.
- Anda tidak dapat menghentikan instans Amazon RDS for SQL Server DB dalam konfigurasi Multi-AZ.
- Kuota layanan berlaku untuk Maintenance Windows dan Systems Manager Automation. Untuk informasi selengkapnya tentang kuota layanan, lihat AWS Systems Manager titik akhir dan kuota dalam dokumentasi. Referensi Umum AWS
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan alur kerja untuk secara otomatis berhenti dan memulai instans Amazon RDS DB.

Alur kerja memiliki langkah-langkah berikut:

1. Buat jendela pemeliharaan dan gunakan ekspresi cron untuk menentukan jadwal berhenti dan mulai untuk instans Amazon RDS DB Anda.

2. Daftarkan tugas Otomasi Systems Manager ke jendela pemeliharaan dengan menggunakan `AWS-StopRdsInstance` atau `AWS-StartRdsInstance` runbook.

3. Daftarkan target dengan jendela pemeliharaan menggunakan grup sumber daya berbasis tag untuk instans Amazon RDS DB Anda.

Tumpukan teknologi

- AWS CloudFormation
- AWS Identity and Access Management (IAM)
- Amazon RDS
- Systems Manager

Otomatisasi dan skala

Anda dapat menghentikan dan memulai beberapa instans Amazon RDS DB secara bersamaan dengan menandai instans Amazon RDS DB yang diperlukan, membuat grup sumber daya yang menyertakan semua instans DB yang ditandai, dan mendaftarkan grup sumber daya ini sebagai target untuk jendela pemeliharaan.

Alat

- AWS CloudFormationadalah layanan yang membantu Anda memodelkan dan mengatur AWS sumber daya Anda.
- AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke AWS sumber daya dengan aman.
- Amazon Relational Database Service (Amazon RDS) adalah layanan web yang membuatnya lebih mudah untuk mengatur, mengoperasikan, dan menskalakan database relasional di. AWS Cloud
- Grup Sumber Daya AWSmembantu Anda mengatur AWS sumber daya ke dalam grup, menandai sumber daya, dan mengelola, memantau, dan mengotomatiskan tugas pada sumber daya yang dikelompokkan.
- AWS Systems Manageradalah Layanan AWS yang dapat Anda gunakan untuk melihat dan mengontrol infrastruktur Anda AWS. Pola ini menggunakan fitur-fitur berikut dari Systems Manager:
  - AWS Systems Manager Otomatisasi menyederhanakan tugas pemeliharaan dan penerapan umum instans Amazon Elastic Compute Cloud (Amazon EC2) dan sumber daya lainnya. AWS
  - AWS Systems Manager Pemeliharaan Windows membantu Anda menentukan jadwal kapan harus melakukan tindakan yang berpotensi mengganggu pada instans Anda.

Epik

Membuat dan mengonfigurasi peran layanan IAM untuk Automasi Systems Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan peran layanan IAM untuk Automasi Systems Manager.	Masuk ke Konsol Manajemen AWS dan buat peran layanan untuk Otomasi Systems Manager. Anda dapat menggunakan salah satu dari dua metode berikut untuk membuat peran layanan ini: • Gunakan AWS CloudFormation untuk mengonfigurasi peran layanan untuk Otomasi Systems Manager • Gunakan IAM untuk mengonfigurasi peran untuk Automasi Systems Manager Alur kerja Otomasi Systems Manager memanggil Amazon RDS dengan menggunakan peran layanan untuk melakukan tindakan mulai dan menghentikan pada instans Amazon RDS DB. Peran layanan harus dikonfigurasi dengan kebijakan sebaris berikut yang memiliki izin untuk memulai dan menghentikan instans Amazon RDS DB: `{`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Sid": "RdsStartStop",
	"Effect": "Allow",
	"Action": [
	"rds:StopDBInstance",
	"rds:StartDBInstance"
	],
	"Resource": "<RDS_Instance_ARN>"
	},
	{
	"Sid": "RdsDescribe",
	"Effect": "Allow",
	"Action": "rds:DescribeDBInstances",
	"Resource": "*"
	}
	]
	}</pre>
	Pastikan Anda mengganti <RDS_Instance_ARN> dengan Amazon Resource Name (ARN) instans Amazon RDS DB Anda.
	
	Jika Anda tidak terbiasa menggunakan kebijakan dan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	peran IAM, ikuti petunjuk di bagian Ikhtisar Solusi pada penghentian Jadwalkan Amazon RDS dan mulai gunakan AWS Systems Manager posting blog. **Important** Pastikan Anda merekam ARN dari peran layanan.	

Buat grup sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tandai instans Amazon RDS DB.	Buka konsol Amazon RDS dan beri tag instans Amazon RDS DB yang ingin Anda tambahkan ke grup sumber daya. Tag adalah metadata yang ditetapkan ke AWS sumber daya dan terdiri dari pasangan kunci-nilai. Kami menyarankan Anda menggunakan Action sebagai kunci Tag dan StartStop sebagai Nilai. Untuk informasi selengkapnya tentang ini, lihat Menambahkan, mencantumkan, dan	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menghapus tag di dokumentasi Amazon RDS.	
Buat grup sumber daya untuk instans Amazon RDS DB yang ditandai.	Buka Grup Sumber Daya AWS konsol dan buat grup sumber daya berdasarkan tag yang Anda buat untuk instans Amazon RDS DB Anda. Di bawah Kriteria Pengelompokan, pastikan Anda memilih AWS: :RDS:: DBInstance untuk jenis sumber daya dan kemudian berikan pasangan nilai kunci tag (misalnya, “Tindakan- “). StartStop Ini memastikan bahwa layanan hanya memeriksa instans Amazon RDS DB dan bukan sumber daya lain yang memiliki tag ini. Pastikan Anda merekam nama grup sumber daya. Untuk informasi selengkapnya dan langkah-langkah mendetail, lihat Membuat kueri berbasis tag dan membuat grup dalam Grup Sumber Daya AWS dokumentasi.	Administrator AWS

Konfigurasikan jendela pemeliharaan untuk menghentikan instans Amazon RDS DB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat jendela pemeliharaan.	1. Buka konsol Systems Manager, pilih Maintenance Windows, lalu pilih Create a maintenance window. Berikan nama untuk jendela pemeliharaan Anda (misalnya, "StopRdsInstance“), masukkan deskripsi, lalu hapus centang Izinkan target yang tidak terdaftar. 2. Pilih ekspresi cron/rate dan berikan ekspresi jadwal untuk menentukan kapan instans Amazon RDS DB harus dihentikan. Masukkan 1 untuk Durasi dan 0 untuk Berhenti memulai tugas. Secara default, zona waktu menunjukkan UTC. Anda dapat mengubah zona waktu untuk memulai jendela pemeliharaan berdasarkan stempel waktu yang ditentukan dalam ekspresi cron Anda. 3. Pilih Buat jendela pemeliharaan. Sistem mengembalikan Anda ke halaman jendela pemeliharaan dan status jendela pemeliharaan Anda Diaktifkan.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important Tugas untuk menghentikan instans DB berjalan hampir secara instan saat dimulai dan tidak mencakup seluruh durasi jendela pemeliharaan. Pola ini memberikan nilai minimum untuk Durasi dan Berhenti memulai tugas karena mereka adalah parameter yang diperlukan untuk jendela pemeliharaan. Untuk informasi selengkap nya dan langkah-langkah terperinci, lihat Membuat jendela pemeliharaan (konsol) di dokumentasi Systems Manager.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan target ke jendela pemeliharaan.	1. Pada konsol Systems Manager, pilih Maintenance Windows, pilih Actions, lalu pilih Daftarkan target. 2. Di area Target, tentukan Pilih grup sumber daya lalu pilih nama grup sumber daya yang ada di akun Anda. 3. Untuk tipe Resource, pilih AWS: :RDS:: DBInstance lalu pilih Daftarkan target. Untuk informasi selengkap nya dan langkah-langkah terperinci, lihat Menetapkan target ke jendela pemelihar aan (konsol) di dokumentasi Systems Manager.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan tugas ke jendela pemeliharaan.	1. Di konsol Systems Manager, pilih Maintenance Windows, lalu pilih jendela pemeliharaan Anda. Pilih Tindakan, lalu pilih Daftarkan tugas Otomasi. 2. Untuk Dokumen, pilih AWS-StopRdsInstance. 3. Di bagian Target, pilih Memilih grup target terdaftar, lalu pilih target jendela pemeliharaan yang Anda daftarkan dengan jendela pemeliharaan saat ini. 4. Untuk kontrol Rate, tentukan 100 persen untuk Concurrency and Error threshold. Anda dapat mengubah nilai kontrol Rate sesuai dengan kebutuhan Anda untuk konkurensi tugas dan ambang kesalahan. Untuk informasi selengkapnya tentang ini, lihat Tentang ambang konkurensi dan kesalahan dalam dokumentasi Systems Manager. 5. Di bagian peran layanan IAM, untuk peran Layanan, biarkan kotak ini kosong atau buat peran kustom Anda sendiri. Jika Anda	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membiarkan kotak kosong, Systems Manager secara otomatis membuat AWSServiceRoleForA mazonSSM peran terkait layanan dan kemudian mengaitkan peran dengan tugas. Untuk membuat peran kustom Anda sendiri, lihat Membuat peran layanan kustom untuk jendela pemeliharaan (konsol), lalu mengaitkan peran kustom itu dengan tugas. 6. Di bagian Parameter Input, tentukan parameter berikut untuk runbook: • InstanceId: `{{RESOURCE_ID}}` **Note** Untuk InstanceId, parameter semu digunakan untuk mengekstrak ID sumber daya Amazon RDS DB dari ARN. Untuk mempelajari lebih lanjut tentang parameter semu, lihat Tentang parameter semu	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di dokumenta si Systems Manager. • AutomationAssumeRo le: Berikan ARN peran layanan yang Anda buat untuk Systems Manager Automation. 7. Pilih tugas Register Automation. **Important** Opsi peran Layanan mendefinisikan peran layanan yang diperlukan untuk jendela pemeliharaan untuk menjalankan tugas. Namun, peran ini tidak identik dengan peran layanan yang Anda buat sebelumny a untuk Otomasi Systems Manager. Untuk informasi selengkap nya dan langkah-langkah terperinci, lihat Menetapkan tugas ke jendela pemelihar aan (konsol) di dokumentasi Systems Manager.	

Konfigurasikan jendela pemeliharaan untuk memulai instans Amazon RDS DB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan jendela pemeliharaan untuk memulai instans Amazon RDS DB.	Ulangi langkah-langkah dari jendela Konfigurasi pemeliharaan untuk menghentikan epik instans Amazon RDS DB untuk mengonfigurasi jendela pemeliharaan lain untuk memulai instans Amazon RDS DB pada waktu yang dijadwalkan. **Important** Anda harus membuat perubahan berikut ketika Anda mengkonfigurasi jendela pemeliharaan untuk memulai instans DB: • Gunakan nama baru untuk jendela pemeliharaan (misalnya, "StartRds Instance“). • Ganti ekspresi cron dengan ekspresi cron yang ingin Anda gunakan untuk memulai instance DB. • Ganti `AWS-StopRdsInstance` runbook dengan `AWS-StartRdsInstance` in Task.	Administrator AWS

Sumber daya terkait

- Gunakan dokumen Systems Manager Automation untuk mengelola instans dan memotong biaya di luar jam kerja (AWS posting blog)

Memusatkan distribusi paket perangkat lunak di AWS Organizations dengan menggunakan Terraform

Pradip Kumar Pandey, Chintamani Aphale, T.V.R.L.Phani Kumar Dadi, Pratap Kumar Nanda, Aarti Rajput, dan Mayuri Shinde, Amazon Web Services

Ringkasan

Perusahaan sering mempertahankan beberapa Akun AWS yang tersebar Wilayah AWS di beberapa untuk menciptakan penghalang isolasi yang kuat antara beban kerja. Agar tetap aman dan patuh, tim administrasi mereka memasang alat berbasis agen seperti CrowdStrike,, atau TrendMicroalat untuk pemindaian keamanan SentinelOne, dan CloudWatch agen Amazon, AgenDatadog, atau agen untuk pemantauan. AppDynamics Tim-tim ini sering menghadapi tantangan ketika mereka ingin mengotomatiskan manajemen dan distribusi paket perangkat lunak secara terpusat di seluruh lanskap besar ini.

Distributor, kemampuan AWS Systems Manager, mengotomatiskan proses pengemasan dan penerbitan perangkat lunak ke instans Microsoft Windows dan Linux yang dikelola di seluruh cloud dan server lokal melalui satu antarmuka yang disederhanakan. Pola ini menunjukkan bagaimana Anda dapat menggunakan Terraform untuk lebih menyederhanakan proses pengelolaan instalasi perangkat lunak dan menjalankan skrip di sejumlah besar instance dan akun anggota di dalamnya dengan sedikit usaha. AWS Organizations

Solusi ini berfungsi untuk instans Amazon, Linux, dan Windows yang dikelola oleh Systems Manager.

Prasyarat dan batasan

- Paket Distributor yang memiliki perangkat lunak yang akan diinstal
- Terraform versi 0.15.0 atau yang lebih baru
- Instans Amazon Elastic Compute Cloud (Amazon EC2) yang dikelola oleh Systems Manager dan memiliki izin dasar untuk mengakses Amazon Simple Storage Service (Amazon S3) di akun target
- Sebuah landing zone untuk organisasi Anda yang disiapkan dengan menggunakan AWS Control Tower
- (Opsional) Account Factory untuk Terraform (AFT)

Arsitektur

Detail sumber daya

Pola ini menggunakan Account Factory for Terraform (AFT) untuk membuat semua AWS sumber daya yang diperlukan dan pipeline kode untuk menyebarkan sumber daya di akun penerapan. Pipeline kode berjalan di dua repositori:

- Kustomisasi global berisi kode Terraform yang akan berjalan di semua akun yang terdaftar di AFT.
- Kustomisasi akun berisi kode Terraform yang akan berjalan di akun penerapan.

Anda juga dapat menerapkan solusi ini tanpa menggunakan AFT, dengan menjalankan perintah Terraform di folder penyesuaian akun.

Kode Terraform menerapkan sumber daya berikut:

- AWS Identity and Access Management (IAM) peran dan kebijakan
  - SystemsManager- AutomationExecutionRole memberikan izin pengguna untuk menjalankan otomatisasi di akun target.
  - SystemsManager- AutomationAdministrationRole memberikan izin pengguna untuk menjalankan otomatisasi di beberapa akun dan unit organisasi (). OUs
- File terkompresi dan manifest.json untuk paket
  - Di Systems Manager, sebuah paket mencakup setidaknya satu file.zip perangkat lunak atau aset yang dapat diinstal.
  - Manifes JSON menyertakan pointer ke file kode paket Anda.
- Bucket S3
  - Paket terdistribusi yang dibagikan di seluruh organisasi disimpan dengan aman di bucket Amazon S3.
- AWS Systems Manager dokumen (dokumen SSM)
  - `DistributeSoftwarePackage`berisi logika untuk mendistribusikan paket perangkat lunak ke setiap instance target di akun anggota.
  - `AddSoftwarePackageToDistributor`berisi logika untuk mengemas aset perangkat lunak yang dapat diinstal dan menambahkannya ke Otomasi, kemampuan. AWS Systems Manager
- Asosiasi Systems Manager
  - Asosiasi Systems Manager digunakan untuk menyebarkan solusi.

Arsitektur dan alur kerja

Diagram ini menggambarkan langkah-langkah berikut:

1. Untuk menjalankan solusi dari akun terpusat, Anda mengunggah paket atau perangkat lunak Anda bersama dengan langkah-langkah penerapan ke bucket S3.
2. Paket kustom Anda akan tersedia di bagian Dokumen konsol Systems Manager, di tab Dimiliki oleh saya.
3. State Manager, kemampuan Systems Manager, membuat, menjadwalkan, dan menjalankan asosiasi untuk paket di seluruh organisasi. Asosiasi menentukan bahwa paket perangkat lunak harus diinstal dan berjalan pada node terkelola sebelum dapat diinstal pada node target.
4. Asosiasi menginstruksikan Systems Manager untuk menginstal paket pada node target.
5. Untuk penginstalan atau perubahan berikutnya, pengguna dapat menjalankan asosiasi yang sama secara berkala atau manual dari satu lokasi untuk melakukan penerapan di seluruh akun.
6. Di akun anggota, Automation mengirimkan perintah deployment ke Distributor.
7. Distributor mendistribusikan paket perangkat lunak di seluruh instance.

Solusi ini menggunakan akun manajemen di dalamnya AWS Organizations, tetapi Anda juga dapat menunjuk akun (administrator yang didelegasikan) untuk mengelola ini atas nama organisasi.

Alat

Layanan AWS

- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data. Pola ini menggunakan Amazon S3 untuk memusatkan dan menyimpan paket terdistribusi dengan aman.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar. Pola ini menggunakan kemampuan Systems Manager berikut:
  - Distributor membantu Anda mengemas dan mempublikasikan perangkat lunak ke instans yang dikelola Systems Manager.
  - Otomasi menyederhanakan tugas pemeliharaan, penyebaran, dan remediasi umum untuk banyak layanan. AWS
  - Documents melakukan tindakan pada instans terkelola Systems Manager di seluruh organisasi dan akun Anda.

- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa AWS akun ke dalam organisasi yang Anda buat dan kelola secara terpusat.

Alat-alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Instruksi dan kode untuk pola ini tersedia di repositori distribusi paket GitHub Terpusat.

Praktik terbaik

- Untuk menetapkan tag ke asosiasi, gunakan AWS Command Line Interface (AWS CLI) atau. AWS Tools for PowerShell Menambahkan tag ke asosiasi menggunakan konsol Systems Manager tidak didukung. Untuk informasi selengkapnya, lihat Menandai sumber daya Systems Manager di dokumentasi Systems Manager.
- Untuk menjalankan asosiasi dengan menggunakan versi baru dokumen yang dibagikan dari akun lain, setel versi dokumen ke`default`.
- Untuk menandai hanya node target, gunakan satu kunci tag. Jika Anda ingin menargetkan node Anda dengan menggunakan beberapa kunci tag, gunakan opsi grup sumber daya.

Epik

Konfigurasikan file sumber dan akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	1. Kloning repositori distribusi paket GitHub Terpusat: `git clone https://github.com/aws-samples/aws-organization-centralised-package-distribution`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	2. Repositori kode Terraform memerlukan dua folder kustomisasi yang dikelola oleh AFT. Konfirmasikan bahwa salinan lokal repositori Anda berisi folder ini:
	`$ cd centralised-package-distribution`
	`$ ls`
	`global-customization account-customization`	

Perbarui variabel global.	Perbarui parameter input berikut dalam `global-customization/variables.tf` file. Variabel ini berlaku untuk semua akun yang dibuat dan dikelola oleh AFT. • `account_id` : ID akun tempat solusi Distributor akan digunakan. • `aws_region` : Di Wilayah AWS mana asosiasi akan dikerahkan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui variabel akun.	Perbarui parameter input berikut dalam `account-customization/variables.tf` file. Variabel ini hanya berlaku untuk akun tertentu yang dibuat dan dikelola oleh AFT. • `package_bucket_name` : Nama bucket S3 yang berisi file distribusi paket. • `package_name` : Nama file distribusi paket. • `package_version` : Versi paket installer.	DevOps insinyur

Kustomisasi parameter dan file penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui parameter input untuk asosiasi Manajer Negara.	Perbarui parameter input berikut dalam `account-customization/association.tf` file untuk menentukan status yang ingin Anda pertahankan pada instance Anda. Anda dapat menggunakan nilai parameter default jika mereka mendukung kasus penggunaan Anda. • `targetAccounts` : Unit organisasi (OU) IDs	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	dalam AWS Organizat ions yang mewakili akun dengan instans target untuk distribusi. OU IDs mulai dengan “ou”.
	• `targetRegions` : Wilayah AWS (misalnya, “us-east-1″ atau “ap-southeast-2") tempat instance target berjalan.
	• `action`: Tentukan apakah akan menginstal atau menghapus paket.
	• `installationType` : Salah satu jenis instalasi berikut:
	  • `uninstall` : Paket dihapus.
	  • `reinstall` : Aplikasi diambil secara offline hingga proses penginsta lan ulang selesai.
	  • `In-place update`: Aplikasi ini tersedia sementara file baru atau yang diperbarui ditambahkan ke instalasi.
	• `name`: Nama paket untuk menginstal atau menghapus instalasi.
	• `version`: Versi paket untuk menginstal atau menghapus instalasi. Jika tidak ada versi	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	paket yang diinstal, sistem mengembalikan kesalahan. • `bucketName` : Nama bucket S3 paket telah digunakan. Bucket ini harus terdiri dari paket dan file manifes saja. • `bucketPrefix` : Awalan S3 tempat aset paket disimpan. • `AutomationAssumeRole` : Nama Sumber Daya Amazon (ARN) dari. `SystemsManager-AutomationAdministrationRole`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan file terkompresi dan `manifest.json` file untuk paket.	Pola ini menyediakan contoh file PowerShell yang dapat diinstal (.msi untuk Windows dan.rpm untuk Linux) dengan skrip instal dan hapus instalan di folder. `account-customization/package` 1. Ganti file yang PowerShell dapat diinstal dengan file Anda sendiri, atau berikan file yang dapat diinstal, instal dan hapus instalan skrip, dan file manifes untuk membuat paket di `account-customization` folder di akun Anda. 2. Sesuaikan `manifest.json` file default yang dihasilkan Terraform di `account-customization` folder sesuai dengan kebutuhan Anda.	DevOps insinyur

Jalankan perintah Terraform untuk menyediakan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi konfigurasi Terraform.	Untuk menerapkan solusi secara otomatis dengan AFT, tekan kode ke AWS CodeCommit:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`$ git add *`
	`$ git commit -m "message"`
	`$ git push`
	
	Anda juga dapat menerapkan solusi ini tanpa menggunakan AFT dengan menjalankan perintah Terraform dari folder. `account-customization` Untuk menginisialisasi direktori kerja yang berisi file Terraform, jalankan:
	
	`$ terraform init`	

Pratinjau perubahan.	Untuk melihat pratinjau perubahan yang akan dilakukan Terraform pada infrastruktur, jalankan perintah: `$ terraform plan` Perintah ini mengevaluasi konfigurasi Terraform untuk menentukan status sumber daya yang diinginkan yang telah dideklarasikan. Ini juga membandingkan keadaan yang diinginkan dengan infrastruktur aktual untuk penyediaan di dalam ruang kerja.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan perubahan.	Jalankan perintah berikut untuk mengimplementasikan perubahan yang Anda buat pada `variables.tf` file: `$ terraform apply`	DevOps insinyur

Validasi sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi pembuatan dokumen SSM.	1. Pada konsol Systems Manager, di panel navigasi kiri, pilih Documents. 2. Pilih tab Dimiliki oleh saya. Anda harus melihat `DistributeSoftwarePackage` dan `AddSoftwarePackageToDistributor` paket.	DevOps insinyur
Validasi keberhasilan penerapan otomatisasi.	1. Pada konsol Systems Manager, di panel navigasi kiri, pilih Otomasi. 2. Dalam daftar eksekusi Otomasi, Anda akan melihat yang terbaru `DistributeSoftwarePackage` dan `AddSoftwarePackageToDistributor` penerapan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Pilih Execution ID untuk memvalidasi bahwa mereka berhasil diselesaikan.	
Validasi bahwa paket disebarkan ke instance akun anggota yang ditargetkan.	1. Pada konsol Systems Manager, di panel navigasi, pilih Run Command. 2. Dalam riwayat Perintah, Anda akan melihat setiap pemanggilan dan statusnya. 3. Pilih Command ID apa pun untuk melihat riwayat penerapan untuk setiap instance target. 4. Pilih ID Instance dan periksa bagian Output untuk distribusi.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Asosiasi Manajer Negara gagal atau terjebak dalam status tertunda.	Lihat informasi pemecahan masalah di Pusat AWS Pengetahuan.
Asosiasi terjadwal gagal dijalankan.	Spesifikasi jadwal Anda mungkin tidak valid. State Manager saat ini tidak mendukung menentukan bulan dalam ekspresi cron untuk asosiasi. Gunakan ekspresi cron atau rate untuk mengonfirmasi jadwal.

Sumber daya terkait

- Distribusi paket terpusat (GitHub repositori)

- Account Factory untuk Terraform (AFT)
- Kasus penggunaan dan praktik terbaik (AWS Systems Manager dokumentasi)

Konfigurasikan logging untuk aplikasi.NET di Amazon CloudWatch Logs dengan menggunakan NLog

Bibhuti Sahu dan Rob Hill (AWS), Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan framework logging NLog open-source untuk mencatat penggunaan dan peristiwa aplikasi.NET di Amazon CloudWatch Logs. Di CloudWatch konsol, Anda dapat melihat pesan log aplikasi dalam waktu dekat. Anda juga dapat mengatur metrik dan mengonfigurasi alarm untuk memberi tahu Anda jika ambang batas metrik terlampaui. Menggunakan CloudWatch Application Insights, Anda dapat melihat dasbor otomatis atau kustom yang menunjukkan potensi masalah untuk aplikasi yang dipantau. CloudWatch Application Insights dirancang untuk membantu Anda dengan cepat mengisolasi masalah yang sedang berlangsung dengan aplikasi dan infrastruktur Anda.

Untuk menulis pesan CloudWatch log ke Log, Anda menambahkan `AWS.Logger.NLog` NuGet paket ke proyek.NET. Kemudian, Anda memperbarui `NLog.config` file untuk menggunakan CloudWatch Log sebagai target.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Aplikasi web atau konsol .NET yang:
  - Menggunakan versi .NET Framework atau .NET Core yang didukung. Untuk informasi selengkapnya, lihat Versi produk.
  - Menggunakan NLog untuk mengirim data log ke Application Insights.
- Izin untuk membuat peran IAM untuk layanan AWS. Untuk informasi selengkapnya, lihat Izin peran layanan.
- Izin untuk meneruskan peran ke layanan AWS. Untuk informasi selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke layanan AWS.

Versi produk

- .NET Framework versi 3.5 atau yang lebih baru

- .NET Core versi 1.0.1, 2.0.0, atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- NLog
- CloudWatch Log Amazon

Arsitektur target

1. Aplikasi.NET menulis data log ke kerangka NLog logging.
2. NLog menulis data log ke CloudWatch Log.
3. Anda menggunakan CloudWatch alarm dan dasbor khusus untuk memantau aplikasi.NET.

Alat

Layanan AWS

- Amazon CloudWatch Application Insights membantu Anda mengamati kesehatan aplikasi dan sumber daya AWS yang mendasarinya.
- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, dan layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Tools for PowerShell adalah sekumpulan PowerShell modul yang membantu Anda menjalankan skrip pada sumber daya AWS Anda dari baris PowerShell perintah.

Alat-alat lainnya

- Logger. NLogadalah NLog target yang mencatat data log ke CloudWatch Log.
- NLogadalah kerangka kerja logging open-source untuk platform.NET yang membantu Anda menulis data log ke target, seperti database, file log, atau konsol.

- PowerShelladalah program manajemen otomatisasi dan konfigurasi Microsoft yang berjalan di Windows, Linux, dan macOS.
- Visual Studio adalah lingkungan pengembangan terintegrasi (IDE) yang mencakup kompiler, alat penyelesaian kode, desainer grafis, dan fitur lain yang mendukung pengembangan perangkat lunak.

Praktik terbaik

- Tetapkan kebijakan retensi untuk grup log target. Ini harus dilakukan di luar NLog konfigurasi. Secara default, data log disimpan di CloudWatch Log tanpa batas waktu.
- Patuhi praktik terbaik untuk mengelola kunci akses AWS.

Epik

Siapkan akses dan alat

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kebijakan IAM.	Ikuti petunjuk dalam Membuat kebijakan menggunakan editor JSON dalam dokumentasi IAM. Masukkan kebijakan JSON berikut, yang memiliki izin hak istimewa paling sedikit yang diperlukan untuk memungkinkan Log membaca dan menulis CloudWatch log. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [`	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"logs:CreateLogGroup",
	"logs:CreateLogStream",
	"logs:GetLogEvents",
	"logs:PutLogEvents",
	"logs:DescribeLogGroups",
	"logs:DescribeLogStreams",
	"logs:PutRetentionPolicy"
	],
	"Resource":
	[
	"*"
	]
	}
	]
	}</pre>	

Buat peran IAM.	Ikuti petunjuk dalam Membuat peran untuk mendelegasikan izin ke layanan AWS dalam dokumentasi IAM. Pilih kebijakan yang Anda buat sebelumnya. Ini adalah peran yang diasumsikan CloudWatch Log untuk melakukan tindakan logging.	Administrator AWS, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS Tools untuk PowerShell.	1. Ikuti petunjuk untuk sistem operasi Anda di Menginstal AWS Tools untuk PowerShell. 2. Gunakan AWS Tools untuk PowerShell cmdlet untuk menyimpan kunci akses dan kunci rahasia Anda di profil. Untuk petunjuk, lihat Mengelola Profil di AWS Tools untuk PowerShell dokumentasi.	AWS Umum

Konfigurasikan NLog

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal NuGet paketnya.	1. Di Visual Studio, pilih File, lalu pilih Buka proyek atau solusi. 2. Pilih proyek tempat Anda ingin menginstal NLog. 3. Di Visual Studio, pilih Tools, NuGet Package Manager, Package Manager Console. 4. Instal `AWS.Logger.NLog` NuGet paket dengan memasukkan perintah berikut. `Install-Package AWS.Logger.NLog -Version 3.1.0`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan target logging.	1. Buka file `NLog.config` . 2. Untuk target`type`, masukkan`AWSTarget` . 3. Untuk target`logGroup`, masukkan nama grup log yang ingin Anda gunakan. Jika grup log belum ada, grup log baru dengan nama yang disediakan akan dibuat secara otomatis. 4. Untuk target`region`, masukkan Wilayah AWS tempat CloudWatch Log dikonfigurasi. 5. Untuk target`profile`, masukkan nama profil yang Anda buat sebelumnya untuk disimpan untuk menyimpan kunci akses dan kunci rahasia. 6. Simpan dan tutup file `NLog.config` . Untuk contoh file konfigurasi, lihat bagian Informasi tambahan dari pola ini. Ketika Anda menjalankan aplikasi Anda, NLog akan menulis pesan log dan mengirimkannya ke CloudWatch Log.	Pengembang aplikasi

Validasi dan pantau log

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi logging.	Ikuti petunjuk di Lihat data log yang dikirim ke CloudWatch Log dalam dokumentasi CloudWatch Log. Validasi bahwa peristiwa log sedang direkam untuk aplikasi. NET. Jika peristiwa log tidak direkam, lihat bagian Pemecahan Masalah dalam pola ini.	AWS Umum
Pantau tumpukan aplikasi. NET.	Konfigurasikan pemantauan CloudWatch sesuai kebutuhan untuk kasus penggunaan Anda. Anda dapat menggunakan Wawasan CloudWatch Log, Wawasan CloudWatch Metrik, dan Wawasan CloudWatch Aplikasi untuk memantau beban kerja.NET Anda. Anda juga dapat mengonfigurasi alarm sehingga Anda dapat menerima peringatan, dan Anda dapat membuat dasbor khusus untuk memantau beban kerja dari satu tampilan.	AWS Umum

Pemecahan masalah

Isu	Solusi
Data log tidak muncul di CloudWatch Log.	Pastikan bahwa kebijakan IAM dilampirkan ke peran IAM yang diasumsikan oleh CloudWatch Log. Untuk petunjuk, lihat bagian Mengatur akses dan alat di bagian Epik.

Sumber daya terkait

- Bekerja dengan grup log dan aliran log (dokumentasi CloudWatch Log)
- CloudWatch Log Amazon dan Kerangka Kerja Logging .NET (postingan blog AWS)

Informasi tambahan

Berikut ini adalah `NLog.config` file sampel.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="nlog" type="NLog.Config.ConfigSectionHandler, NLog" />
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <nlog>
    <extensions>
      <add assembly="NLog.AWS.Logger" />
    </extensions>
    <targets>
      <target name="aws" type="AWSTarget" logGroup="NLog.TestGroup" region="us-east-1"
 profile="demo"/>
    </targets>
    <rules>
      <logger name="*" minlevel="Info" writeTo="aws" />
    </rules>
  </nlog>
</configuration>
```

Salin produk AWS Service Catalog di berbagai akun AWS dan Wilayah AWS

Sachin Vighe dan Santosh Kale, Amazon Web Services

Ringkasan

AWS Service Catalog adalah layanan Regional dan ini berarti portofolio dan produk AWS Service Catalog hanya terlihat di Wilayah AWS tempat produk tersebut dibuat. Jika Anda menyiapkan hub AWS Service Catalog di Wilayah baru, Anda harus membuat ulang produk yang sudah ada dan ini bisa menjadi proses yang memakan waktu.

Pendekatan pola ini membantu menyederhanakan proses ini dengan menjelaskan cara menyalin produk dari hub AWS Service Catalog di akun AWS sumber atau Wilayah AWS ke hub baru di akun tujuan atau Wilayah. Untuk informasi selengkapnya tentang hub AWS Service Catalog dan model spoke, lihat Model hub dan spoke AWS Service Catalog: Cara mengotomatiskan penerapan dan pengelolaan AWS Service Catalog ke banyak akun di AWS Management and Governance Blog.

Pola ini juga menyediakan paket kode terpisah yang diperlukan untuk menyalin produk AWS Service Catalog di seluruh akun atau ke Wilayah lain. Dengan menggunakan pola ini, organisasi Anda dapat menghemat waktu, membuat versi produk yang ada dan sebelumnya tersedia di hub AWS Service Catalog baru, meminimalkan risiko kesalahan manual, dan menskalakan pendekatan di beberapa akun atau Wilayah.

> **Note**
>
> Bagian Epik pola ini menyediakan dua opsi untuk menyalin produk. Anda dapat menggunakan Opsi 1 untuk menyalin produk di seluruh akun atau memilih Opsi 2 untuk menyalin produk di seluruh Wilayah.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Produk AWS Service Catalog yang ada di akun sumber atau Wilayah.
- Hub AWS Service Catalog yang ada di akun tujuan atau Wilayah.

- Jika Anda ingin menyalin produk di seluruh akun, Anda harus membagikan dan kemudian mengimpor portofolio AWS Service Catalog yang berisi produk ke akun tujuan Anda. Untuk informasi selengkapnya tentang hal ini, lihat Berbagi dan mengimpor portofolio dalam dokumentasi AWS Service Catalog.

Batasan

- Produk AWS Service Catalog yang ingin Anda salin di seluruh Wilayah atau akun tidak dapat dimiliki lebih dari satu portofolio.

Arsitektur

Diagram berikut menunjukkan penyalinan produk AWS Service Catalog dari akun sumber ke akun tujuan.

Diagram berikut menunjukkan penyalinan produk AWS Service Catalog dari Wilayah sumber ke Wilayah tujuan.

Tumpukan teknologi

- Amazon CloudWatch
- AWS Identity and Access Management (IAM)
- AWS Lambda
- AWS Service Catalog

Otomatisasi dan skala

Anda dapat menskalakan pendekatan pola ini dengan menggunakan fungsi Lambda yang dapat diskalakan tergantung pada jumlah permintaan yang diterima atau berapa banyak produk AWS Service Catalog yang perlu Anda salin. Untuk informasi selengkapnya tentang ini, lihat penskalaan fungsi Lambda dalam dokumentasi AWS Lambda.

Alat

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Service Catalog membantu Anda mengelola katalog layanan TI secara terpusat yang disetujui untuk AWS. Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.

Kode

Anda dapat menggunakan `cross-account-copy` paket (terlampir) untuk menyalin produk AWS Service Catalog di seluruh akun atau `cross-region-copy` paket (terlampir) untuk menyalin produk di seluruh Wilayah.

`cross-account-copy`Paket berisi file-file berikut:

- `copyconf.properties`— File konfigurasi yang berisi parameter ID akun Wilayah dan AWS untuk menyalin produk di seluruh akun.
- `scProductCopyLambda.py`— Fungsi Python untuk menyalin produk di seluruh akun.
- `createDestAccountRole.sh`— Skrip untuk membuat peran IAM di akun tujuan.
- `createSrcAccountRole.sh`— Skrip untuk membuat peran IAM di akun sumber.
- `copyProduct.sh`— Skrip untuk membuat dan memanggil fungsi Lambda untuk menyalin produk di seluruh akun.

`cross-region-copy`Paket berisi file-file berikut:

- `copyconf.properties`— File konfigurasi yang berisi parameter ID akun Wilayah dan AWS untuk menyalin produk di seluruh Wilayah.
- `scProductCopyLambda.py`— Fungsi Python untuk menyalin produk di seluruh Wilayah.
- `copyProduct.sh`— Skrip untuk membuat peran IAM dan membuat serta memanggil fungsi Lambda untuk menyalin produk di seluruh Wilayah.

Epik

Opsi 1 — Salin produk AWS Service Catalog di seluruh akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file konfigurasi.	1. Unduh `cross-account-copy` paket (terlampir) ke mesin lokal Anda. 2. Perbarui file `copyconf.properties` konfigurasi dengan nilai-nilai berikut: • `srcRegion` — Menyediakan wilayah sumber yang berisi produk. • `destRegion` — Menyediakan Wilayah tujuan untuk produk. • `sourceAccountId` — Berikan ID akun AWS untuk akun sumber Anda. • `destAccountId` — Berikan ID akun AWS untuk akun tujuan Anda.	Administrator AWS, administrator sistem AWS, administrator Cloud

Konfigurasikan kredensional Anda untuk AWS CLI di akun tujuan.	Konfigurasikan kredensional Anda untuk mengakses AWS CLI di akun tujuan Anda dengan menjalankan `aws configure` perintah dan memberikan nilai berikut:
	`$aws configure`
	`AWS Access Key ID [None]: <your_access_key_id>`
		Administrator AWS, administrator sistem AWS, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>AWS Secret Access Key
	 [None]: <your_sec
	ret_access_key>
	Default region name
	 [None]: Region
	Default output format
	 [None]:</pre>
	Untuk informasi selengkapnya tentang hal ini, lihat Dasar-dasar konfigurasi dalam dokumentasi AWS Command Line Interface.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan kredensional Anda untuk AWS CLI di akun sumber.	Konfigurasikan kredensional Anda untuk mengakses AWS CLI di akun sumber Anda dengan menjalankan `aws configure` perintah dan memberikan nilai berikut:
	`$aws configure`
	`AWS Access Key ID [None]: <your_access_key_id>`
	`AWS Secret Access Key [None]: <your_secret_access_key>`
	`Default region name [None]: Region`
	`Default output format [None]:`
	Untuk informasi selengkapnya tentang hal ini, lihat Dasar-dasar konfigurasi dalam dokumentasi AWS Command Line Interface.
		Administrator AWS, administrator sistem AWS, administrator Cloud

Buat peran eksekusi Lambda di akun tujuan Anda.	Jalankan `createDestAccountRole.sh` skrip di akun tujuan Anda. Skrip mengimplementasikan tindakan berikut: • Membuat peran eksekusi Lambda di akun tujuan • Membuat dan melampirkan kebijakan IAM untuk peran eksekusi Lambda	Administrator AWS, administrator sistem AWS, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM lintas akun di akun sumber Anda.	Jalankan `createSrcAccountRole.sh` skrip di akun sumber Anda. Skrip mengimplementasikan tindakan berikut: • Membuat peran IAM lintas akun di akun sumber Anda yang diasumsikan oleh peran eksekusi Lambda di akun tujuan untuk menyalin produk • Membuat dan melampirkan kebijakan IAM untuk peran lintas akun di akun sumber Anda	Administrator AWS, administrator sistem AWS, administrator Cloud
Jalankan skrip CopyProduct di akun tujuan.	Jalankan `copyProduct.sh` skrip di akun tujuan Anda. Skrip mengimplementasikan tindakan berikut: • Membuat dan memanggil fungsi Lambda untuk menyalin produk dari akun sumber ke akun tujuan	Administrator AWS, administrator sistem AWS, administrator Cloud

Opsi 2 — Salin produk AWS Service Catalog dari Wilayah sumber ke Wilayah tujuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file konfigurasi.	1. Unduh `cross-region-copy` paket (terlampir) ke mesin lokal Anda.	Administrator sistem AWS, administrator Cloud, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Perbarui file `copyconf.properties` konfigurasi dengan nilai-nilai berikut: • `srcRegion` — Menyediakan wilayah sumber yang berisi produk. • `destRegion` — Menyediakan Wilayah tujuan untuk produk. • `accountId` — Berikan ID akun AWS Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan kredensional Anda untuk AWS CLI.	Konfigurasikan kredensional Anda untuk mengakses AWS CLI di lingkungan Anda dengan menjalankan `aws configure` perintah dan memberikan nilai berikut:
	`$aws configure`
	`AWS Access Key ID [None]: <your_access_key_id>`
	`AWS Secret Access Key [None]: <your_secret_access_key>`
	`Default region name [None]: Region`
	`Default output format [None]:`
	Untuk informasi selengkapnya tentang hal ini, lihat Dasar-dasar konfigurasi dalam dokumentasi AWS Command Line Interface.
		Administrator AWS, administrator sistem AWS, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip CopyProduct.	Jalankan `copyProduct.sh` skrip di Wilayah tujuan Anda. Skrip mengimplementasikan tindakan berikut: • Membuat peran eksekusi Lambda • Membuat dan melampirkan kebijakan IAM untuk peran eksekusi Lambda • Membuat dan memanggil fungsi Lambda untuk menyalin produk dari Wilayah sumber ke Wilayah tujuan	Administrator AWS, administrator sistem AWS, administrator Cloud

Sumber daya terkait

- Membuat peran eksekusi Lambda (dokumentasi AWS Lambda)
- Buat fungsi Lambda (dokumentasi AWS Lambda)
- Referensi API AWS Service Catalog
- Dokumentasi AWS Service Catalog

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat matriks RACI atau RASCI untuk model operasi cloud

Teddy Germade, Jerome Descreux, Florian Leroux, dan Josselin LE MINEUR, Amazon Web Services

Ringkasan

Cloud Center of Excellence (CCoE) atau CEE (Cloud Enablement Engine) adalah tim yang berdaya dan akuntabel yang berfokus pada kesiapan operasional untuk cloud. Fokus utama mereka adalah mengubah organisasi TI informasi dari model operasi lokal menjadi model operasi cloud. CCoE harus menjadi tim lintas fungsi yang mencakup representasi dari infrastruktur, aplikasi, operasi, dan keamanan.

Salah satu komponen kunci dari model operasi cloud adalah matriks RACI atau matriks RASCI. Ini digunakan untuk menentukan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dukungan (S), berkonsultasi (C), dan diinformasikan (I). Jenis dukungan adalah opsional. Jika Anda memasukkannya, itu disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

Dengan memulai dengan template terlampir, tim CCo E Anda dapat membuat matriks RACI atau RASCI untuk organisasi Anda. Template berisi tim, peran, dan tugas yang umum dalam model operasi cloud. Dasar dari matriks ini adalah tugas-tugas yang berkaitan dengan integrasi operasi dan kemampuan CCo E. Namun, Anda dapat menyesuaikan template ini untuk memenuhi kebutuhan struktur dan kasus penggunaan organisasi Anda.

Tidak ada batasan untuk implementasi matriks RACI. Pendekatan ini bekerja untuk organisasi besar, start-up, dan segala sesuatu di antaranya. Untuk organisasi kecil, sumber daya yang sama dapat mengisi beberapa peran.

Epik

Buat matriks

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi pemangku kepentingan utama.	Identifikasi layanan utama dan manajer tim yang terkait	Manajer proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan tujuan strategis model operasi cloud Anda.	

Sesuaikan templat matriks.	Unduh templat di bagian Lampiran, lalu perbarui matriks RACI atau RASCI sebagai berikut:
	• Di lembar kerja Cloud Teams, perbarui nama aliran CCo E, nama tim, dan deskripsi tim sesuai kebutuhan untuk organisasi Anda.
	• Pada lembar kerja Cloud Roles, perbarui peran, nama tim, dan deskripsi peran yang diperlukan untuk organisasi Anda.
	• Pada lembar kerja RASCI, perbarui yang berikut ini sesuai kebutuhan untuk organisasi Anda:
	  • Di baris 1 dan kolom A, perbarui aliran CCo E.
	  • Di baris 2, perbarui nama tim.
	  • Di baris 3, perbarui nama peran.
	  • Di kolom D dan E, perbarui bidang umum dan aktivitas yang ingin Anda sertakan dalam bagan RASCI Anda.
		Manajer proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Rencanakan pertemuan.	1. Mengkomunikasikan tujuan RASCI kepada semua pemangku kepentingan. 2. Rencanakan satu atau lebih pertemuan sehingga perwakilan yang diberdaya kan dari masing-masing tim dapat hadir.	Manajer proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi matriks.	Dalam pertemuan dengan semua pemangku kepentingan, lakukan hal berikut: 1. Konfirmasikan bahwa perwakilan dari masing-masing tim hadir. Partisipasi tim adalah wajib sehingga Anda dapat secara akurat menetapkan jenis tanggung jawab untuk setiap tugas. 2. Tinjau apa itu matriks RASCI dan tujuannya dengan para peserta. 3. Tinjau model tanggung jawab bersama dengan peserta sehingga mereka memahami ruang lingkup tanggung jawab organisasi mereka untuk keamanan di cloud. 4. Pada lembar kerja RASCI, untuk setiap tugas atau aktivitas, lengkapi kolom F hingga AN untuk menetapkan jenis tanggung jawab berikut: • Bertanggung jawab (R) — Peran ini bertanggung jawab untuk melakukan pekerjaan untuk menyelesaikan tugas. • Akuntabel (A) — Peran ini bertanggung jawab	Manajer proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk memastikan tugas selesai. Peran ini juga bertanggung jawab untuk memastikan prasyarat terpenuhi dan mendelegasikan tugas kepada mereka yang bertanggung jawab. • Support (S) — Peran ini membantu mereka yang bertanggung jawab menyelesaikan tugas. Jenis tanggung jawab ini opsional, dan Anda dapat memilih untuk mengecualikannya untuk membuat matriks RACI yang lebih tradisional. • Konsultasikan (C) — Peran ini harus dikonsultasikan untuk pendapat atau keahlian tentang tugas tersebut. Tergantung pada tugasnya, jenis tanggung jawab ini mungkin tidak diperlukan. • Informed (I) — Peran ini harus selalu up to date pada kemajuan tugas dan diberitahu ketika tugas selesai. • Kosong — Peran ini tidak terlibat dalam aktivitas atau tugas.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bagikan matriks RASCI.	Ketika matriks RACI atau RASCI selesai, minta disetujui oleh kepemimpinan. Simpan di repositori bersama atau lokasi pusat tempat semua pemangku kepentingan dapat mengaksesnya. Kami menyarankan Anda menggunakan proses kontrol dokumen standar untuk merekam dan menyetujui revisi matriks.	Manajer proyek

Sumber daya terkait

- Model tanggung jawab bersama AWS

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat alarm untuk metrik kustom menggunakan deteksi anomali Amazon CloudWatch

Ram Kandaswamy dan Raheem Jiwani, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, Anda dapat menggunakan Amazon CloudWatch untuk membuat alarm yang memantau metrik dan mengirim notifikasi atau secara otomatis membuat perubahan jika ambang batas dilanggar.

Agar tidak dibatasi oleh ambang batas statis, Anda dapat membuat alarm berdasarkan pola masa lalu dan yang memberi tahu Anda jika metrik tertentu berada di luar jendela operasi normal. Misalnya, Anda dapat memantau waktu respons API Anda dari Amazon API Gateway dan menerima pemberitahuan tentang anomali yang mencegah Anda memenuhi perjanjian tingkat layanan (SLA).

Pola ini menjelaskan cara menggunakan deteksi CloudWatch anomali untuk metrik kustom. Pola ini menunjukkan cara membuat metrik kustom di Amazon CloudWatch Logs Insights atau menerbitkan metrik kustom dengan fungsi AWS Lambda, lalu mengatur deteksi anomali dan membuat notifikasi menggunakan Amazon Simple Notification Service (Amazon SNS).

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- Topik SNS yang ada, dikonfigurasi untuk mengirim pemberitahuan email. Untuk informasi selengkapnya tentang ini, lihat Memulai Amazon SNS di dokumentasi Amazon SNS.
- Aplikasi yang ada, dikonfigurasi dengan CloudWatch Log.

Batasan

- CloudWatch metrik tidak mendukung interval waktu milidetik. Untuk informasi selengkapnya tentang perincian metrik reguler dan kustom, lihat Amazon. CloudWatch FAQs

Arsitektur

Diagram menunjukkan alur kerja berikut:

1. Log yang menggunakan metrik yang dibuat dan diperbarui oleh CloudWatch Log dialirkan ke. CloudWatch
2. Alarm dimulai berdasarkan ambang batas dan mengirimkan peringatan ke topik SNS.
3. Amazon SNS mengirimi Anda pemberitahuan email.

Tumpukan teknologi

- CloudWatch
- AWS Lambda
- Amazon SNS

Alat

- Amazon CloudWatch menyediakan solusi pemantauan yang andal, terukur, dan fleksibel.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa menyediakan atau mengelola server.
- Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan.

Epik

Siapkan deteksi anomali untuk metrik khusus

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 1 - Buat metrik khusus dengan fungsi Lambda.	Unduh `lambda_function.py` file (terlampir) dan kemudian ganti `lambda_function.py` file sampel di aws-lambda-developer-guiderepositori pada Dokumentasi AWS. GitHub Ini memberi Anda contoh fungsi Lambda yang mengirimkan metrik khusus	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ke Log. CloudWatch Fungsi Lambda menggunakan API Boto3 untuk diintegrasikan dengan. CloudWatch Setelah menjalankan fungsi Lambda, Anda dapat masuk ke AWS Managemen t Console, membuka CloudWatch konsol, dan metrik yang dipublikasikan tersedia di bawah namespace yang dipublikasikan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 2 - Buat metrik khusus dari grup CloudWatch log.	Masuk ke AWS Management Console, buka CloudWatch konsol, lalu pilih Grup log. Pilih grup log yang ingin Anda buat metrik. Pilih Tindakan dan kemudian pilih Buat filter metrik. Untuk pola Filter, masukkan pola filter yang ingin Anda gunakan. Untuk informasi selengkapnya, lihat Filter dan sintaks pola dalam CloudWatch dokumentasi. Untuk menguji pola filter Anda, masukkan satu atau beberapa peristiwa log di bawah Pola Uji. Setiap log acara harus dalam satu baris, karena jeda baris yang digunakan untuk memisahkan log acara di kotak pesan log acara (Pesan log acara). Setelah menguji pola, Anda dapat memasukkan nama dan nilai untuk metrik Anda di bawah Detail metrik. Untuk informasi selengkapnya dan langkah-langkah untuk membuat metrik kustom, lihat Membuat filter metrik untuk grup log dalam CloudWatch dokumentasi.	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat alarm untuk metrik kustom Anda.	Di CloudWatch konsol, pilih Alarm dan kemudian pilih Buat Alarm. Pilih Pilih metrik dan masukkan nama metrik yang Anda buat sebelumnya ke dalam kotak pencarian. Pilih tab Metrik grafik dan konfigura sikan opsi sesuai dengan kebutuhan Anda. Di bawah Kondisi, pilih Deteksi anomali alih-alih Ambang batas statis. Ini menunjukkan kepada Anda sebuah band berdasarkan dua standar standar standar deviasi. Anda dapat mengatur ambang batas dan menyesuaikannya sesuai dengan kebutuhan Anda. Pilih Berikutnya. **Note** Band ini dinamis dan tergantung pada kualitas titik data. Saat Anda mulai mengumpulkan lebih banyak data, pita dan ambang batas diperbarui secara otomatis.	DevOps insinyur, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan notifikasi SNS.	Di bawah Pemberitahuan, pilih topik SNS untuk memberi tahu saat alarm dalam `ALARM` status, status, atau `OK` `INSUFFICIENT_DATA` status. Agar alarm mengirimkan beberapa notifikasi untuk status alarm yang sama atau untuk status alarm yang berbeda, silakan pilih Tambahkan notifikasi. Pilih Berikutnya. Masukkan sebuah nama dan deskripsi untuk alarm tersebut. Nama hanya boleh berisi karakter ASCII. Lalu pilih Selanjutnya. Di bawah Pratinjau dan buat, konfirmasikan bahwa informasi dan kondisi sudah benar, lalu pilih Buat alarm.	DevOps insinyur, AWS DevOps

Sumber daya terkait

- Menerbitkan metrik khusus ke CloudWatch
- Menggunakan CloudWatch deteksi anomali
- Acara alarm dan Amazon EventBridge
- Apa praktik terbaik yang harus diikuti saat mendorong metrik khusus ke Cloud Watch? (video)
- Pengantar Wawasan CloudWatch Aplikasi (video)
- Mendeteksi anomali dengan CloudWatch (video)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Buat AWS Cloud9 IDE yang menggunakan volume Amazon EBS dengan enkripsi default

Janardhan Malyala dan Dhrubajyoti Mukherjey, Amazon Web Services

Ringkasan

Pemberitahuan: AWS Cloud9 tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS Cloud9 dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

Anda dapat menggunakan enkripsi secara default untuk menerapkan enkripsi volume Amazon Elastic Block Store (Amazon EBS) dan salinan snapshot di Amazon Web Services (AWS) Cloud.

Anda dapat membuat lingkungan pengembangan terintegrasi AWS Cloud9 (IDE) yang menggunakan volume EBS yang dienkripsi secara default. Namun, peran terkait layanan AWS Identity and Access Management (IAM) untuk AWS Cloud9 memerlukan akses ke kunci AWS Key Management Service (AWS KMS) untuk volume EBS ini. Jika akses tidak disediakan, AWS Cloud9 IDE mungkin gagal diluncurkan dan proses debug mungkin sulit dilakukan.

Pola ini menyediakan langkah-langkah untuk menambahkan peran terkait layanan untuk AWS Cloud9 ke kunci AWS KMS yang digunakan oleh volume EBS Anda. Pengaturan yang dijelaskan oleh pola ini membantu Anda berhasil membuat dan meluncurkan IDE yang menggunakan volume EBS dengan enkripsi secara default.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Enkripsi default diaktifkan untuk volume EBS. Untuk informasi selengkapnya tentang enkripsi secara default, lihat enkripsi Amazon EBS di dokumentasi Amazon Elastic Compute Cloud EC2 (Amazon).
- Kunci KMS yang dikelola pelanggan yang sudah ada untuk mengenkripsi volume EBS Anda.

> **Note**
>
> Anda tidak perlu membuat peran terkait layanan untuk AWS Cloud9. Saat Anda membuat lingkungan pengembangan AWS Cloud9, AWS Cloud9 membuat peran terkait layanan untuk Anda.

Arsitektur

Tumpukan teknologi

- AWS Cloud9
- IAM
- AWS KMS

Alat

- AWS Cloud9 adalah lingkungan pengembangan terintegrasi (IDE) yang membantu Anda membuat kode, membangun, menjalankan, menguji, dan men-debug perangkat lunak. Ini juga membantu Anda merilis perangkat lunak ke AWS Cloud.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans Amazon Elastic Compute Cloud (Amazon). EC2
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.

Epik

Temukan nilai kunci enkripsi default

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Catat nilai kunci enkripsi default untuk volume EBS.	Masuk ke AWS Management Console dan buka EC2 konsol Amazon. Pilih EC2 dasbor, lalu pilih Perlindungan dan keamanan data di atribut Akun. Di bagian enkripsi EBS, salin dan catat nilainya dalam kunci enkripsi Default.	Arsitek awan, DevOps insinyur

Menyediakan akses ke kunci AWS KMS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan AWS Cloud9 akses ke kunci KMS untuk volume EBS.	1. Buka konsol AWS KMS, lalu pilih Kunci yang dikelola pelanggan. Pilih kunci AWS KMS yang digunakan untuk enkripsi Amazon EBS, lalu pilih tombol Lihat. 2. Pada tab Kebijakan kunci, konfirmasikan bahwa Anda dapat melihat bentuk teks kebijakan kunci. Jika Anda tidak dapat melihat formulir teks, pilih Beralih ke tampilan kebijakan. 3. Pilih Edit. Tambahkan kode di bagian Informasi tambahan ke kebijakan, lalu pilih Simpan perubahan	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	. Perubahan kebijakan mengizinkan peran terkait layanan untuk AWS Cloud9`AWSServiceRoleForAWSCloud9` , untuk mengakses kunci. Untuk informasi selengkapnya tentang memperbarui kebijakan utama, lihat Cara mengubah kebijakan kunci (dokumentasi AWS KMS). **Important** Peran terkait layanan untuk AWS Cloud9 dibuat secara otomatis saat Anda meluncurkan IDE pertama. Untuk informasi selengkapnya, lihat Membuat peran terkait layanan dalam dokumentasi AWS Cloud9.	

Buat dan luncurkan IDE

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dan luncurkan AWS Cloud9 IDE.	Buka konsol AWS Cloud9 dan pilih Buat lingkungan. Konfigurasikan IDE sesuai dengan kebutuhan Anda	Arsitek awan, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan mengikuti langkah-langkah dari Membuat EC2 lingkungan dalam dokumentasi AWS Cloud9.	

Sumber daya terkait

- Enkripsi volume EBS yang digunakan oleh AWS Cloud9
- Membuat peran terkait layanan untuk AWS Cloud9
- Buat EC2 lingkungan di AWS Cloud9

Informasi tambahan

Pembaruan kebijakan utama AWS KMS

Ganti <aws_accountid> dengan ID akun AWS Anda.

```
{
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<aws_accountid>:role/aws-service-role/
cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "arn:aws:iam::<aws_accountid>:role/aws-service-role/
cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
            },
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
```

Menggunakan kunci lintas akun

Jika Anda ingin menggunakan kunci KMS lintas akun, Anda harus menggunakan hibah yang dikombinasikan dengan kebijakan kunci KMS. Ini memungkinkan akses lintas akun ke kunci. Di akun yang sama yang Anda gunakan untuk membuat lingkungan Cloud9, jalankan perintah berikut di terminal.

```
aws kms create-grant \
 --region <Region where Cloud9 environment is created> \
 --key-id <The cross-account KMS key ARN> \
 --grantee-principal arn:aws:iam::<The account where Cloud9 environment is
 created>:role/aws-service-role/cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9 \
 --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"
```

Setelah menjalankan perintah ini, Anda dapat membuat lingkungan Cloud9 dengan menggunakan enkripsi EBS dengan kunci di akun yang berbeda.

Buat CloudWatch dasbor Amazon berbasis tag secara otomatis

Janak Vadaria, Vinodkumar Mandalapu, dan RAJNEESH TYAGI, Amazon Web Services

Ringkasan

Membuat CloudWatch dasbor Amazon yang berbeda secara manual dapat memakan waktu, terutama ketika Anda harus membuat dan memperbarui beberapa sumber daya untuk secara otomatis menskalakan lingkungan Anda. Solusi yang membuat dan memperbarui CloudWatch dasbor Anda secara otomatis dapat menghemat waktu Anda. Pola ini membantu Anda menerapkan AWS Cloud Development Kit (AWS CDK) pipeline otomatis penuh yang membuat dan memperbarui CloudWatch dasbor untuk AWS sumber daya Anda berdasarkan peristiwa perubahan tag, untuk menampilkan metrik Sinyal Emas.

Dalam rekayasa keandalan situs (SRE), Sinyal Emas mengacu pada serangkaian metrik komprehensif yang menawarkan pandangan luas layanan dari perspektif pengguna atau konsumen. Metrik ini terdiri dari latensi, lalu lintas, kesalahan, dan saturasi. Untuk informasi lebih lanjut, lihat Apa itu Rekayasa Keandalan Situs (SRE)? di situs AWS web.

Solusi yang diberikan oleh pola ini didorong oleh peristiwa. Setelah digunakan, ia terus memantau peristiwa perubahan tag dan secara otomatis memperbarui CloudWatch dasbor dan alarm.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Prasyarat untuk v2 AWS CDK
- Lingkungan bootstrapped pada AWS
- Python versi 3
- AWS SDK untuk Python (Boto3), diinstal
- Node.js versi 18 atau yang lebih baru
- Manajer paket node (npm), diinstal dan dikonfigurasi untuk AWS CDK
- Keakraban sedang (level 200) dengan dan AWS CDK AWS CodePipeline

Batasan

Solusi ini saat ini membuat dasbor otomatis hanya untuk layanan AWS berikut:

- Amazon Relational Database Service (Amazon RDS)
- AWS Auto Scaling
- Layanan Pemberitahuan Sederhana Amazon (Amazon SNS)
- Amazon DynamoDB
- AWS Lambda

Arsitektur

Tumpukan teknologi target

- CloudWatch dasbor
- CloudWatch alarm

Arsitektur target

1. Peristiwa perubahan AWS tag untuk tag aplikasi yang dikonfigurasi atau perubahan kode memulai pipeline AWS CodePipeline untuk membangun dan menerapkan dasbor yang diperbarui CloudWatch .
2. AWS CodeBuild menjalankan skrip Python untuk menemukan sumber daya yang telah mengkonfigurasi tag dan menyimpan sumber daya IDs dalam file lokal di lingkungan. CodeBuild
3. CodeBuild menjalankan cdk synth untuk menghasilkan CloudFormation template yang menyebarkan CloudWatch dasbor dan alarm.
4. CodePipeline menyebarkan CloudFormation template ke yang ditentukan Akun AWS dan Wilayah.
5. Ketika CloudFormation tumpukan telah berhasil digunakan, Anda dapat melihat CloudWatch dasbor dan alarm.

Otomatisasi dan skala

Solusi ini telah diotomatisasi dengan menggunakan AWS CDK. Anda dapat menemukan kode di Dasbor Sinyal GitHub Emas di CloudWatch repositori Amazon. Untuk penskalaan tambahan dan untuk membuat dasbor khusus, Anda dapat mengonfigurasi beberapa kunci dan nilai tag.

Alat

Layanan Amazon

- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data real-time dari berbagai sumber, termasuk AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS Command Line Interface (AWS CLI) adalah alat open source yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Praktik terbaik

Sebagai praktik keamanan terbaik, Anda dapat menggunakan enkripsi dan otentikasi untuk repositori sumber yang terhubung ke saluran pipa Anda. Untuk praktik terbaik tambahan, lihat praktik CodePipeline terbaik dan kasus penggunaan dalam CodePipeline dokumentasi.

Epik

Konfigurasikan dan terapkan aplikasi sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan dan terapkan aplikasi sampel.	1. Kloning repositori kode GitHub sampel dengan menggunakan perintah:	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/aws-samples/golden-signals-dashboards-sample-app` 2. Arahkan ke repositori kloning di komputer Anda dan buka `src/project-settings.ts` file dengan editor pilihan Anda. 3. Ubah nilai `projectSettings` konstan sesuai dengan tag AWS sumber daya dan pemetaan aplikasi Anda. 4. Mengatur`AWS_ACCOUNT` ,`AWS_REGION` , dan variabel `GS_DASHBOARD_INSTANCE` lingkungan: • Setel `AWS_ACCOUNT` ke ID akun AWS akun Anda. • Setel `AWS_REGION` ke Wilayah tempat Anda ingin menyebarkan aplikasi sampel. • Setel `GS_DASHBOARD_INSTANCE` ke `devtest`,, atau`prod`, tergantung pada lingkungan pengembangan Anda. (Kami merekomendasikan `test`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	untuk prosedur pengujian yang dijelaskan dalam pola ini.) 5. Siapkan AWS CLI dengan AWS kredensil Anda. Untuk informasi selengkapnya, lihat Mengatur dan melihat pengaturan konfigurasi menggunakan perintah dalam AWS CLI dokumentasi. 6. Jalankan perintah berikut untuk menyebarkan aplikasi sampel dasbor Sinyal Emas: `sh deploy.sh`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat dasbor dan alarm secara otomatis.	Setelah menerapkan aplikasi sampel, Anda dapat membuat sumber daya apa pun yang didukung solusi ini dengan nilai tag yang diharapkan, yang secara otomatis akan membuat dasbor dan alarm yang ditentukan. Untuk menguji solusi ini, buat AWS Lambda fungsi: 1. Masuk ke Konsol Manajemen AWS Wilayah AWS tempat Anda menerapkan aplikasi sampel. 2. Buka konsol Lambda di. https://console.aws.amazon.com/lambda/ 3. Pilih Buat fungsi, lalu masukkan nama fungsi. 4. Di panel Pengaturan lanjutan, pilih Aktifkan tag, lalu pilih Tambahkan tag baru. Masukkan kunci dan nilai berikut: • Kunci: `AutoDashboard` • Nilai: `True` 5. Pilih Buat fungsi. Fungsi Lambda segera memulai pipeline kode, yang membuat dasbor dan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	alarm untuk fungsi Lambda tertentu secara otomatis. 6. Untuk melihat dasbor dan alarm otomatis, buka CloudWatch konsol di. https://console.aws.amazon.com/cloudwatch/ Anda dapat melihat dasbor dan alarm khusus untuk fungsi yang Anda tentukan dalam `projectSettings` konstanta (APP1-lambda secara default). 7. Pilih dasbor untuk fungsi Lambda untuk melihat dasbor otomatis tambahan yang dibuat sebagai bagian dari solusi ini. 8. Ulangi langkah-langkah ini untuk layanan lain, seperti Amazon RDS, Amazon SNS, dan DynamoDB AWS Auto Scaling, untuk menghasilkan dasbor terkait. Untuk contoh Amazon RDS, lihat bagian Informasi tambahan.	

Hapus aplikasi sampel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus `golden-signals-dashboard` konstruksinya.	1. Untuk menghapus semua CloudFormation tumpukan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang dibuat oleh aplikasi sampel, Anda harus mengkonfigurasi ulang variabel`AWS_ACCOUNT` ,`AWS_REGION` , dan `GS_DASHBOARD_INSTANCE` lingkungan. `destroy.sh` Perintah membutuhkan konfigurasi ini. • `AWS_ACCOUNT` adalah ID akun AWS akun Anda. • `AWS_REGION` adalah Wilayah tempat Anda menerapkan aplikasi sampel Anda. • `GS_DASHBOARD_INSTANCE` adalah`dev`,`test`, atau`prod`, berdasarkan pengaturan Anda sebelumnya. 2. Siapkan AWS CLI dengan AWS kredensil Anda. 3. Jalankan perintah berikut untuk menghapus aplikasi sampel dan semua CloudFormation tumpukan terkait: `sh destroy.sh`	

Pemecahan masalah

Isu	Solusi
Perintah Python tidak ditemukan (mengacu pada`findresources.sh` , baris 8).	Periksa versi instalasi Python Anda. Jika Anda telah menginstal Python versi 3, ganti `python` dengan `python3` on line 8 `resources.sh` file, dan jalankan `sh deploy.sh` perintah lagi untuk menyebarkan solusi.

Sumber daya terkait

- Bootstrapping (dokumentasi)AWS CDK
- Menggunakan profil bernama (AWS CLI dokumentasi)
- AWS CDK Lokakarya

Informasi tambahan

Ilustrasi berikut menunjukkan dasbor sampel untuk Amazon RDS yang dibuat sebagai bagian dari solusi ini.

Dokumentasikan desain AWS landing zone

Michael Daehnert, Florian Langer, dan Michael Lodemann, Amazon Web Services

Ringkasan

Landing zone adalah lingkungan multi-akun yang dirancang dengan baik yang didasarkan pada praktik terbaik keamanan dan kepatuhan. Ini adalah wadah di seluruh perusahaan yang menampung semua unit organisasi Anda (OUs),, pengguna Akun AWS, dan sumber daya lainnya. Sebuah landing zone dapat disesuaikan dengan kebutuhan perusahaan dari berbagai ukuran. AWS memiliki dua opsi untuk membuat landing zone Anda: menggunakan landing zone berbasis layanan AWS Control Toweratau landing zone khusus yang Anda bangun. Setiap opsi membutuhkan tingkat AWS pengetahuan yang berbeda.

AWS dibuat AWS Control Tower untuk membantu Anda menghemat waktu dengan mengotomatiskan pengaturan landing zone. AWS Control Tower dikelola oleh AWS dan menggunakan praktik dan pedoman terbaik untuk membantu Anda menciptakan lingkungan dasar Anda. AWS Control Tower menggunakan layanan terintegrasi, seperti AWS Service Catalogdan AWS Organizations, untuk menyediakan akun di landing zone Anda dan mengelola akses ke akun tersebut.

AWS Proyek landing zone bervariasi dalam persyaratan, detail implementasi, dan item tindakan operasional. Ada aspek kustomisasi yang perlu ditangani dengan setiap implementasi landing zone. Ini termasuk (tetapi tidak terbatas pada) bagaimana manajemen akses ditangani, tumpukan teknologi mana yang digunakan, dan apa persyaratan pemantauan untuk keunggulan operasional. Pola ini menyediakan template yang membantu Anda mendokumentasikan proyek landing zone Anda. Dengan menggunakan template, Anda dapat mendokumentasikan proyek Anda lebih cepat dan membantu tim pengembangan dan operasi Anda memahami landing zone Anda.

Prasyarat dan batasan

Batasan

Pola ini tidak menggambarkan apa itu landing zone atau bagaimana menerapkannya. Untuk informasi selengkapnya tentang topik ini, lihat bagian Sumber daya terkait.

Epik

Buat dokumen desain

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi pemangku kepentingan utama.	Identifikasi manajer layanan dan tim utama yang terhubung ke landing zone Anda.	Manajer proyek
Sesuaikan template.	Unduh templat di bagian Lampiran, lalu perbarui template sebagai berikut: 1. Hapus bagian apa pun yang tidak berlaku untuk landing zone atau proses organisasi Anda. 2. Tambahkan bagian apa pun yang unik untuk organisasi Anda.	Manajer proyek
Lengkapi template.	Dalam pertemuan dengan para pemangku kepentingan atau dengan menggunakan suatu write-and-review proses, lengkapi template sebagai berikut: 1. Gunakan panduan dan informasi di kotak biru untuk menyelesaikan setiap bagian. 2. Ganti atau hapus bidang kuning apa pun dengan nilai khusus untuk organisasi Anda.	Manajer proyek

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Ganti atau hapus bidang gambar apa pun dengan arsitektur kustom atau diagram alur Anda. 4. Lengkapi bagian Riwayat Revisi dan Kontributor pada template.	
Bagikan dokumen desain.	Ketika dokumentasi desain landing zone Anda selesai, simpan di repositori bersama atau lokasi pusat di mana semua pemangku kepentingan dapat mengaksesnya. Kami menyarankan Anda menggunakan proses kontrol dokumen standar untuk merekam dan menyetujui revisi dokumen desain.	Manajer proyek

Sumber daya terkait

- AWS Control Tower dokumentasi
  - Rencanakan AWS Control Tower landing zone
  - AWS strategi multi-akun untuk AWS Control Tower landing zone Anda
  - Kiat administratif untuk pengaturan landing zone
  - Harapan untuk konfigurasi landing zone
- Kustomisasi untuk AWS Control Tower (AWS Solutions Library)
- Menyiapkan lingkungan multi-akun AWS yang aman dan terukur (Panduan AWS Preskriptif)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Tingkatkan kinerja operasional dengan mengaktifkan Amazon DevOps Guru di beberapa Wilayah AWS, akun, dan OUs dengan AWS CDK

Dr. Rahul Sharad Gaikwad, Amazon Web Services

Ringkasan

Pola ini menunjukkan langkah-langkah untuk mengaktifkan layanan Amazon DevOps Guru di beberapa Wilayah, akun, dan unit organisasi () Amazon Web Services (AWSOUs) dengan menggunakan AWS Cloud Development Kit (AWS CDK) di. TypeScript Anda dapat menggunakan tumpukan AWS CDK untuk menerapkan CloudFormation StackSets AWS dari akun AWS administrator (utama) untuk mengaktifkan DevOps Amazon Guru di beberapa akun, alih-alih masuk ke setiap akun dan DevOps mengaktifkan Guru satu per satu untuk setiap akun.

Amazon DevOps Guru menyediakan fitur operasi kecerdasan buatan (AIOps) untuk membantu Anda meningkatkan ketersediaan aplikasi dan menyelesaikan masalah operasional dengan lebih cepat. DevOps Guru mengurangi upaya manual Anda dengan menerapkan rekomendasi yang didukung pembelajaran mesin (ML), tanpa memerlukan keahlian ML apa pun. DevOps Guru menganalisis sumber daya dan data operasional Anda. Jika mendeteksi anomali apa pun, ia menyediakan metrik, peristiwa, dan rekomendasi untuk membantu Anda mengatasi masalah tersebut.

Pola ini menjelaskan tiga opsi penerapan untuk mengaktifkan Amazon DevOps Guru:

- Untuk semua sumber daya tumpukan di beberapa akun dan Wilayah
- Untuk semua sumber daya tumpukan OUs
- Untuk sumber daya tumpukan tertentu di beberapa akun dan Wilayah

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. (Lihat Menginstal, memperbarui, dan menghapus instalan AWS CLI dalam dokumentasi AWS CLI.)
- AWS CDK Toolkit, diinstal dan dikonfigurasi. (Lihat AWS CDK Toolkit dalam dokumentasi AWS CDK.)
- Node Package Manager (npm), diinstal dan dikonfigurasi untuk AWS CDK di. TypeScript (Lihat Mengunduh dan menginstal Node.js dan npm di dokumentasi npm.)

- Python3 diinstal dan dikonfigurasi, untuk menjalankan skrip Python untuk menyuntikkan lalu lintas ke dalam aplikasi tanpa server sampel. (Lihat Pengaturan dan Penggunaan Python dalam dokumentasi Python.)
- Pip, diinstal dan dikonfigurasi untuk menginstal pustaka permintaan Python. (Lihat petunjuk pemasangan pip di PyPI situs web.)

Versi produk

- AWS CDK Toolkit versi 1.107.0 atau yang lebih baru
- npm versi 7.9.0 atau yang lebih baru
- Node.js versi 15.3.0 atau yang lebih baru

Arsitektur

Teknologi

Arsitektur untuk pola ini mencakup layanan berikut:

- DevOps Guru Amazon
- AWS CloudFormation
- Amazon API Gateway
- AWS Lambda
- Amazon DynamoDB
- Amazon CloudWatch
- AWS CloudTrail

Tumpukan AWS CDK

Pola ini menggunakan tumpukan AWS CDK berikut:

- `CdkStackSetAdminRole`— Membuat peran administrator AWS Identity and Access management (IAM) untuk membangun hubungan kepercayaan antara administrator dan akun target.
- `CdkStackSetExecRole`— Membuat peran IAM untuk mempercayai akun administrator.

- `CdkDevopsGuruStackMultiAccReg`— Mengaktifkan DevOps Guru di beberapa Wilayah AWS dan akun untuk semua tumpukan, dan menyiapkan notifikasi Amazon Simple Notification Service (Amazon SNS).
- `CdkDevopsGuruStackMultiAccRegSpecStacks`— Mengaktifkan DevOps Guru di beberapa Wilayah AWS dan akun untuk tumpukan tertentu, dan menyiapkan notifikasi Amazon SNS.
- `CdkDevopsguruStackOrgUnit`— Mengaktifkan DevOps Guru di seluruh OUs, dan mengatur notifikasi Amazon SNS.
- `CdkInfrastructureStack`— Menyebarkan komponen aplikasi tanpa server sampel seperti API Gateway, Lambda, dan DynamoDB di akun administrator untuk menunjukkan injeksi kesalahan dan pembuatan wawasan.

Contoh arsitektur aplikasi

Diagram berikut menggambarkan arsitektur aplikasi tanpa server sampel yang telah digunakan di beberapa akun dan Wilayah. Pola ini menggunakan akun administrator untuk menerapkan semua tumpukan AWS CDK. Ini juga menggunakan akun administrator sebagai salah satu akun target untuk mengatur DevOps Guru.

1. Ketika DevOps Guru diaktifkan, pertama-tama ini mendasarkan perilaku setiap sumber daya dan kemudian menyerap data operasional dari CloudWatch metrik yang dijual.
2. Jika mendeteksi anomali, itu berkorelasi dengan peristiwa dari CloudTrail, dan menghasilkan wawasan.
3. Wawasan memberikan urutan peristiwa yang berkorelasi bersama dengan rekomendasi yang ditentukan untuk memungkinkan operator mengidentifikasi sumber daya pelakunya.
4. Amazon SNS mengirimkan pesan notifikasi ke operator.

Otomatisasi dan skala

GitHub Repositori yang disediakan dengan pola ini menggunakan AWS CDK sebagai alat infrastruktur sebagai kode (IAc) untuk membuat konfigurasi untuk arsitektur ini. AWS CDK membantu Anda mengatur sumber daya dan mengaktifkan Guru DevOps di beberapa akun AWS, Wilayah, dan. OUs

Alat

Layanan AWS

- AWS CDK — AWS Cloud Development Kit (AWS CDK) membantu Anda mendefinisikan infrastruktur cloud Anda sebagai kode dalam salah satu dari lima bahasa pemrograman yang didukung: TypeScript,, JavaScript Python, Java, dan C#.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat terpadu yang menyediakan antarmuka baris perintah yang konsisten untuk berinteraksi dengan layanan dan sumber daya AWS.

Kode

Kode sumber untuk pola ini tersedia di GitHub, di repositori Sampel CDK DevOps Guru Amazon. Kode AWS CDK ditulis dalam TypeScript. Untuk mengkloning dan menggunakan repositori, ikuti instruksi di bagian selanjutnya.

> **Important**
>
> Beberapa cerita dalam pola ini termasuk contoh perintah AWS CDK dan AWS CLI yang diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan garis miring terbalik (\) di akhir setiap baris dengan tanda sisipan (^).

Epik

Siapkan sumber daya AWS untuk penerapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan profil bernama AWS.	Siapkan profil bernama AWS Anda sebagai berikut untuk menerapkan tumpukan di lingkungan multi-akun. Untuk akun administrator: `$aws configure --profile administrator`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>AWS Access Key ID
	 [****]: <your-adm
	inistrator-access-
	key-ID>
	AWS Secret Access
	 Key [****]: <your-adm
	inistrator-secret-
	access-key>
	Default region name
	 [None]: <your-adm
	inistrator-region>
	Default output format
	 [None]: json</pre>
	Untuk akun target:
	<pre>$aws configure --profile
	 target
	AWS Access Key ID
	 [****: <your-target-
	access-key-ID>
	AWS Secret Access Key
	 [****]: <your-target-
	secret-access-key>
	Default region name
	 [None]: <your-target-
	region>
	Default output format
	 [None]: json</pre>
	Untuk informasi selengkap nya, lihat Menggunakan profil bernama dalam dokumentasi AWS CLI.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi konfigurasi profil AWS.	(Opsional) Anda dapat memverifikasi konfigurasi profil AWS di `config` file `credentials` dan dengan mengikuti petunjuk di Setel dan melihat pengaturan konfigurasi dalam dokumentasi AWS CLI.	DevOps insinyur
Verifikasi versi AWS CDK.	Verifikasi versi AWS CDK Toolkit dengan menjalankan perintah berikut: `$cdk --version` Pola ini membutuhkan versi 1.107.0 atau yang lebih baru. Jika Anda memiliki AWS CDK versi sebelumnya, ikuti petunjuk dalam dokumentasi AWS CDK untuk memperbaruinya.	DevOps insinyur
Kloning kode proyek.	Kloning GitHub repositori untuk pola ini dengan menggunakan perintah: `$git clone https://github.com/aws-samples/amazon-devopsguru-cdk-samples.git`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Instal dependensi paket dan kompilasi file. TypeScript	Instal dependensi paket dan kompilasi TypeScript file dengan menjalankan perintah berikut:
	`$cd amazon-devopsguru-cdk-samples`
	`$npm install`
	`$npm fund`
	Perintah ini menginstal semua paket dari repositori sampel.
	Important
	Jika Anda mendapatkan kesalahan tentang paket yang hilang, gunakan salah satu perintah berikut:
	`$npm ci`
	— atau —
	`$npm install -g @aws-cdk/<package-name>`
	Anda dapat menemukan daftar nama dan versi paket di `Dependencies` bagian `/amazon-devopsguru-cdk-samples/package.json` file. Untuk
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi selengkapnya, lihat npm ci dan npm install di dokumentasi npm.	

Membangun (mensintesis) tumpukan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan alamat email untuk notifikasi Amazon SNS.	Ikuti langkah-langkah berikut untuk memberikan alamat email untuk notifikasi Amazon SNS: 1. Edit file `/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-multi-acc-reg-stack.ts` dan `/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-org-uni-stack.ts` . 2. Di `Subscription` bagian `DevOpsGuruTopic` , perbarui `Endpoint` parameter dengan alamat email Anda. 3. Simpan dan tutup file.	DevOps insinyur
Membangun kode proyek.	Buat kode proyek dan sintesis tumpukan dengan menjalankan perintah:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`npm run build && cdk synth`
	
	Anda akan melihat output yang serupa dengan yang berikut:
	
	`$npm run build && cdk synth`
	`> cdk-devopsguru@0.1.0 build`
	`> tsc`
	`Successfully synthesized to ~/amazon-devopsguru-cdk-samples/cdk.out`
	`Supply a stack id (CdkDevopsGuruStackMultiAccReg,CdkDevopsGuruStackMultiAccRegSpecStacks, CdkDevopsguruStackOrgUnit, CdkInfrastructureStack, CdkStackSetAdminRole, CdkStackSetExecRole) to display its template.`
	
	Untuk informasi dan langkah selengkapnya, lihat aplikasi AWS CDK pertama Anda di dokumentasi AWS CDK.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat daftar tumpukan AWS CDK.	Jalankan perintah berikut untuk mencantumkan semua tumpukan AWS CDK:
	`$cdk list`
	Perintah menampilkan daftar berikut:
	`CdkDevopsGuruStack MultiAccReg CdkDevopsGuruSt ackMultiAccRegSpec Stacks CdkDevopsguruStackOr gUnit CdkInfrastructureStac k CdkStackSetAdminRole CdkStackSetExecRole`
		DevOps insinyur

Opsi 1 - Aktifkan DevOps Guru untuk semua sumber daya tumpukan di beberapa akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan tumpukan AWS CDK untuk membuat peran IAM.	Pola ini menggunakan AWS CloudFormation StackSets untuk melakukan operasi tumpukan di beberapa akun. Jika Anda membuat kumpulan tumpukan pertama, Anda harus membuat peran IAM berikut untuk menyiapkan izin yang diperlukan di akun AWS Anda:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `AWSCloudFormationStackSetAdministrationRole` • `AWSCloudFormationStackSetExecutionRole` **Note** Peran harus memiliki nama-nama yang tepat ini. 1. Buat `AWSCloudFormationStackSetAdministrationRole` peran IAM di akun administrator (primer) dengan menjalankan perintah CLI berikut: `$cdk deploy CdkStackSetAdminRole --profile administrator` 2. Buat `AWSCloudFormationStackSetExecutionRole` peran IAM di semua akun target tempat Anda ingin menjalankan instance tumpukan. Untuk membuat peran ini, jalankan perintah CLI ini:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`$cdk deploy CdkStackSetExecRole \ --parameters AdministratorAccountId=<administrator-account-ID> \ --profile administrator`
	
	`$cdk deploy CdkStackSetExecRole \ --parameters AdministratorAccountId=<administrator-account-ID> \ --profile target`
	
	Untuk informasi selengkapnya, lihat Memberikan izin yang dikelola sendiri dalam dokumentasi CloudFormation AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan tumpukan AWS CDK untuk mengaktifkan DevOps Guru di beberapa akun.	`CdkDevopsGuruStack MultiAccReg` Tumpukan AWS CDK membuat kumpulan tumpukan untuk menerapkan instance tumpukan di beberapa akun dan Wilayah. Untuk menyebarkan tumpukan, jalankan perintah CLI berikut dengan parameter yang ditentukan: `$cdk deploy CdkDevops GuruStackMultiAccReg \ --profile administr ator \ --parameters AdministratorAccou ntId=<administrator- account-ID> \ --parameters TargetAccountId=<t arget-account-ID> \ --parameters RegionIds="<region -1>,<region-2>"` Saat ini Amazon DevOps Guru tersedia di Wilayah AWS yang tercantum dalam FAQ DevOps Guru.	DevOps insinyur

Opsi 2 - Aktifkan DevOps Guru untuk semua sumber daya tumpukan OUs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ekstrak OU IDs.	Di konsol AWS Organizations, identifikasi unit organisasi tempat Anda ingin mengaktif kan DevOps Guru. IDs	DevOps insinyur
Aktifkan izin yang dikelola layanan untuk. OUs	Jika Anda menggunakan AWS Organizations untuk manajemen akun, Anda harus memberikan izin yang dikelola layanan untuk mengaktif kan Guru. DevOps Alih-alih membuat peran IAM secara manual, gunakan akses tepercaya berbasis organisas i dan peran terkait layanan (). SLRs	DevOps insinyur
Terapkan tumpukan AWS CDK untuk mengaktifkan DevOps Guru di seluruh. OUs	`CdkDevopsguruStack OrgUnit` Tumpukan AWS CDK memungkinkan layanan DevOps Guru di seluruh OUs. Untuk menyebarkan tumpukan, jalankan perintah berikut dengan parameter yang ditentukan: `$cdk deploy CdkDevops guruStackOrgUnit \ --profile administr ator \ --parameters RegionIds="<region -1>,<region-2>" \`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`--parameters OrganizationalUnitIds="<OU-1>,<OU-2>"`	

Opsi 3 - Aktifkan DevOps Guru untuk sumber daya tumpukan tertentu di beberapa akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan tumpukan AWS CDK untuk membuat peran IAM.	Jika Anda belum membuat peran IAM yang diperlukan yang ditampilkan di opsi pertama, lakukan itu terlebih dahulu: 1. Buat `AWSCloudFormationStackSetAdministrationRole` peran IAM di akun administrator (primer) dengan menjalankan perintah CLI berikut: `$cdk deploy CdkStackSetAdminRole --profile administrator` 2. Buat `AWSCloudFormationStackSetExecutionRole` peran IAM di semua akun target tempat Anda ingin menjalankan instance tumpukan. Untuk membuat peran ini, jalankan perintah CLI:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`$cdk deploy CdkStackSetExecRole \ --parameters AdministratorAccountId=<administrator-account-ID> \ --profile administrator`
	
	`$cdk deploy CdkStackSetExecRole \ --parameters AdministratorAccountId=<administrator-account-ID> \ --profile target`
	
	Untuk informasi selengkapnya, lihat Memberikan izin yang dikelola sendiri dalam dokumentasi CloudFormation AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus tumpukan yang ada.	Jika Anda sudah menggunakan opsi pertama untuk mengaktifkan DevOps Guru untuk semua sumber daya tumpukan, Anda dapat menghapus tumpukan lama dengan menggunakan perintah berikut: `$cdk destroy CdkDevopsGuruStackMultiAccReg --profile administrator` Atau, Anda dapat mengubah `RegionIds` parameter saat menerapkan ulang tumpukan untuk menghindari kesalahan Stacks already exist.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui tumpukan AWS CDK dengan daftar tumpukan.	1. Edit file `/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-multi-acc-reg-spec-stack.ts` 2. Di bawah `Resources CloudFormation ,StackNames ,,` daftar tumpukan yang ingin Anda aktifkan DevOps Guru. Untuk tujuan demonstrasi, parameter menentukan `CdkInfrastructureStack` tumpukan, tetapi Anda dapat mengedit entri ini berdasarkan kebutuhan Anda. 3. Simpan dan tutup file . 4. Untuk mensintesis dan memperbarui template tumpukan, jalankan: `$cdk synth`	Insinyur data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan tumpukan AWS CDK untuk mengaktifkan DevOps Guru untuk sumber daya tumpukan tertentu di beberapa akun.	`CdkDevopsGuruStack MultiAccRegSpecSta cks` Tumpukan AWS CDK memungkinkan DevOps Guru untuk sumber daya tumpukan tertentu di beberapa akun. Untuk menyebarkan tumpukan, jalankan perintah berikut: `$cdk deploy CdkDevops GuruStackMultiAccR egSpecStacks \ --profile administr ator \ --parameters AdministratorAccou ntId=<administrator- account-ID> \ --parameters TargetAccountId=<t arget-account-ID> \ --parameters RegionIds="<region -1>,<region-2>"` **Note** Jika sebelumnya Anda menerapka n tumpukan ini untuk opsi 1, ubah `RegionIds` parameter (pastikan untuk memilih dari Wilayah yang tersedia) untuk menghindari	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kesalahan Stacks sudah ada.	

Menerapkan tumpukan infrastruktur AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan contoh tumpukan infrastruktur tanpa server.	`CdkInfrastructureStack` Tumpukan AWS CDK menerapkan komponen tanpa server seperti API Gateway, Lambda, dan tabel DynamoDB untuk mendemonstrasikan wawasan Guru. DevOps Untuk menyebarkan tumpukan, jalankan perintah berikut: `$cdk deploy CdkInfrastructureStack --profile administrator`	DevOps insinyur
Masukkan catatan sampel di DynamoDB.	Jalankan perintah berikut untuk mengisi tabel DynamoDB dengan catatan sampel. Berikan jalur yang benar untuk `populate-shops-dynamodb-table.json` skrip. `$aws dynamodb batch-write-item \ --request-items file://scripts/pop`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`ulate-shops-dynamodb-table.json \`
	`  --profile administrator`
	
	Perintah menampilkan output berikut:
	
	`{`
	`  "UnprocessedItems": {}`
	`}`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi catatan yang dimasukkan di DynamoDB.	Untuk memverifikasi bahwa tabel DynamoDB menyertakan catatan sampel dari `populate-shops-dynamodb-table.json` file, akses URL untuk `ListRestApiEndpointMonitorOperator` API, yang diterbitkan sebagai output dari tumpukan AWS CDK. Anda juga dapat menemukan URL ini di tab Output CloudFormation konsol AWS untuk `CdkInfrastructureStack` tumpukan. Output AWS CDK akan terlihat mirip dengan berikut ini:
	
	`CdkInfrastructureStack.CreateRestApiMonitorOperatorEndpointD1D00045 = https://oure17c5vob.execute-api.<your-region>.amazonaws.com/prod/`
	
	`CdkInfrastructureStack.ListRestApiMonitorOperatorEndpointABBDB8D8 = https://cdff8icfrn4.execute-api.<your-region>.amazonaws.com/prod/`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tunggu sumber daya untuk menyelesaikan baselining.	Tumpukan tanpa server ini memiliki beberapa sumber daya. Kami menyarankan Anda menunggu selama 2 jam sebelum Anda melakukan langkah selanjutnya. Jika Anda menerapkan tumpukan ini di lingkungan produksi, mungkin diperlukan waktu hingga 24 jam untuk menyelesaikan baselining, tergantung pada jumlah sumber daya yang Anda pilih untuk dipantau di Guru. DevOps	DevOps insinyur

Hasilkan wawasan DevOps Guru

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui tumpukan infrastruktur AWS CDK.	Untuk mencoba wawasan DevOps Guru, Anda dapat membuat beberapa perubahan konfigurasi untuk mereproduksi masalah operasional yang khas. 1. Edit file `/amazon-devopsguru-cdk-samples/lib/infrastructure-stack.ts` 2. Pada `DDB Table` bagian ini, ubah kapasitas baca	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	untuk tabel DynamoDB dari 5 menjadi 1.
	3. Simpan dan tutup file .
	4. Jalankan perintah berikut untuk mensintesis dan menerapkan tumpukan infrastruktur AWS CDK yang diperbarui:
	`$cdk synth`
	`$cdk deploy CdkInfrastructureStack --profile administrator`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyuntikkan permintaan HTTP pada API.	Menyuntikkan lalu lintas masuk dalam bentuk permintaan HTTP pada API: `ListRestApiMonitorOperatorEndpointxxxx` 1. Edit skrip Python. `/amazon-devopsguru-cdk-samples/scripts/sendAPIRequest.py` 2. Perbarui `url` variabel dengan tautan API untuk`ListRestApiMonitorOperatorEndpointxxxx` . Anda dapat menemukan URL ini di output perintah penerapan AWS CDK atau di konsol AWS CloudFormation, di tab Output untuk tumpukan. 3. Simpan dan tutup file . 4. Jalankan skrip Python dengan menggunakan perintah: `$python sendAPIRequest.py` 5. Pastikan Anda mendapatkan kode status 200. 6. Anda mungkin perlu menjalankan skrip melalui	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	beberapa (lebih disukai empat) terminal untuk menyuntikkan lalu lintas dengan kecepatan tinggi. 7. Setelah skrip berjalan sekitar 10 menit dalam satu lingkaran, Anda dapat melihat wawasan operasional di konsol DevOps Guru.	
Tinjau wawasan DevOps Guru.	Dalam kondisi standar, dasbor DevOps Guru menampilkan nol di penghitung wawasan yang sedang berlangsung. Jika mendeteksi anomali, itu menimbulkan peringatan dalam bentuk wawasan. Di panel navigasi, pilih Wawasan untuk melihat detail anomali, termasuk ikhtisar, metrik agregat, peristiwa yang relevan, dan rekomendasi. Untuk informasi selengkapnya tentang meninjau wawasan, lihat Mendapatkan wawasan operasional dengan menggunakan posting blog AIOps Amazon DevOps Guru.	DevOps insinyur

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Bersihkan dan hapus sumber daya.	Setelah Anda menelusuri pola ini, Anda harus menghapus sumber daya yang Anda buat untuk menghindari biaya lebih lanjut. Jalankan perintah ini:
	`$cdk destroy CdkDevopsGuruStackMultiAccRegprofile administrator`
	`$cdk destroy CdkDevopsguruStackOrgUnit --profile administrator`
	`$cdk destroy CdkDevopsGuruStackMultiAccRegSpecStacks --profile administrator`
	`$cdk destroy CdkInfrastructureStack --profile administrator`
	`$cdk destroy CdkStackSetAdminRole --profile administrator`
	`$cdk destroy CdkStackSetExecRole --profile administrator`
	`$cdk destroy CdkStackSetExecRole --profile target`
		DevOps insinyur

Sumber daya terkait

- Mendapatkan wawasan operasional dengan AIOps menggunakan Amazon Guru DevOps
- Konfigurasikan Amazon DevOps Guru dengan mudah di beberapa akun dan Wilayah menggunakan AWS CloudFormation StackSets

- DevOps Lokakarya Guru

Mengatur set izin untuk beberapa akun dengan menggunakan Account Factory untuk Terraform

Anand Krishna Varanasi dan Siamak Heshmati, Amazon Web Services

Ringkasan

Pola ini membantu Anda mengintegrasikan AWS Control Tower Account Factory Terraform (AFT) dengan untuk mengonfigurasi izin untuk beberapa AWS IAM Identity Center Akun AWS pada skala besar. Pendekatan ini menggunakan AWS Lambda fungsi kustom untuk mengotomatiskan penetapan set izin ke Akun AWS yang dikelola sebagai organisasi. Ini merampingkan proses karena tidak memerlukan intervensi manual dari tim teknik platform Anda. Solusi ini dapat meningkatkan efisiensi operasional, keamanan dan konsistensi. Ini mempromosikan proses orientasi yang aman dan terstandarisasi AWS Control Tower, sehingga sangat diperlukan bagi perusahaan yang memprioritaskan kelincahan dan keandalan untuk infrastruktur cloud mereka.

Prasyarat dan batasan

Prasyarat

- Akun AWS, dikelola melalui AWS Control Tower. Untuk informasi selengkapnya, lihat Memulai dengan AWS Control Tower.
- Account Factory untuk Terraform, digunakan di akun khusus di lingkungan Anda. Untuk informasi selengkapnya, lihat Menerapkan AWS Control Tower Account Factory untuk Terraform.
- Instance IAM Identity Center, disiapkan di lingkungan Anda. Untuk informasi selengkapnya, lihat Memulai Pusat Identitas IAM.
- Grup Pusat Identitas IAM aktif, dikonfigurasi. Untuk informasi selengkapnya, lihat Menambahkan grup ke grup Direktori Pusat Identitas IAM.
- Python versi 3.9 atau yang lebih baru, diinstal

Batasan

- Solusi ini hanya dapat digunakan dengan akun yang dikelola melalui AWS Control Tower. Solusi ini digunakan dengan menggunakan Account Factory untuk Terraform.
- Pola ini tidak termasuk instruksi untuk mengatur federasi identitas dengan sumber identitas. Untuk informasi selengkapnya tentang cara menyelesaikan pengaturan ini, lihat tutorial sumber identitas Pusat Identitas IAM di dokumentasi Pusat Identitas IAM.

Arsitektur

Ikhtisar AFT

AFT menyiapkan pipeline Terraform yang membantu Anda menyediakan dan menyesuaikan akun Anda. AWS Control Tower AFT mengikuti GitOps model yang mengotomatiskan proses penyediaan akun di. AWS Control Tower Anda membuat file Terraform permintaan akun dan memasukkannya ke repositori. Ini memulai alur kerja AFT untuk penyediaan akun. Setelah penyediaan akun selesai, AFT dapat secara otomatis menjalankan langkah-langkah penyesuaian tambahan. Untuk informasi selengkapnya, lihat arsitektur AFT dalam AWS Control Tower dokumentasi.

AFT menyediakan repositori utama berikut:

- `aft-account-request`— Repositori ini berisi kode Terraform untuk membuat atau memperbarui. Akun AWS
- `aft-account-customizations`— Repositori ini berisi kode Terraform untuk membuat atau menyesuaikan sumber daya berdasarkan per akun.
- `aft-global-customizations`— Repositori ini berisi kode Terraform untuk membuat atau menyesuaikan sumber daya untuk semua akun, dalam skala besar.
- `aft-account-provisioning-customizations`— Repositori ini mengelola kustomisasi yang diterapkan hanya untuk akun tertentu yang dibuat oleh dan dikelola dengan AFT. Misalnya, Anda dapat menggunakan repositori ini untuk menyesuaikan penetapan pengguna atau grup di Pusat Identitas IAM atau untuk mengotomatiskan penutupan akun.

Ikhtisar solusi

Solusi khusus ini mencakup mesin AWS Step Functions status dan AWS Lambda fungsi yang menetapkan set izin untuk pengguna dan grup untuk beberapa akun. Mesin status yang digunakan melalui pola ini beroperasi bersama dengan mesin status AFT `aft_account_provisioning_customizations` yang sudah ada sebelumnya. Pengguna mengirimkan permintaan untuk memperbarui tugas pengguna dan grup IAM Identity Center baik saat baru Akun AWS dibuat atau setelah akun dibuat. Mereka melakukan ini dengan mendorong perubahan ke `aft-account-request` repositori. Permintaan untuk membuat atau memperbarui akun memulai aliran di Amazon DynamoDB Streams. Ini memulai fungsi Lambda, yang memperbarui pengguna dan grup Pusat Identitas IAM untuk target. Akun AWS

Berikut ini adalah contoh parameter yang dapat Anda berikan dalam fungsi Lambda untuk penetapan set izin untuk menargetkan pengguna dan grup:

```
custom_fields = {
    "InstanceArn"         = "<Organization ID>",
    "PermissionSetArn"    = "<Permission set ARN>",
    "PrincipalId"         = "<Principal ID>",
  }
```

Berikut ini adalah parameter dalam pernyataan ini:

- `InstanceArn`— Nama Sumber Daya Amazon (ARN) organisasi
- `PermissionSetArn`— ARN dari set izin
- `PrincipalId`— Pengidentifikasi pengguna atau grup di Pusat Identitas IAM tempat set izin akan diterapkan

> **Note**
>
> Anda harus membuat set izin target, pengguna, dan grup sebelum menjalankan solusi ini.

Meskipun `InstanceArn` nilainya harus tetap konsisten, Anda dapat memodifikasi fungsi Lambda untuk menetapkan beberapa set izin ke beberapa identitas target. Parameter untuk set izin harus diakhiri`PermissionSetArn`, dan parameter untuk pengguna dan grup harus diakhiri`PrincipalId`. Anda harus mendefinisikan kedua atribut. Berikut ini adalah contoh cara mendefinisikan beberapa set izin dan target pengguna dan grup:

```
custom_fields = {
    "InstanceArn"                    = "<Organization ID>",
    "AdminAccessPermissionSetArn"    = "<Admin privileges permission set ARN>",
    "AdminAccessPrincipalId"         = "<Admin principal ID>",
    "ReadOnlyAccessPermissionSetArn" = "<Read-only privileges permission set ARN>",
    "ReadOnlyAccessPrincipalId"      = "<Read-only principal ID>",
  }
```

Diagram berikut menunjukkan step-by-step alur kerja tentang cara solusi memperbarui set izin untuk pengguna dan grup dalam target Akun AWS pada skala besar. Saat pengguna memulai permintaan pembuatan akun, AFT memulai mesin status `aft-account-provisioning-framework` Step Functions. Mesin status ini memulai fungsi `extract-alternate-sso` Lambda. Fungsi Lambda menetapkan set izin untuk pengguna dan grup dalam target. Akun AWS Pengguna atau grup ini

dapat berasal dari sumber identitas yang dikonfigurasi di IAM Identity Center. Contoh sumber identitas termasuk Okta, Active Directory, atau Ping Identity.

Diagram menunjukkan alur kerja berikut saat akun baru dibuat:

1. Seorang pengguna mendorong `custom_fields` perubahan ke `aft-account-request` repositori.
2. AWS CodePipeline memulai AWS CodeBuild pekerjaan yang merekam metadata yang ditentukan pengguna ke dalam tabel Amazon `aft-request-audit` DynamoDB. Tabel ini memiliki atribut untuk merekam metadata yang ditentukan pengguna. `ddb_event_name`Atribut mendefinisikan jenis operasi AFT:
   - Jika nilainya`INSERT`, maka solusi akan menetapkan izin yang disetel ke identitas target saat yang baru Akun AWS dibuat.
   - Jika nilainya`UPDATE`, maka solusi akan menetapkan izin yang disetel ke identitas target setelah dibuat. Akun AWS
3. Amazon DynamoDB Streams memulai `aft_alternate_sso_extract` fungsi Lambda.
4. Fungsi `aft_alternate_sso_extract` Lambda mengasumsikan peran AWS Identity and Access Management (IAM) dalam akun manajemen. AWS Control Tower
5. Fungsi Lambda menetapkan set izin ke pengguna dan grup target dengan membuat panggilan API AWS SDK for Python (Boto3) create_account_assignment ke IAM Identity Center. Ini mengambil set izin dan penetapan identitas dari tabel Amazon `aft-request-audit` DynamoDB.
6. Ketika alur kerja Step Functions selesai, set izin ditetapkan ke identitas target.

Otomatisasi dan skala

AFT beroperasi pada skala dengan menggunakan Layanan AWS seperti CodePipeline,, DynamoDB AWS CodeBuild, dan Lambda, yang sangat skalabel. Untuk otomatisasi tambahan, Anda dapat mengintegrasikan solusi ini dengan tiket atau sistem manajemen masalah, seperti Jira. Untuk informasi selengkapnya, lihat bagian Informasi tambahan dari pola ini.

Alat

Layanan AWS

- Account Factory for Terraform (AFT) adalah alat utama dalam solusi ini. `aft-account-provisioning-customizations`Repositori berisi kode Terraform untuk membuat penyesuaian, seperti penugasan pengguna atau grup Pusat Akun AWS Identitas IAM khusus.
- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.

Alat-alat lainnya

- Python adalah bahasa pemrograman komputer tujuan umum.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Repositori kode untuk AFT tersedia di GitHub AWS Control Tower Account Factory for Terraform repositori. Kode untuk pola ini tersedia di Govern SSO Assignments untuk menggunakan Account Akun AWS Factory for Terraform (AFT) repositori.

Praktik terbaik

- Memahami model tanggung jawab AWS bersama.
- Ikuti rekomendasi keamanan untuk AWS Control Tower. Untuk informasi selengkapnya, lihat Keamanan di AWS Control Tower.
- Ikuti prinsip hak istimewa yang paling sedikit. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa paling sedikit.
- Membangun set izin khusus dan terfokus dan peran IAM untuk grup dan unit bisnis.

Epik

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat peran IAM.	Di akun AWS Control Tower manajemen, gunakan Terraform untuk membuat peran IAM. Peran ini memiliki akses lintas akun dan kebijakan kepercayaan yang memungkinkan akses federasi dari penyedia identitas. Ini juga memiliki izin untuk memberikan akses ke akun lain melalui AWS Control Tower. Fungsi Lambda akan mengambil peran ini. Lakukan hal-hal berikut:
	1. Unduh `AFTCrossAccountRole.tf` file dari repositori GitHub kode.
	2. Ubah `AFTCrossAccountRole.tf` file seperlunya untuk AWS lingkungan Anda.
	3. Di Terraform, masukkan perintah berikut untuk membuat peran IAM ini:
	`terraform init`
	`terraform plan`
	`terraform apply`
	4. Validasi bahwa peran berhasil diterapkan dan
		AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memiliki akses lintas akun yang diharapkan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan solusi untuk lingkungan Anda.	1. Masukkan perintah berikut untuk mengkloning Govern SSO Assignments untuk menggunakan Account Akun AWS Factory for Terraform (AFT) repositori ke workstation lokal Anda. `git clone https://github.com/aws-samples/aft-custom-sso-assignment.git` 2. Di `aft-account-provisioning-customizations/terraform` folder, buka `variables.tf` file. 3. Ubah variabel sesuai kebutuhan untuk lingkungan Anda. 4. Simpan dan tutup file `variables.tf` . 5. Buka `account-request.tf` file di `aft-account-request` repositori Anda. 6. Ubah `custom_fields` parameter untuk menentukan set izin dan target pengguna dan grup. Untuk informasi lebih lanjut, lihat bagian Arsitektur dari pola ini.	AWS DevOps, arsitek Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Simpan dan tutup file `account-request.tf` .	
Menyebarkan solusinya.	1. Di repositori kloning, salin isi `terraform` folder, lalu tempel ke `terraform` folder di repositori. `aft-account-provisioning-customizations` 2. Di akun manajemen AFT, mulai `ct-aft-account-provisioning-customizations` pipa. Ini menyebarkan solusi khusus. Untuk petunjuk, lihat Memulai pipeline di CodePipeline. 3. Validasi bahwa sumber daya berhasil digunakan di akun manajemen AFT.	AWS DevOps, arsitek Cloud
Siapkan koneksi repositori kode.	Siapkan koneksi antara repositori kode tempat Anda akan menyimpan file konfigurasi dan file Anda. Akun AWS Untuk petunjuk, lihat Tambahkan penyedia sumber pihak ketiga ke saluran pipa yang digunakan CodeConnections dalam AWS CodePipeline dokumentasi.	AWS DevOps, arsitek Cloud

Gunakan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai pipeline AFT untuk menerapkan akun baru.	Ikuti petunjuk di Menyediakan akun baru dengan AFT untuk memulai pipeline yang membuat yang baru Akun AWS di AWS Control Tower lingkungan Anda. Tunggu proses pembuatan akun selesai.	AWS DevOps, arsitek Cloud
Validasi perubahan.	1. Buka konsol AWS IAM Identity Center. 2. Dalam daftar akun, pilih akun yang baru dibuat. 3. Validasi bahwa set izin telah ditetapkan untuk memberikan akses ke pengguna dan grup target.	AWS DevOps, arsitek Cloud

Pemecahan masalah

Isu	Solusi
Penugasan set izin tidak berfungsi.	Pastikan parameter ARN grup, id organisasi, dan Lambda sudah benar. Sebagai contoh, lihat bagian Ikhtisar solusi dari pola ini.
Memperbarui kode di repositori tidak memulai pipa.	Masalah ini terkait dengan konektivitas antara Anda Akun AWS dan repositori. Dalam Konsol Manajemen AWS, validasi bahwa koneksi aktif. Untuk informasi selengkapnya, lihat GitHub koneksi dalam AWS CodePipeline dokumentasi.

Informasi tambahan

Mengintegrasikan dengan alat manajemen tiket

Anda dapat memilih untuk mengintegrasikan solusi ini dengan alat manajemen tiket atau masalah, seperti Jira atau ServiceNow. Diagram berikut menunjukkan contoh alur kerja untuk opsi ini. Anda dapat mengintegrasikan alat manajemen tiket dengan repositori solusi AFT dengan menggunakan konektor alat Anda. Untuk konektor Jira, lihat Mengintegrasikan Jira dengan. GitHub Untuk ServiceNow konektor, lihat Mengintegrasikan dengan GitHub. Anda bahkan dapat membuat solusi khusus yang mengharuskan pengguna memberikan ID tiket sebagai bagian dari persetujuan permintaan tarik. Jika permintaan untuk membuat baru Akun AWS dengan menggunakan AFT disetujui, peristiwa tersebut dapat memulai alur kerja yang menambahkan bidang khusus ke repositori. `aft-account-request` GitHub Anda dapat mendesain alur kerja khusus apa pun yang memenuhi persyaratan kasus penggunaan Anda.

Diagram menunjukkan alur kerja berikut:

1. Pengguna meminta penetapan set izin khusus di alat manajemen tiket, seperti Jira.
2. Setelah kasus disetujui, alur kerja mulai memperbarui penetapan set izin. (Opsional) Anda dapat menggunakan plugin untuk otomatisasi kustom langkah ini.
3. Operator mengirim kode Terraform dengan parameter set izin yang diperbarui ke `aft-account-request` repositori ke cabang pengembangan atau fitur.
4. GitHub Tindakan dimulai AWS CodeBuild dengan menggunakan panggilan OpenID Connect (OIDC). CodeBuild melakukan pemindaian keamanan infrastruktur sebagai kode (IAc) dengan menggunakan alat seperti tfsec dan checkov. Ini memperingatkan operator dari setiap pelanggaran keamanan.
5. Jika tidak ditemukan pelanggaran, GitHub Actions akan membuat permintaan tarik otomatis dan menetapkan tinjauan kode kepada pemilik kode. Ini juga membuat tag untuk permintaan tarik.
6. Jika pemilik kode menyetujui peninjauan kode, alur kerja GitHub Tindakan lain akan dimulai. Ini memeriksa standar permintaan tarik, termasuk:
   - Jika judul permintaan tarik memenuhi persyaratan.
   - Jika badan permintaan tarik berisi nomor kasus yang disetujui.
   - Jika permintaan tarik ditandai dengan benar.
7. Jika permintaan tarik memenuhi standar, GitHub Actions akan memulai alur kerja produk AFT. Ini menggunakan memulai `ct-aft-account-request` pipa di AWS CodePipeline. Pipeline ini

memulai mesin status `aft-account-provisioning-framework` kustom di Step Functions. Mesin status ini berfungsi seperti yang dijelaskan sebelumnya di bagian Ikhtisar solusi dari pola ini.

Implementasikan Account Factory for Terraform (AFT) dengan menggunakan pipeline bootstrap

Vinicius Elias dan Edgar Costa Filho, Amazon Web Services

Ringkasan

Pola ini menyediakan metode sederhana dan aman untuk menerapkan AWS Control Tower Account Factory for Terraform (AFT) dari akun manajemen. AWS Organizations Inti dari solusi ini adalah CloudFormation template yang mengotomatiskan konfigurasi AFT dengan membuat pipeline Terraform, yang disusun agar mudah beradaptasi untuk penerapan awal atau pembaruan berikutnya.

Keamanan dan integritas data adalah prioritas utama di AWS, sehingga file status Terraform, yang merupakan komponen penting yang melacak status infrastruktur dan konfigurasi terkelola, disimpan dengan aman di bucket Amazon Simple Storage Service (Amazon S3). Bucket ini dikonfigurasi dengan beberapa langkah keamanan, termasuk enkripsi sisi server dan kebijakan untuk memblokir akses publik, untuk membantu memastikan bahwa status Terraform Anda terlindungi dari akses yang tidak sah dan pelanggaran data.

Akun manajemen mengatur dan mengawasi seluruh lingkungan, sehingga merupakan sumber daya penting dalam. AWS Control Tower Pola ini mengikuti praktik AWS terbaik dan memastikan bahwa proses penyebaran tidak hanya efisien tetapi juga selaras dengan standar keamanan dan tata kelola, untuk menawarkan cara yang komprehensif, aman, dan efisien untuk menyebarkan AFT di lingkungan Anda. AWS

Untuk informasi lebih lanjut tentang AFT, lihat AWS Control Tower dokumentasi.

Prasyarat dan batasan

Prasyarat

- Lingkungan AWS multi-akun dasar dengan akun berikut minimal: akun manajemen, akun Arsip Log, akun Audit, dan satu akun tambahan untuk manajemen AFT.
- AWS Control Tower Lingkungan yang mapan. Akun manajemen harus dikonfigurasi dengan benar, karena CloudFormation template akan digunakan di dalamnya.
- Izin yang diperlukan dalam akun AWS manajemen. Anda akan memerlukan izin yang cukup untuk membuat dan mengelola sumber daya seperti bucket S3, AWS Lambda fungsi, peran AWS Identity and Access Management (IAM), dan proyek. AWS CodePipeline

- Keakraban dengan Terraform. Memahami konsep inti dan alur kerja Terraform adalah penting karena penerapan melibatkan pembuatan dan pengelolaan konfigurasi Terraform.

Batasan

- Waspadai kuota AWS sumber daya di akun Anda. Penerapan dapat membuat beberapa sumber daya, dan menghadapi kuota layanan dapat menghambat proses penerapan.
- Template ini dirancang untuk versi tertentu dari Terraform dan. Layanan AWS Memutakhirkan atau mengubah versi mungkin memerlukan modifikasi templat.
- Template tidak mendukung layanan sistem kontrol versi (VCS) yang dikelola sendiri seperti GitHub Enterprise.

Versi produk

- Terraform versi 1.6.6 atau yang lebih baru
- AFT versi 1.11 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

- CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon EventBridge
- IAM
- AWS Lambda
- Amazon S3

Arsitektur target

Diagram berikut menggambarkan implementasi yang dibahas dalam pola ini.

Alur kerja terdiri dari tiga tugas utama: membuat sumber daya, menghasilkan konten, dan menjalankan pipeline.

Menciptakan sumber daya

CloudFormation Template yang disediakan dengan pola ini membuat dan mengatur semua sumber daya yang diperlukan, tergantung pada parameter yang Anda pilih saat Anda menerapkan template. Minimal, template membuat sumber daya berikut:

- CodePipeline Pipa untuk mengimplementasikan AFT
- Bucket S3 untuk menyimpan file status Terraform yang terkait dengan implementasi AFT
- Dua CodeBuild proyek untuk mengimplementasikan rencana Terraform dan menerapkan perintah di berbagai tahap pipa
- Peran CodeBuild dan CodePipeline layanan IAM
- Bucket S3 kedua untuk menyimpan artefak runtime pipeline

Bergantung pada penyedia VCS yang Anda pilih (CodeCommit atau VCS eksternal), template membuat sumber daya berikut.

- Untuk CodeCommit:
  - CodeCommit Repositori untuk menyimpan kode bootstrap AFT Terraform
  - EventBridge Aturan untuk menangkap perubahan CodeCommit repositori di cabang `main`
  - Peran IAM lain untuk aturan EventBridge
- Untuk penyedia VCS eksternal lainnya, seperti GitHub:
  - AWS CodeConnections Koneksi

Selain itu, saat Anda memilih CodeCommit sebagai penyedia VCS, jika Anda menyetel `Generate AFT Files` parameternya`true`, template akan membuat sumber daya tambahan ini untuk menghasilkan konten:

- Bucket S3 untuk menyimpan konten yang dihasilkan dan digunakan sebagai sumber repositori CodeCommit
- Fungsi Lambda untuk memproses parameter yang diberikan dan menghasilkan konten yang sesuai
- Fungsi IAM untuk menjalankan fungsi Lambda

- Sumber daya CloudFormation kustom yang menjalankan fungsi Lambda saat template di-deploy

Menghasilkan konten

Untuk menghasilkan file bootstrap AFT dan kontennya, solusinya menggunakan fungsi Lambda dan bucket S3. Fungsi membuat folder di bucket, dan kemudian membuat dua file di dalam folder: `main.tf` dan`backend.tf`. Fungsi ini juga memproses CloudFormation parameter yang disediakan dan mengisi file-file ini dengan kode yang telah ditentukan, menggantikan nilai parameter masing-masing.

Untuk melihat kode yang digunakan sebagai template untuk menghasilkan file, lihat GitHub repositori solusi. Pada dasarnya, file dihasilkan sebagai berikut.

utama.tf

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory?
ref=<aft_version>"

  # Required variables
  ct_management_account_id  = "<ct_management_account_id>"
  log_archive_account_id    = "<log_archive_account_id>"
  audit_account_id          = "<audit_account_id>"
  aft_management_account_id = "<aft_management_account_id>"
  ct_home_region            = "<ct_home_region>"

  # Optional variables
  tf_backend_secondary_region = "<tf_backend_secondary_region>"
  aft_metrics_reporting       = "<false|true>"

  # AFT Feature flags
  aft_feature_cloudtrail_data_events      = "<false|true>"
  aft_feature_enterprise_support          = "<false|true>"
  aft_feature_delete_default_vpcs_enabled = "<false|true>"

  # Terraform variables
  terraform_version      = "<terraform_version>"
  terraform_distribution = "<terraform_distribution>"

  # VCS variables (if you have chosen an external VCS)
  vcs_provider                                = "<github|githubenterprise|gitlab|
gitlabselfmanaged|bitbucket>"
```

```
  account_request_repo_name                       = "<org-name>/aft-account-request"
  account_customizations_repo_name                = "<org-name>/aft-account-
customizations"
  account_provisioning_customizations_repo_name = "<org-name>/aft-account-provisioning-
customizations"
  global_customizations_repo_name                 = "<org-name>/aft-global-
customizations"

}
```

backend.tf

```
terraform {
  backend "s3" {
    region = "<aft-main-region>"
    bucket = "<s3-bucket-name>"
    key    = "aft-setup.tfstate"
  }
}
```

Selama pembuatan CodeCommit repositori, jika Anda menyetel `Generate AFT Files` parameternya`true`, template menggunakan bucket S3 dengan konten yang dihasilkan sebagai sumber `main` cabang untuk mengisi repositori secara otomatis.

Menjalankan pipa

Setelah sumber daya dibuat dan file bootstrap telah dikonfigurasi, pipeline berjalan. Tahap pertama (Sumber) mengambil kode sumber dari cabang utama repositori, dan tahap kedua (Build) menjalankan perintah Terraform plan dan menghasilkan hasil yang akan ditinjau. Pada tahap ketiga (Persetujuan), pipeline menunggu tindakan manual untuk menyetujui atau menolak tahap terakhir (Deploy). Pada tahap terakhir, pipeline menjalankan perintah Terraform dengan menggunakan hasil `apply` perintah Terraform `plan` sebelumnya sebagai input. Terakhir, peran lintas akun dan izin di akun manajemen digunakan untuk membuat sumber daya AFT di akun manajemen AFT.

> **Note**
>
> Jika Anda memilih penyedia VCS eksternal, Anda harus mengotorisasi koneksi dengan kredenal penyedia VCS Anda. Untuk menyelesaikan penyiapan, ikuti langkah-langkah di Perbarui koneksi yang tertunda di dokumentasi konsol Alat AWS Pengembang.

Alat

Layanan AWS

- CloudFormationmembantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS CodeBuildadalah layanan build yang dikelola sepenuhnya yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS CodePipelinemembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS CodeConnectionsmemungkinkan AWS sumber daya dan layanan, seperti CodePipeline, untuk terhubung ke repositori kode eksternal, seperti. GitHub
- AWS Lambdaadalah layanan komputasi yang menjalankan kode Anda sebagai respons terhadap peristiwa dan secara otomatis mengelola sumber daya komputasi, menyediakan cara cepat untuk membuat aplikasi modern tanpa server untuk produksi.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS

Alat-alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) yang memungkinkan Anda membangun, mengubah, dan membuat versi infrastruktur dengan aman dan efisien. Ini termasuk komponen tingkat rendah seperti instance komputasi, penyimpanan, dan jaringan; dan komponen tingkat tinggi seperti entri DNS dan fitur SaaS.
- Python adalah bahasa pemrograman yang mudah dipelajari dan kuat. Ini memiliki struktur data tingkat tinggi yang efisien dan menyediakan pendekatan sederhana namun efektif untuk pemrograman berorientasi objek.

Repositori kode

Kode untuk pola ini tersedia di repositori pipeline bootstrap GitHub AFT.

Untuk repositori AFT resmi, lihat AWS Control Tower Account Factory untuk Terraform di. GitHub

Praktik terbaik

Saat Anda menerapkan AFT dengan menggunakan CloudFormation templat yang disediakan, kami sarankan Anda mengikuti praktik terbaik untuk membantu memastikan implementasi yang aman, efisien, dan berhasil. Pedoman dan rekomendasi utama untuk menerapkan dan mengoperasikan AFT meliputi yang berikut ini.

- Tinjauan parameter secara menyeluruh: Tinjau dan pahami setiap parameter dalam CloudFormation templat dengan cermat. Konfigurasi parameter yang akurat sangat penting untuk pengaturan dan fungsi AFT yang benar.
- Pembaruan template reguler: Pertahankan template diperbarui dengan AWS fitur terbaru dan versi Terraform. Pembaruan rutin membantu Anda memanfaatkan fungsionalitas baru dan menjaga keamanan.
- Pembuatan Versi: Sematkan versi modul AFT Anda dan gunakan penerapan AFT terpisah untuk pengujian jika memungkinkan.
- Cakupan: Gunakan AFT hanya untuk menyebarkan pagar pembatas infrastruktur dan penyesuaian. Jangan menggunakannya untuk menyebarkan aplikasi Anda.
- Linting dan validasi: Pipeline AFT memerlukan konfigurasi Terraform yang dilinted dan divalidasi. Jalankan lint, validasi, dan uji sebelum mendorong konfigurasi ke repositori AFT.
- Modul Terraform: Buat kode Terraform yang dapat digunakan kembali sebagai modul, dan selalu tentukan versi Terraform dan AWS penyedia agar sesuai dengan kebutuhan organisasi Anda.

Epik

Siapkan dan konfigurasikan AWS lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan AWS Control Tower lingkungan.	Siapkan dan konfigurasikan AWS Control Tower di AWS lingkungan Anda untuk memastikan manajemen dan tata kelola terpusat untuk Anda. Akun AWS Untuk informasi selengkapnya, lihat	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Memulai AWS Control Tower dalam AWS Control Tower dokumentasi.	
Luncurkan akun manajemen AFT.	Gunakan AWS Control Tower Account Factory untuk meluncurkan akun baru Akun AWS yang berfungsi sebagai akun manajemen AFT Anda. Untuk informasi selengkap nya, lihat Menyediakan AWS Service Catalog akun dengan Account Factory di AWS Control Tower dokumentasi.	Administrator awan

Menyebarkan CloudFormation template di akun manajemen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan CloudFormation template.	Dalam epik ini, Anda menerapkan CloudForm ation template yang disediaka n dengan solusi ini untuk mengatur pipeline bootstrap AFT di akun AWS manajemen Anda. Pipeline menyebark an solusi AFT di akun manajemen AFT yang Anda siapkan di epik sebelumnya. Langkah 1: Buka CloudForm ation konsol • Masuk ke Konsol Manajemen AWS dan buka	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CloudFormation konsol. Pastikan Anda beroperasi di wilayah AWS Control Tower utama yang benar. Langkah 2: Buat tumpukan baru 1. Pilih untuk membuat tumpukan baru. 2. Pilih opsi untuk mengunggah file templat, dan unggah CloudFormation templat yang disediakan dengan pola ini. Langkah 3: Konfigurasikan parameter tumpukan • `VCS Provider`: Pilih penyedia sistem kontrol versi (VCS) yang akan digunakan. Anda dapat memilih VCS eksternal seperti GitHub, atau menggunakan CodeCommit jika akun Anda diizinkan untuk menggunakan layanan. • `Repository Name` : Tentukan nama repositori untuk menyimpan modul bootstrap AFT. Untuk penyedia VCS eksternal	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	, gunakan jalur lengkap, termasuk nama organisasi (misalnya,`my-github-org/my-repo`). • `Branch Name`: Tentukan cabang repositori sumber. • `CodeBuild Docker Image`: Pilih file yang akan digunakan sebagai gambar dasar CodeBuild Docker. • Jika Anda mengatur penyedia VCS ke opsi selain CodeCommit, lanjutkan ke langkah 8. Langkah 4: Tentukan pembuatan file • Jika Anda memilih CodeCommit sebagai penyedia VCS, Anda dapat menggunakan `Generate AFT Files` parameter untuk mengontrol pembuatan file penerapan AFT default. Setel parameter ini ke: • `true`untuk secara otomatis membuat dan menyimpan file penyebaran AFT di repositori yang ditentukan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `false`jika Anda ingin menangani pembuatan file secara manual atau sudah memiliki file di tempatnya. • Jika Anda memilih`false`, lanjutkan ke langkah 8; jika tidak, ikuti langkah 5-7 terlebih dahulu. Langkah 5: Isi AWS Control Tower dan detail akun AFT • Jika Anda mengatur `Generate AFT Files` parameter k`true`, berikan informasi spesifik akun berikut AWS Control Tower dan AFT. • `Log Archive Account ID`: ID ID akun Arsip Log di AWS Control Tower. • `Audit Account ID`: ID akun Audit di AWS Control Tower. • `AFT Managemen t Account ID` : ID akun manajemen AFT yang Anda buat di epik pertama. • `AFT Main Region`dan`AFT Secondary Region`:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Utama dan sekunder Wilayah AWS untuk penyebaran AFT.
	
	Langkah 6: Konfigurasikan opsi AFT
	
	• Siapkan pelaporan metrik:
	  • `AFT Enable Metrics Reporting` : Mengaktifkan atau menonaktifkan pelaporan metrik AFT. Untuk informasi selengkapnya, lihat Metrik operasional dalam AWS Control Tower dokumentasi.
	
	• Atur opsi fitur AFT:
	  • `Enable AFT CloudTrail Data Events`: Aktifkan peristiwa CloudTrail data di semua akun terkelola AFT. Untuk informasi selengkapnya, lihat peristiwa AWS CloudTrail data dalam AWS Control Tower dokumentasi.
	  • `Enable AFT Enterprise Support` : Aktifkan Enterprise Support di semua akun terkelola AFT. Untuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi selengkapnya, lihat paket AWS Enterprise Support di AWS Control Tower dokumentasi. • `Enable AFT Delete Default VPC`: Hapus semua VPCs di akun manajemen AFT saja. Untuk informasi selengkapnya, lihat Menghapus VPC AWS default dalam dokumentasi. AWS Control Tower Langkah 7: Tentukan versi • `AFT Terraform Version`: Pilih versi Terraform untuk digunakan di saluran pipa AFT. • `AFT Version`: Tentukan versi AFT untuk penerapan. Pertahankan pengaturan default (`latest`) untuk menggunakan versi AFT terbaru. Langkah 8: Tinjau dan buat tumpukan • Tinjau semua parameter dan pengaturan. Jika semuanya beres, lanjutkan untuk membuat tumpukan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Langkah 9: Pantau pembuatan tumpukan • CloudFormation menyediakan dan mengkonfigurasi sumber daya yang Anda tentukan. Pantau proses pembuatan tumpukan di CloudFormation konsol. Proses ini mungkin memakan waktu beberapa menit. Langkah 10: Verifikasi penyebaran • Ketika status tumpukan menunjukkan CREATE_COMPLETE, verifikasi bahwa semua sumber daya telah dibuat dengan benar. • Di bagian Output, perhatikan `TerraformBackendBucketName` nilainya.	

Mengisi dan memvalidasi repositori dan pipeline bootstrap AFT

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 1: Isi repositori bootstrap AFT untuk VCS eksternal.	Jika Anda mengatur penyedia VCS ke VCS eksternal (bukan ke CodeCommit), ikuti langkah-langkah ini.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	(Opsional) Setelah Anda menerapkan CloudFormation template, Anda dapat mengisi atau memvalidasi konten di repositori bootstrap AFT yang baru dibuat, dan menguji apakah pipeline telah berhasil berjalan. Langkah 1: Perbarui koneksi 1. Di CodePipeline konsol, di panel navigasi, pilih Pengaturan, Koneksi. 2. Pilih `aft-vcs-connection` koneksi. Itu harus dalam `Pending` status. 3. Pilih Perbarui koneksi yang tertunda dan ikuti petunjuk di Perbarui koneksi yang tertunda di dokumentasi konsol Alat Pengembang. 4. Saat koneksi dalam `Available` status, pindah ke langkah berikutnya. Langkah 2: Isi repositori 1. Gunakan kredenal VCS eksternal Anda untuk mengkloning repositori yang Anda tentukan dalam template ke mesin lokal Anda. Jika Anda	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menyimpan nama default, repositori dipanggil. `aft-setup` 2. Di repositori, buat folder bernama `terraform` dengan dua file kosong di dalamnya: `backend.tf` dan. `main.tf` 3. Buka `backend.tf` file dan tambahkan potongan kode ini: `terraform { backend "s3" { region = "<aft-main-region>" bucket = "<s3-bucket-name>" key = "aft-setup" } }` Dalam file: • Ganti `<aft-main-region>` dengan Wilayah AFT utama. Ini harus sesuai dengan Wilayah AWS Control Tower utama. • Ganti `<s3-bucket-name>` dengan nama bucket backend Terraform. Anda dapat menemukan ini di	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`TerraformBackendBucketName` output yang dihasilkan oleh CloudFormation template yang Anda gunakan sebelumnya. 4. Buka `main.tf` file dan gunakan salah satu contoh yang tersedia di repositori AFT untuk menyebarkan AFT. Misalnya, Anda dapat bekerja dengan penyedia VCS pilihan Anda (CodeCommit, GitHub, atau Bitbucket) atau menyesuaikan VPC AFT. Untuk opsi input AFT lainnya, lihat file README di repositori AFT. Langkah 2: Komit dan dorong perubahan Anda • Setelah Anda membuat dan mengisi folder dan file, konfirmasikan perubahan Anda, dan unggah kode ke repositori. Pipeline dimulai secara otomatis, berjalan melalui tahapan Sumber dan Bangun, dan kemudian menunggu tindakan Persetujuan sebelum tahap Penerapan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 2: Isi repositori bootstrap AFT untuk. CodeCommit	Jika Anda mengatur penyedia VCS CodeCommit, ikuti langkah-langkah ini. (Opsional) Setelah Anda menerapkan CloudFormation template, Anda dapat mengisi atau memvalidasi konten di repositori bootstrap AFT yang baru dibuat, dan menguji apakah pipeline telah berhasil berjalan. Jika Anda menyetel `Generate AFT Files` parameternya`true`, lewati ke cerita berikutnya (memvalidasi pipeline). Langkah 1: Isi repositori 1. Buka AWS CodeCommit konsol dan pilih repositori yang baru dibuat. Jika Anda menyimpan nama default, nama repositori seharusny a. `aft-setup` 2. Kloning repositori ke mesin lokal Anda dengan menggunakan SSH, HTTPS, atau HTTPS (GRC), dan buka di editor. 3. Buat folder bernama `terraform` dan dua file kosong di	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dalamnya: `backend.tf` dan`main.tf`. 4. Buka `backend.tf` file dan tambahkan potongan kode ini: `terraform { backend "s3" { region = "<aft-main-region>" bucket = "<s3-bucket-name>" key = "aft-setup" } }` Dalam file: • Ganti `<aft-main-region>` dengan Wilayah AFT utama. Ini harus sesuai dengan Wilayah AWS Control Tower utama. • Ganti `<s3-bucket-name>` dengan nama bucket backend Terraform. Anda dapat menemukan ini di `TerraformBackendBucketName` output yang dihasilkan oleh CloudFormation template yang Anda gunakan sebelumnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Buka `main.tf` file dan gunakan salah satu contoh yang tersedia di repositori AFT untuk menyebarkan AFT. Misalnya, Anda dapat bekerja dengan penyedia sistem kontrol versi (VCS) pilihan Anda (CodeCommit, GitHub, atau Bitbucket) atau menyesuaikan VPC AFT. Untuk opsi input AFT lainnya, lihat file README di repositori AFT. Langkah 2: Komit dan dorong perubahan Anda • Setelah Anda membuat dan mengisi folder dan file, konfirmasikan perubahan Anda, dan unggah kode ke repositori. Pipeline dimulai secara otomatis, berjalan melalui tahapan Sumber dan Bangun, dan kemudian menunggu tindakan Persetujuan sebelum tahap Penerapan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi pipeline bootstrap AFT.	Langkah 1: Lihat pipa • Buka CodePipeline konsol dan periksa apakah `aft-bootstrap-pipeline` pipa berhasil dimulai. Itu harus menjalankan rencana Terraform atau menunggu tindakan persetujuan manual. Langkah 2: Setujui hasil rencana Terraform • Anda dapat meninjau hasil paket Terraform dengan melihat log eksekusi tahap Build, lalu menyetujui atau menolak eksekusi pada tahap Persetujuan. Jika Anda menyetujui, pipeline mulai menerapkan sumber daya AFT di akun manajemen AFT yang disediakan. Langkah 3: Tunggu penyebaran • Tunggu pipa berjalan dengan sukses. Ini akan memakan waktu sekitar 30 menit. Kegagalan apa pun yang mungkin Anda temui sering disebabkan oleh	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kuota API. Dalam kasus ini, Anda dapat menjalankan kembali pipeline untuk melanjutkan penerapan. Langkah 4: Periksa sumber daya yang dibuat • Akses akun manajemen AFT dan konfirmasikan bahwa sumber daya telah dibuat.	

Pemecahan masalah

Isu	Solusi
Fungsi Lambda khusus yang disertakan dalam CloudFormation template gagal selama penerapan.	Periksa CloudWatch log Amazon untuk fungsi Lambda untuk mengidentifikasi kesalahan. Log memberikan informasi terperinci dan dapat membantu menentukan masalah spesifik. Konfirmasikan bahwa fungsi Lambda memiliki izin yang diperlukan dan variabel lingkungan telah disetel dengan benar.
Anda mengalami kegagalan dalam pembuatan atau manajemen sumber daya yang disebabkan oleh izin yang tidak memadai.	Tinjau peran dan kebijakan IAM yang dilampirkan pada fungsi Lambda CodeBuild, dan layanan lain yang terlibat dalam penerapan. Konfirmasikan bahwa mereka memiliki izin yang diperlukan. Jika ada masalah izin, sesuaikan kebijakan IAM untuk memberikan akses yang diperlukan.
Anda menggunakan versi CloudFormation template yang sudah ketinggalan zaman	Perbarui CloudFormation template secara teratur agar kompatibel dengan rilis terbaru

Isu	Solusi
dengan versi yang lebih baru Layanan AWS atau Terraform.	AWS dan Terraform. Periksa catatan rilis atau dokumentasi untuk setiap perubahan atau persyaratan khusus versi.
Anda mencapai Layanan AWS kuota selama penerapan.	Sebelum Anda menerapkan pipeline, periksa Layanan AWS kuota untuk sumber daya seperti bucket S3, peran IAM, dan fungsi Lambda. Permintaan meningkat jika perlu. Untuk informasi lebih lanjut, lihat Layanan AWS kuota di AWS situs web.
Anda mengalami kesalahan karena parameter input yang salah dalam CloudFormation template.	Periksa kembali semua parameter input untuk kesalahan ketik atau nilai yang salah. Konfirmasikan bahwa pengidentifikasi sumber daya, seperti nama akun IDs dan Wilayah, akurat.

Sumber daya terkait

Untuk menerapkan pola ini dengan sukses, tinjau sumber daya berikut. Sumber daya ini memberikan informasi dan panduan tambahan yang dapat sangat berharga dalam menyiapkan dan mengelola AFT dengan menggunakan. CloudFormation

AWSdokumentasi:

- AWS Control Tower Panduan Pengguna menawarkan informasi rinci tentang pengaturan dan pengelolaan AWS Control Tower.
- CloudFormation dokumentasi memberikan wawasan tentang CloudFormation template, tumpukan, dan manajemen sumber daya.

Kebijakan dan praktik terbaik IAM:

- Praktik terbaik keamanan di IAM menjelaskan cara membantu mengamankan AWS sumber daya dengan menggunakan peran dan kebijakan IAM.

Terraform pada: AWS

- Dokumentasi AWS Penyedia Terraform memberikan informasi komprehensif tentang penggunaan Terraform dengan. AWS

Layanan AWS kuota:

- Layanan AWS kuota memberikan informasi tentang cara melihat Layanan AWS kuota dan cara meminta kenaikan.

Kelola produk AWS Service Catalog di beberapa akun AWS dan Wilayah AWS

Ram Kandaswamy, Amazon Web Services

Ringkasan

Amazon Web Services (AWS) Service Catalog menyederhanakan dan mempercepat tata kelola dan distribusi templat infrastruktur sebagai kode (IAc) untuk perusahaan. Anda menggunakan CloudFormation templat AWS untuk menentukan kumpulan sumber daya AWS (tumpukan) yang diperlukan untuk suatu produk. AWS CloudFormation StackSets memperluas fungsionalitas ini dengan memungkinkan Anda membuat, memperbarui, atau menghapus tumpukan di beberapa akun dan Wilayah AWS dengan satu operasi.

Administrator AWS Service Catalog membuat produk dengan menggunakan CloudFormation templat yang ditulis oleh pengembang, dan mempublikasikannya. Produk-produk ini kemudian dikaitkan dengan portofolio, dan kendala diterapkan untuk tata kelola. Agar produk Anda tersedia bagi pengguna di akun AWS lain atau unit organisasi (OUs), Anda biasanya membagikan portofolio Anda dengan mereka. Pola ini menjelaskan pendekatan alternatif untuk mengelola penawaran produk AWS Service Catalog yang didasarkan pada AWS. CloudFormation StackSets Alih-alih berbagi portofolio, Anda menggunakan batasan set tumpukan untuk menyetel Wilayah AWS dan akun tempat produk Anda dapat digunakan dan digunakan. Dengan menggunakan pendekatan ini, Anda dapat menyediakan produk AWS Service Catalog Anda di beberapa akun OUs, dan Wilayah AWS, serta mengelolanya dari lokasi pusat, sekaligus memenuhi persyaratan tata kelola Anda.

Manfaat dari pendekatan ini:

- Produk disediakan dan dikelola dari akun utama, dan tidak dibagikan dengan akun lain.
- Pendekatan ini memberikan pandangan konsolidasi dari semua produk yang disediakan (tumpukan) yang didasarkan pada produk tertentu.
- Konfigurasi dengan AWS Service Management Connector lebih mudah, karena hanya menargetkan satu akun.
- Lebih mudah untuk menanyakan dan menggunakan produk dari AWS Service Catalog.

Prasyarat dan batasan

Prasyarat

- CloudFormation Template AWS untuk IAc dan pembuatan versi
- Penyiapan multi-akun dan AWS Service Catalog untuk menyediakan dan mengelola sumber daya AWS

Batasan

- Pendekatan ini menggunakan AWS CloudFormation StackSets, dan batasan StackSets penerapan:
  - StackSets tidak mendukung penerapan CloudFormation template melalui makro. Jika Anda menggunakan makro untuk memproses template, Anda tidak akan dapat menggunakan penerapan StackSets berbasis.
  - StackSets menyediakan kemampuan untuk memisahkan tumpukan dari kumpulan tumpukan, sehingga Anda dapat menargetkan tumpukan tertentu untuk memperbaiki masalah. Namun, tumpukan yang terputus tidak dapat dikaitkan kembali dengan kumpulan tumpukan.
- AWS Service Catalog membuat StackSet nama secara otomatis. Kustomisasi saat ini tidak didukung.

Arsitektur

Arsitektur target

1. Pengguna membuat CloudFormation template AWS untuk menyediakan sumber daya AWS, dalam format JSON atau YAMM.
2. CloudFormation Template membuat produk di AWS Service Catalog, yang ditambahkan ke portofolio.
3. Pengguna membuat produk yang disediakan, yang membuat CloudFormation tumpukan di akun target.
4. Setiap tumpukan menyediakan sumber daya yang ditentukan dalam CloudFormation templat.

Alat

Layanan AWS

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Service Catalog membantu Anda mengelola katalog layanan TI secara terpusat yang disetujui untuk AWS. Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.

Epik

Menyediakan produk di seluruh akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat portofolio.	Portofolio adalah wadah yang mencakup satu atau lebih produk yang dikelompokkan bersama berdasarkan kriteria tertentu. Menggunakan portofolio untuk produk Anda membantu Anda menerapkan kendala umum di seluruh rangkaian produk Anda. Untuk membuat portofolio, ikuti petunjuk dalam dokumentasi AWS Service Catalog. Jika Anda menggunakan AWS CLI, berikut adalah contoh perintah:	AWS Service Catalog, IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`aws servicecatalog create-portfolio --provider-name my-provider --display-name my-portfolio` Untuk informasi selengkapnya, lihat dokumentasi AWS CLI.	
Buat CloudFormation template.	Buat CloudFormation template yang menjelaskan sumber daya. Nilai properti sumber daya harus diparameterisasi jika berlaku.	AWS CloudFormation, JSON/YAMB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat produk dengan informasi versi.	CloudFormation Template menjadi produk saat Anda mempublikasikannya di AWS Service Catalog. Berikan nilai untuk parameter detail versi opsional, seperti judul dan deskripsi versi; ini akan membantu untuk menanyakan produk nanti. Untuk membuat produk, ikuti petunjuk dalam dokumentasi AWS Service Catalog. Jika Anda menggunakan AWS CLI, perintah contohnya adalah: `aws servicecatalog create-product --cli-input-json file://create-product-input.json` di `create-product-input.json` mana file yang melewati parameter untuk produk. Untuk contoh file ini, lihat bagian Informasi tambahan. Untuk informasi selengkapnya, lihat dokumentasi AWS CLI.	AWS Service Catalog

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Terapkan kendala.	Terapkan batasan set tumpukan ke portofolio, untuk mengonfigurasi opsi penerapan produk seperti beberapa akun AWS, Wilayah, dan izin. Untuk petunjuk, lihat dokumentasi AWS Service Catalog.	AWS Service Catalog
Tambahkan izin.	Berikan izin kepada pengguna sehingga mereka dapat meluncurkan produk dalam portofolio. Untuk petunjuk konsol, lihat dokumentasi AWS Service Catalog. Jika Anda menggunakan AWS CLI, berikut adalah contoh perintah: `aws servicecatalog associate-principal-with-portfolio \ --portfolio-id port-2s6abcdefwdh4 \ --principal-arn arn:aws:iam::444455556666:role/Admin \ --principal-type IAM` Untuk informasi selengkapnya, lihat dokumentasi AWS CLI.	AWS Service Catalog, IAM

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan produk.	Produk yang disediakan adalah instans sumber daya dari suatu produk. Penyediaan produk berdasarkan CloudFormation template meluncurkan CloudFormation tumpukan dan sumber daya dasarnya. Menyediakan produk dengan menargetkan Wilayah dan akun AWS yang berlaku, berdasarkan batasan kumpulan tumpukan. Di AWS CLI, berikut adalah contoh perintah: `aws servicecatalog provision-product \ --product-id prod-abcdfz3syn2rg \ --provisioning-artifact-id pa-abc347pcsccfm \ --provisioned-product-name "mytestppname3"` Untuk informasi selengkapnya, lihat dokumentasi AWS CLI.	AWS Service Catalog

Sumber daya terkait

Referensi

- Ikhtisar AWS Service Catalog

- Menggunakan AWS CloudFormation StackSets

Tutorial dan video

- AWS re:Invent 2019: Otomatiskan semuanya: Opsi dan praktik terbaik (video)

Informasi tambahan

Bila Anda menggunakan `create-product` perintah, `cli-input-json` parameter menunjuk ke file yang menentukan informasi seperti pemilik produk, email dukungan, dan rincian CloudFormation template. Berikut adalah contoh dari file tersebut:

```
{
   "Owner": "Test admin",
      "SupportDescription": "Testing",
         "Name": "SNS",
            "SupportEmail": "example@example.com",
            "ProductType": "CLOUD_FORMATION_TEMPLATE",
               "AcceptLanguage": "en",
                  "ProvisioningArtifactParameters": {
                     "Description": "SNS product",
                        "DisableTemplateValidation": true,
                           "Info": {
                              "LoadTemplateFromURL": "<url>"
                     },
                           "Name": "version 1"
}
```

Pantau cluster SAP RHEL Pacemaker dengan menggunakan layanan AWS

Harsh Thoria, Randy Germann, dan RAVEENDRA Voore, Amazon Web Services

Ringkasan

Pola ini menguraikan langkah-langkah untuk memantau dan mengonfigurasi peringatan untuk cluster Pacemaker Red Hat Enterprise Linux (RHEL) untuk aplikasi SAP dan layanan database SAP HANA dengan menggunakan Amazon dan Amazon Simple Notification Service (Amazon CloudWatch SNS).

Konfigurasi ini memungkinkan Anda untuk memantau SAP SCS atau ASCS, Enqueue Replication Server (ERS), dan sumber daya cluster SAP HANA ketika mereka berada dalam keadaan “berhenti” dengan bantuan aliran CloudWatch log, filter metrik, dan alarm. Amazon SNS mengirimkan email ke infrastruktur atau tim SAP Basis tentang status cluster yang dihentikan.

Anda dapat membuat AWS sumber daya untuk pola ini dengan menggunakan AWS CloudFormation skrip atau konsol AWS layanan. Pola ini mengasumsikan bahwa Anda menggunakan konsol; itu tidak menyediakan CloudFormation skrip atau mencakup penyebaran infrastruktur untuk dan Amazon CloudWatch SNS. Perintah alat pacu jantung digunakan untuk mengatur konfigurasi peringatan cluster.

Prasyarat dan batasan

Prasyarat

- AWS Akun aktif.
- Amazon SNS diatur untuk mengirim email atau pemberitahuan seluler.
- SAP ASCS/ERS untuk ABAP atau SCS/ERS untuk Java, dan SAP HANA Database RHEL Pacemaker cluster. Untuk petunjuk, lihat yang berikut ini:
  - Pengaturan kluster SAP HANA
  - Pengaturan kluster SAP Netweaver ABAP/Java

Batasan

- Solusi ini saat ini berfungsi untuk RHEL versi 7.3 dan kluster berbasis Pacemaker yang lebih baru. Itu belum diuji pada sistem operasi SUSE.

Versi produk

- RHEL 7.3 dan yang lebih baru

Arsitektur

Tumpukan teknologi target

- Agen yang digerakkan oleh peristiwa peringatan RHEL Pacemaker
- Amazon Elastic Compute Cloud (Amazon EC2)
- CloudWatch alarm
- CloudWatch grup log dan filter metrik
- Amazon SNS

Arsitektur target

Diagram berikut menggambarkan komponen dan alur kerja untuk solusi ini.

Otomatisasi dan skala

- Anda dapat mengotomatiskan pembuatan AWS sumber daya dengan menggunakan CloudFormation skrip. Anda juga dapat menggunakan filter metrik tambahan untuk menskalakan dan mencakup beberapa cluster.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

Alat

- CloudWatch agent (unified) adalah alat yang mengumpulkan metrik tingkat sistem, log, dan jejak dari EC2 instance, dan mengambil metrik kustom dari aplikasi Anda.

- Agen peringatan alat pacu jantung (untuk RHEL 7.3 dan yang lebih baru) adalah alat yang memulai tindakan ketika ada perubahan, seperti ketika sumber daya berhenti atau restart, di cluster Pacemaker.

Praktik terbaik

- Untuk praktik terbaik dalam menggunakan beban kerja SAP AWS, lihat Lensa SAP untuk Kerangka AWS Well-Architected.
- Pertimbangkan biaya yang terlibat dalam pengaturan CloudWatch pemantauan untuk cluster SAP HANA. Lihat informasi yang lebih lengkap dalam dokumentasi CloudWatch .
- Pertimbangkan untuk menggunakan pager atau mekanisme tiket untuk peringatan Amazon SNS.
- Selalu periksa versi RHEL high availability (HA) dari paket RPM untuk PC, Pacemaker, dan agen anggar. AWS

Epik

Siapkan Amazon SNS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat sebuah topik SNS.	1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon SNS di https://console.aws.amazon.com/sns/ v3/home. 2. Pada dasbor Amazon SNS, pada Tindakan umum, silakan pilih Buat Topik. 3. Dalam kotak dialog Buat topik baru, untuk Jenis, pilih Standar. 4. Untuk nama Topik, masukkan nama untuk topik (misalnya,`my-topic`).	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Pilih Buat topik. Ini membuat topik SNS dengan kebijakan sumber daya yang memungkinkan Anda mempublikasikan notifikasi. 6. Salin Topik ARN (misalnya ,`arn:aws:sns:us-east-1:111122223333:my-topic`). Anda akan menggunakan ARN ini di langkah selanjutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah kebijakan akses untuk topik SNS.	1. Di konsol Amazon SNS, di panel navigasi, pilih Topik, lalu pilih topik yang Anda buat. 2. Pilih Edit dan buka bagian Kebijakan akses. 3. Pastikan bahwa kebijakan akses termasuk CloudWatch sebagai salah satu prinsip layanan yang diizinkan untuk mempublikasikan ke topik ini. Contoh: `{ "Sid": "Allow AWS CloudWatch to Publish to this SNS topic", "Effect": "Allow", "Principal": { "Service": [ "cloudwatch.amazonaws.com" ] }, "Action": "SNS:Publish", "Resource": "arn:aws:sns:us-east-1:111122223333:my-topic" }` 4. Pilih Simpan perubahan.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berlangganan topik SNS.	1. Di konsol Amazon SNS, di panel navigasi, pilih Langganan, Buat langganan. 2. Untuk Topik ARN, tempel ARN yang Anda buat di tugas pertama. 3. Untuk Protokol, pilih Email. 4. Untuk Endpoint, masukkan alamat email untuk orang atau tim yang bertanggu ng jawab untuk cluster SAP Pacemaker dan harus menerima pemberitahuan. Misalnya, ini bisa menjadi alamat email untuk SAP Basis atau daftar distribusi tim infrastruktur. 5. Pilih Buat langganan. 6. Dari aplikasi email Anda, buka pesan dari AWS Pemberitahuan dan konfirmasikan langganan Anda. Browser web Anda menampilk an respons konfirmasi dari Amazon SNS.	Administrator sistem AWS

Konfirmasikan pengaturan cluster

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa status cluster.	Gunakan perintah status pcs untuk mengonfirmasi bahwa sumber daya sedang online.	Administrator SAP Basis

Konfigurasikan peringatan Pacemaker

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan agen peringatan Pacemaker pada instance cluster utama.	Masuk ke EC2 instance di cluster pimary dan jalankan perintah berikut:
	`install --mode=0755 /usr/share/pacemaker/alerts/alert_file.sh.sample`
	`touch /var/lib/pacemaker/alert_file.sh`
	`touch /var/log/pcmk_alert_file.log`
	`chown hacluster:haclient /var/log/pcmk_alert_file.log`
	`chmod 600 /var/log/pcmk_alert_file.log`
	`pcs alert create id=alert_file description="Log events to a file." path=/var/lib/pacemaker/alert_file.sh`
	`pcs alert recipient add alert_file id=my-alert_logfile value=/va`
		Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`r/log/pcmk_alert_f ile.log`	

Konfigurasikan agen peringatan Pacemaker pada instance cluster sekunder.	Masuk ke EC2 instance cluster sekunder di cluster sekunder dan jalankan perintah berikut:
	`install --mode=0755 / usr/share/pacemak er/alerts/alert_fi le.sh.sample`
	`touch /var/lib/ pacemaker/alert_fi le.sh`
	`touch /var/log/ pcmk_alert_file.log`
	`chown hacluster :haclient /var/log/ pcmk_alert_file.log`
	`chmod 600 /var/log/ pcmk_alert_file.log`
		Administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfirmasikan bahwa sumber daya peringatan RHEL telah dibuat.	Gunakan perintah berikut untuk mengonfirmasi bahwa sumber daya peringatan telah dibuat:
	`pcs alert`
	Output dari perintah akan terlihat seperti ini:
	`[root@xxxxxxx ~]# pcs alert`
	`Alerts:`
	`Alert: alert_file (path=/var/lib/pacemaker/alert_file.sh)`
	`Description: Log events to a file.`
	`Recipients:`
	`Recipient: my-alert_logfile (value=/var/log/pcmk_alert_file.log)`
		Administrator SAP Basis

Konfigurasikan CloudWatch agen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal CloudWatch agen.	Ada beberapa cara untuk menginstal CloudWatch agen pada sebuah EC2 instance. Untuk menggunakan baris perintah: 1. Unduh paket CloudWatch agen:	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`wget https://s3.<region>.amazonaws.com/amazoncloudwatch-agent-region/redhat/amd64/latest/amazon-cloudwatch-agent.rpm`
	
	di Wilayah AWS mana `<region>` adalah tempat EC2 instance berada (misalnya,`us-west-2`).
	2. Opsional) Verifikasi tanda tangan paket. Untuk petunjuk, lihat Memverifikasi tanda tangan paket CloudWatch agen dalam CloudWatch dokumentasi.
	3. Instal paket pada contoh pertama:
	
	`sudo rpm -U ./amazon-cloudwatch-agent.rpm`
	
	4. Ulangi untuk contoh sekunder.
	
	Lihat informasi yang lebih lengkap dalam dokumentasi CloudWatch .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan peran IAM ke EC2 instance.	Untuk mengaktifkan CloudWatch agen mengirim data dari instance, Anda harus melampirkan CloudWatc hAgentServerRoleperan IAM ke setiap instance. Atau, Anda dapat menambahkan kebijakan untuk CloudWatch agen ke peran IAM Anda yang ada. Lihat informasi yang lebih lengkap dalam dokumentasi CloudWatch .	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan CloudWatch agen untuk memantau file log agen peringatan Pacemaker pada instance cluster utama.	1. Konfigurasikan instance cluster primer dengan menjalankan perintah: `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard` 2. Pilih 1 untuk Linux, lalu pilih opsi untuk strategi pemantauan Anda. 3. Untuk pertanyaan “Apakah Anda ingin memantau file log apa pun,” pilih Ya dan berikan jalur file log Pacemaker dari perintah peringatan pcs. Dalam kasus kami, itu`var/log/pcmk_alert_file.log`. 4. Berikan nama grup log dan aliran log. Jika Anda tidak menentukan aliran log, ID AWS instance digunakan sebagai default. 5. Ulangi langkah 1-4 untuk instance cluster sekunder.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mulai CloudWatch agen pada instance cluster primer dan sekunder.	Untuk memulai agen, jalankan perintah berikut pada EC2 instance di cluster primer dan sekunder: `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json`	Administrator AWS

Siapkan CloudWatch sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan grup CloudWatch log.	1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/ 2. Di panel navigasi, pilih Grup log, Buat grup log. 3. Masukkan nama untuk grup log, lalu pilih Buat grup log. CloudWatch Agen akan mentransfer file peringatan Pacemaker ke grup CloudWatch log sebagai aliran log.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan filter CloudWatch metrik.	Filter metrik membantu Anda mencari pola seperti `stop <cluster-resource-name>` di aliran CloudWatch log. Ketika pola ini diidentifikasi, filter metrik memperbarui metrik khusus. 1. Di CloudWatch konsol, di panel navigasi, pilih Grup log. 2. Pilih nama grup log yang Anda buat di tugas sebelumnya. 3. Pilih Actions (Tindakan), Create metric filter (Buat filter metrik). 4. Untuk pola Filter, masukkan pola filter yang akan digunakan, seperti`stop ABC_scs`, untuk mencocokkan peristiwa berhenti untuk sumber daya cluster SAP SCS bernama. `ABC_scs` Untuk informasi selengkapnya, lihat Filter sintaks pola dalam CloudWatch dokumentasi. 5. (Opsional) Untuk menguji pola filter Anda, di Test Pattern (Pola Uji), masukkan satu atau	Administrator AWS, administrator SAP Basis

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	beberapa log acara untuk digunakan menguji pola. Setiap peristiwa log harus ditentukan pada baris terpisah, karena jeda baris digunakan untuk memisahkan peristiwa log di kotak pesan peristiwa Log. 6. Pilih Next (Selanjutnya), lalu masukkan nama untuk filter. 7. Di bawah Detail metrik, untuk namespace Metrik, masukkan nama untuk CloudWatch namespace tempat metrik akan dipublikasikan (misalnya,). `sapcluster_monitoring` Jika namespace ini belum ada, pilih Buat baru. 8. Untuk nama Metrik, masukkan nama untuk metrik baru (misalnya `sapcluster_<sid>` , di `<sid>` mana nama identifikasi sistem SAP). 9. Untuk nilai Metrik, masukkan 1. Atau, Anda dapat memasukkan token seperti`$size`. Peningkatan ini akan menambahkan metrik sebesar nilai angka	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di bidang `size` untuk setiap log acara yang berisi bidang `size`. 10. Untuk nilai Default, masukkan 0. 11. Pilih Create metric filter (Buat filter metrik). Ketika filter metrik mengidentifikasi pola pada langkah 4, itu memperbarui nilai metrik CloudWatch kustom `sapcluster_abc` ke 1. CloudWatch Alarm `SAP-Cluster-QA1-ABC` memonitor metrik `sapcluster_abc` dan mengirimkan pemberitahuan SNS ketika nilai metrik berubah menjadi 1. Ini menunjukkan bahwa sumber daya cluster telah berhenti dan tindakan perlu diambil.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan alarm CloudWatch metrik untuk metrik SAP ASCS/SCS dan ERS.	Untuk membuat alarm berdasarkan metrik tunggal: 1. Di CloudWatch konsol, di panel navigasi, pilih Alarm, Semua alarm. 2. Pilih Buat alarm. 3. Pilih Pilih Metrik. 4. Cari metrik kustom `sapcluster_monitoring` yang dibuat di tugas sebelumnya. 5. Pilih nama metrik untuk SAP SCS (misalnya ,`sapcluster_<abc>`), yang juga dibuat dalam tugas sebelumnya. 6. Pada tab Graphed metrics, atur yang berikut ini: • Untuk Statistik, pilih Maksimum. • Untuk Periode, pilih 1 menit. • Untuk tipe Threshold , pilih Static dan atur ambang batas **`sapcluster_<sid>`** untuk nilai yang lebih besar dari atau sama dengan 1. 7. Pilih Berikutnya.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. Untuk Pemberitahuan, pilih topik SNS yang Anda buat di epik pertama. 9. Untuk Nama dan Deskripsi, berikan nama alarm dan deskripsi singkat, lalu pilih Berikutnya. 10. Pilih Buat Alarm.	
Siapkan alarm CloudWatch metrik untuk metrik SAP HANA.	Ulangi langkah-langkah untuk mengatur alarm CloudWatch metrik dari tugas sebelumnya, dengan perubahan ini: • Untuk langkah 5, pilih nama metrik untuk SAP HANA (misalnya, `sapcluster_db_<abc>`). • Untuk langkah 6, tetapkan ambang batas `sapcluster_<sid>` untuk nilai yang lebih besar dari 0.	Administrator AWS

Sumber daya terkait

- Memicu Skrip untuk Acara Cluster (dokumentasi RHEL)
- Buat file konfigurasi CloudWatch agen dengan wizard (CloudWatch dokumentasi)
- Menginstal dan menjalankan CloudWatch agen di server Anda (CloudWatch dokumentasi)
- Buat CloudWatch alarm berdasarkan ambang statis (CloudWatch dokumentasi)
- Penyebaran manual SAP HANA di AWS dengan cluster ketersediaan tinggi (dokumentasi SAP di situs web) AWS
- NetWeaver Panduan SAP (dokumentasi SAP di situs web) AWS

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Memantau aktivitas aplikasi dengan menggunakan Wawasan CloudWatch Log

Ram Kandaswamy, Amazon Web Services

Ringkasan

Pola ini memberikan solusi untuk mendeteksi dan memperingatkan pengecualian aplikasi secara otomatis dengan menggunakan Amazon CloudWatch Logs Insights. Dengan menerapkan analisis log otomatis dan peringatan, Anda dapat dengan cepat mengidentifikasi dan menanggapi masalah aplikasi di lingkungan produksi Anda.

Log memainkan peran penting dalam memantau perilaku sistem, mengidentifikasi masalah, dan memastikan kinerja yang optimal. Selama proses migrasi, file log sangat berharga untuk memvalidasi fungsi sistem di lingkungan baru, mendeteksi masalah kompatibilitas, dan mengidentifikasi perilaku yang tidak terduga. Masalah bisa terkait dengan operasi atau keamanan. Untuk masalah terkait keamanan, memungkinkan deteksi upaya akses yang tidak sah atau aktivitas mencurigakan sejak dini sangat penting untuk menjaga keamanan dan kepatuhan terhadap peraturan. Kemampuan ini sangat penting ketika berhadapan dengan data sensitif atau sistem kritis.

Pola ini sangat berharga bagi tim yang perlu melakukan hal berikut:

- Pertahankan ketersediaan aplikasi yang tinggi.
- Menanggapi masalah produksi dengan cepat.
- Analisis kesalahan khusus aplikasi yang tidak ditangkap oleh Layanan AWS log.
- Lakukan analisis log sesuai permintaan tanpa infrastruktur yang dibangun sebelumnya.

CloudWatch Logs Insights optimal untuk menganalisis log yang dihasilkan aplikasi di mana konteks kesalahan hanya ada dalam kode aplikasi Anda. CloudWatch Log Insights unggul dalam tugas-tugas berikut:

- Kueri data log tidak terstruktur atau semi-terstruktur.
- Lakukan analisis sesuai permintaan selama respons insiden.
- Korelasikan peristiwa di beberapa grup log.
- Buat visualisasi cepat tanpa alat eksternal.

Prasyarat dan batasan

Prasyarat

- Aplikasi produksi yang digunakan secara aktif Akun AWS
- Pemahaman dasar tentang format logging aplikasi produksi dan pola pengecualian
- Log aplikasi dikonfigurasi untuk streaming ke Amazon CloudWatch Logs

Batasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat AWS Layanan menurut Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.

Arsitektur

Diagram berikut menunjukkan cara CloudWatch Logs Insights mengevaluasi log sumber daya dan mengirimkan visualisasi data yang relevan ke dasbor. CloudWatch

Diagram menunjukkan alur kerja berikut:

1. Sumber daya mempublikasikan log ke CloudWatch Log. Sumber daya dapat menyertakan AWS sumber daya seperti instans Amazon Elastic Compute Cloud (Amazon EC2) atau bucket Amazon Simple Storage Service (Amazon S3). Contoh lain termasuk sistem lokal dengan CloudWatch Agen diinstal yang dapat mempublikasikan log ke CloudWatch.
2. CloudWatch Filter Wawasan Log untuk string pola yang relevan. Contoh string pola pencarian termasuk “kesalahan”, “pengecualian”, atau ekspresi reguler tertentu.
3. Biasanya, tim dukungan produksi atau pengembang menambahkan visualisasi pola ke CloudWatch dasbor.

Otomatisasi dan skala

Pengembang dapat mengotomatiskan solusi pola ini dengan menggunakan AWS Cloud Development Kit (AWS CDK), AWS CloudFormation, atau AWS SDKs untuk menangani beberapa pola string. Tim dapat memasukkan otomatisasi ini ke dalam proses integrasi dan penerapan berkelanjutan (CI/CD) mereka. DevOps

Alat

Layanan AWS

- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.

Praktik terbaik

Efisiensi kueri

- Tentukan dan konfigurasikan grup log untuk menganalisis data log yang relevan.
- Gunakan penjelajah lapangan untuk memahami struktur dan bidang yang tersedia di data log Anda.
- Tulis kueri yang efisien dengan menggunakan sintaks kueri CloudWatch Logs Insights.
- Sesuaikan kueri sampel dengan kebutuhan spesifik Anda untuk analisis yang lebih cepat.
- Batasi rentang waktu kueri untuk mengurangi data yang dipindai dan meningkatkan kinerja.
- Simpan kueri untuk penggunaan di masa mendatang untuk menghemat waktu dan memastikan analisis yang konsisten.

Keamanan

- Menerapkan kebijakan IAM yang sesuai ke Wawasan CloudWatch Log dan grup log. Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.
- Aktifkan enkripsi data log menggunakan AWS KMS data log sensitif.

Optimalisasi biaya

- CloudWatch Log Insights mengenakan biaya per GB data yang dipindai per kueri. Rentang waktu yang sempit dan targetkan grup log tertentu untuk mengurangi biaya.
- Konfigurasikan kebijakan penyimpanan log yang sesuai untuk mengelola biaya penyimpanan.
- Untuk analisis yang sering dari kumpulan data historis besar, pertimbangkan untuk mengekspor log ke Amazon S3 dan menggunakan Amazon Athena.
- Tinjau CloudWatch harga untuk memahami implikasi biaya untuk kasus penggunaan Anda.

Epik

Buat grup log dan konfigurasikan log untuk dilihat di dasbor.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan izin IAM.	Untuk mengonfigurasi izin IAM, lakukan hal berikut: 1. Identifikasi peran IAM yang digunakan oleh aplikasi Anda untuk menulis log dan oleh pengguna atau layanan yang akan membuat dasbor, kueri, dan alarm. 2. Untuk peran aplikasi, tambahkan tindakan dan sumber daya API berikut ke kebijakan: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [`	Administrator AWS, AWS DevOps, administrator sistem AWS, Administrator cloud, arsitek Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"logs:CreateLogGroup",
	"logs:CreateLogStream",
	"logs:PutLogEvents"
	],
	"Resource": [
	"arn:aws:logs:*:*:*"
	]
	}
	]
	}</pre>
	3. Untuk mengelola CloudWatch sumber daya, tambahkan hal berikut ke kebijakan:
	<pre>{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"cloudwatch:PutDashboard",
	"cloudwatch:GetDashboard",
	"logs:StartQuery",</pre>	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"logs:StopQuery",
	
	"logs:GetQueryResu
	lts",
	
	"cloudwatch:PutMet
	ricAlarm"
	],
	"Resource
	": "*"
	}
	]
	}</pre>
	Untuk informasi tentang cara membuat kebijakan IAM atau menambahkan izin ke kebijakan yang ada, lihat Menentukan izin IAM khusus dengan kebijakan yang dikelola pelanggan dan Mengedit kebijakan IAM di Panduan Pengguna IAM. Untuk informasi selengkapnya, lihat Identitas dan manajemen akses untuk referensi izin CloudWatch CloudWatch Log dan Log Amazon di Panduan Pengguna CloudWatch Log Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Membuat sebuah grup log.	Untuk membuat grup log, gunakan salah satu opsi berikut:
	• Buat atau buka file CloudFormation template (format YAMAL atau JSON) dengan ekstensi `.yaml` atau `.json`. (Kode berikut menggunakan format YAMB.) Tambahkan definisi sumber daya berikut ke bagian Sumber daya template Anda:
	`MyLogGroup:`
	`  Type: AWS::Logs`
	`::LogGroup`
	`  Properties:`
	`  LogGroupName: my-`
	`log-group`
	Untuk informasi selengkap nya, lihat Mulai Cepat: Gunakan AWS CloudFormation untuk memulai CloudWatch Log di Panduan Pengguna CloudWatch Log Amazon.
	• Gunakan CloudWatch konsol, seperti yang dijelaskan dalam Membuat grup CloudWatch log di
		Administrator AWS, AWS DevOps, administrator sistem AWS, Administrator cloud, arsitek Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Log di Panduan Pengguna CloudWatch Log Amazon. • Gunakan CreateLog Groupoperasi di CloudWatch Logs API.	
Hasilkan kueri Wawasan CloudWatch Log.	Untuk membuat dan menyimpan kueri Wawasan CloudWatch Log, lakukan hal berikut: 1. Buka CloudWatch konsol. 2. Arahkan ke Wawasan Log. 3. Buat kueri Anda dengan menggunakan salah satu metode berikut: • Mulailah dengan kueri sampel dari bagian Kueri. • Tulis kueri khusus. • Ubah contoh query berikut: `fields @timestamp, @message | filter @message like /(?i)exception/|stats count() by bin(30s)` Kueri ini memeriksa file log, mengekstrak stempel waktu dan konten pesan, memfilter kata “pengecualian” (case-insensitive), dan menghitung kejadian dalam interval 30 detik.	Administrator AWS, AWS DevOps, administrator sistem AWS, Administrator cloud, arsitek Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat visualisasi di CloudWatch dasbor.	Untuk menggunakan CloudWatch dasbor untuk membuat visualisasi, lakukan hal berikut: 1. Buka CloudWatch konsol, dan buat CloudWatch dasbor (jika Anda belum memilikinya) atau buka dasbor yang ada. 2. Tambahkan visualisasi pemantauan pengecualian Anda. Bagan batang, garis, dan lingkaran tersedia, tergantung pada kebutuhan bisnis Anda. Untuk informasi selengkapnya, lihat Menggunakan widget di CloudWatch dasbor di Panduan Pengguna CloudWatch Log Amazon. Untuk visualisasi data real-time, Anda dapat menyesuaikan widget untuk memenuhi kebutuhan Anda. Untuk informasi selengkapnya tentang opsi dan kemampuan dasbor, lihat Menggunakan CloudWatch dasbor Amazon dan Membuat CloudWatch dasbor fleksibel dengan variabel dasbor di Panduan	Administrator AWS, AWS DevOps, administrator sistem AWS, Administrator cloud, arsitek Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pengguna CloudWatch Log Amazon.	

Pemecahan masalah

Isu	Solusi
Tidak dapat melihat hasil kueri atau kueri tampaknya rusak	Mulailah dengan kueri kerja yang dimodifik asi dari kueri sampel. Lakukan perubahan inkremental kecil pada bagian kueri (seperti filter atau bidang), dan manfaatkan fitur generator kueri CloudWatch Log.
Grup log tidak membuat aliran log	Dalam kebijakan IAM, pastikan bahwa sumber daya untuk CreateLogStreamdan CreateLog Groupoperasi dilengkapi dengan nilai karakter (`*`) wildcard. `create` Operasi tidak akan berhasil tanpa izin wildcard ini.
Batas waktu kueri atau kinerja lambat	Kurangi rentang waktu, targetkan grup log tertentu, atau sederhanakan kueri. Pola ekspresi reguler (`regex`) yang kompleks dan rentang waktu yang besar meningkatkan waktu kueri.
Tidak ada data yang dikembalikan untuk rentang waktu yang valid	Verifikasi pemilihan grup log dan periksa apakah log sedang dicerna (tinjau aliran log), dan konfirmasikan pola filter cocok dengan format log Anda.

Sumber daya terkait

- Menganalisis data log dengan Wawasan CloudWatch Log
- Amazon CloudWatch FAQs

- Membuat CloudWatch dasbor fleksibel dengan variabel dasbor
- Memulai dengan Logs Insights QL: Tutorial kueri
- Gunakan bahasa alami untuk menghasilkan dan memperbarui kueri Wawasan CloudWatch Log
- Gunakan PutDashboard dengan AWS SDK atau CLI
- Bekerja dengan grup log dan aliran log

Pantau penggunaan Gambar Mesin Amazon bersama di beberapa Akun AWS

Naveen Suthar dan Sandeep Gawande, Amazon Web Services

Ringkasan

Amazon Machine Images (AMIs) digunakan untuk membuat instans Amazon Elastic Compute Cloud (Amazon EC2) di lingkungan Amazon Web Services (AWS) Anda. Anda dapat membuat AMIs akun AWS terpusat yang terpisah, yang disebut akun pembuat dalam pola ini. Anda kemudian dapat membagikan AMI di beberapa Akun AWS yang sama Wilayah AWS, yang disebut akun konsumen dalam pola ini. Mengelola AMIs dari satu akun memberikan skalabilitas dan menyederhanakan tata kelola. Di akun konsumen, Anda dapat mereferensikan AMI bersama di templat peluncuran Amazon EC2 Auto Scaling dan grup node Amazon Elastic Kubernetes Service (Amazon EKS).

Ketika AMI bersama tidak digunakan lagi, dideregistrasi, atau tidak dibagikan, yang Layanan AWS merujuk ke AMI di akun konsumen tidak dapat menggunakan AMI ini untuk meluncurkan instans baru. Setiap peristiwa penskalaan otomatis atau peluncuran ulang instance yang sama gagal. Hal ini dapat menyebabkan masalah di lingkungan produksi, seperti downtime aplikasi atau penurunan kinerja. Ketika peristiwa berbagi dan penggunaan AMI terjadi dalam beberapa kali Akun AWS, akan sulit untuk memantau aktivitas ini.

Pola ini membantu Anda memantau penggunaan dan status AMI bersama di seluruh akun di Wilayah yang sama. Ini menggunakan tanpa server Layanan AWS, seperti Amazon, Amazon DynamoDB EventBridge, AWS Lambda, dan Amazon Simple Email Service (Amazon SES). Anda menyediakan infrastruktur sebagai kode (IAc) dengan menggunakan HashiCorp Terraform. Solusi ini memberikan peringatan ketika layanan di akun konsumen mereferensikan AMI yang dideregistrasi atau tidak dibagikan.

Prasyarat dan batasan

Prasyarat

- Dua atau lebih aktif Akun AWS: satu akun kreator dan satu atau lebih akun konsumen
- Satu atau lebih AMIs yang dibagikan dari akun pembuat ke akun konsumen
- Terraform CLI, diinstal (dokumentasi Terraform)
- AWS Penyedia Terraform, dikonfigurasi (dokumentasi Terraform)

- (Opsional, tetapi disarankan) Backend Terraform, dikonfigurasi (dokumentasi Terraform)
- Git, diinstal

Batasan

- Pola ini memantau AMIs yang telah dibagikan ke akun tertentu dengan menggunakan ID akun. Pola ini tidak memantau AMIs yang telah dibagikan ke organisasi dengan menggunakan ID organisasi.
- AMIs hanya dapat dibagikan ke akun yang berada dalam hal yang sama Wilayah AWS. Pola ini memantau AMIs dalam satu Wilayah target. Untuk memantau penggunaan AMIs di beberapa Wilayah, Anda menerapkan solusi ini di setiap Wilayah.
- Pola ini tidak memantau apa pun AMIs yang dibagikan sebelum solusi ini diterapkan. Jika Anda ingin memantau yang dibagikan sebelumnya AMIs, Anda dapat membatalkan pembagian AMI dan kemudian membagikannya kembali dengan akun konsumen.

Versi produk

- Terraform versi 1.2.0 atau yang lebih baru
- AWS Penyedia Terraform versi 4.20 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

Sumber daya berikut disediakan sebagai IAc melalui Terraform:

- Tabel Amazon DynamoDB
- EventBridge Aturan Amazon
- AWS Identity and Access Management Peran (IAM)
- AWS Lambda fungsi
- Amazon SES

Arsitektur target

Diagram menunjukkan alur kerja berikut:

1. AMI di akun pembuat dibagikan dengan akun konsumen dalam hal yang sama Wilayah AWS.
2. Saat AMI dibagikan, EventBridge aturan di akun pembuat akan menangkap `ModifyImageAttribute` acara dan memulai fungsi Lambda di akun pembuat.
3. Fungsi Lambda menyimpan data yang terkait dengan AMI dalam tabel DynamoDB di akun pembuat.
4. Saat akun Layanan AWS konsumen menggunakan AMI bersama untuk meluncurkan EC2 instans Amazon atau saat AMI bersama dikaitkan dengan templat peluncuran, EventBridge aturan di akun konsumen menangkap penggunaan AMI bersama.
5. EventBridge Aturan memulai fungsi Lambda di akun konsumen. Fungsi Lambda melakukan hal berikut:
   a. Fungsi Lambda memperbarui data terkait AMI dalam tabel DynamoDB di akun konsumen.
   b. Fungsi Lambda mengasumsikan peran IAM di akun pembuat dan memperbarui tabel Lambda di akun pembuat. Dalam `Mapping` tabel, ia membuat item yang memetakan ID instance atau meluncurkan ID template ke ID AMI masing-masing.
6. AMI yang dikelola secara terpusat di akun pembuat tidak digunakan lagi, dideregistrasi, atau tidak dibagikan.
7. EventBridge Aturan di akun pembuat menangkap `DeregisterImage` peristiwa `ModifyImageAttribute` atau dengan `remove` tindakan dan memulai fungsi Lambda.
8. Fungsi Lambda memeriksa tabel DynamoDB untuk menentukan apakah AMI digunakan di salah satu akun konsumen. Jika tidak ada instance IDs atau template peluncuran IDs yang terkait dengan AMI dalam `Mapping` tabel, maka prosesnya selesai.
9. Jika ada instance IDs atau template peluncuran IDs yang terkait dengan AMI dalam `Mapping` tabel, maka fungsi Lambda menggunakan Amazon SES untuk mengirim pemberitahuan email ke pelanggan yang dikonfigurasi.

Alat

Layanan AWS

- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data real-time dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Email Service (Amazon SES) membantu Anda mengirim dan menerima email dengan menggunakan alamat email dan domain Anda sendiri.

Alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub cross-account-ami-monitoring-terraform-samples.

Praktik terbaik

- Ikuti Praktik terbaik untuk bekerja dengan AWS Lambda fungsi.
- Ikuti Praktik terbaik untuk membangun AMIs.
- Saat membuat peran IAM, ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Berikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi IAM.
- Siapkan pemantauan dan peringatan untuk AWS Lambda fungsi. Untuk informasi selengkapnya, lihat Memantau dan memecahkan masalah fungsi Lambda.

Epik

Sesuaikan file konfigurasi Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat profil AWS CLI bernama.	Untuk akun pembuat dan setiap akun konsumen, buat profil bernama AWS Command Line Interface (AWS CLI). Untuk petunjuk, lihat Mengatur AWS CLI di Pusat Sumber Daya AWS Memulai.	DevOps insinyur
Kloning repositori.	Masukkan perintah berikut. Ini mengkloning repositori cross-account-ami-monitoring-terraform-samples dari dengan menggunakan SSH. GitHub `git clone git@github.com:aws-samples/cross-account-ami-monitoring-terraform-samples.git`	DevOps insinyur
Perbarui file provider.tf.	1. Masukkan perintah berikut untuk menavigasi ke `terraform` folder di repositori kloning. `cd cross-account-ami-monitoring/terraform` 2. Buka file `provider.tf` . 3. Perbarui konfigurasi AWS Penyedia Terraform untuk	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akun pembuat dan akun konsumen sebagai berikut: • Untuk`alias`, masukkan nama untuk konfigurasi penyedia. • Untuk`region`, masukkan target Wilayah AWS tempat Anda ingin menerapkan solusi ini. • Untuk`profile`, masukkan profil AWS CLI bernama untuk mengakses akun. 4. Jika Anda mengonfig urasi lebih dari satu akun konsumen, buat profil untuk setiap akun konsumen tambahan. 5. Simpan dan tutup file `provider.tf` . Untuk informasi selengkap nya tentang mengonfigurasi penyedia, lihat Beberapa konfigurasi penyedia dalam dokumentasi Terraform.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file terraform.tfvars.	1. Buka file `terraform.tfvars`. 2. Dalam `account_email_mapping` parameter, konfigurasikan peringatan untuk akun pembuat dan akun konsumen sebagai berikut: • Untuk `account`, masukkan ID akun. • Untuk `email`, masukkan alamat email tempat Anda ingin mengirim peringatan. Anda hanya dapat memasukkan satu alamat email untuk setiap akun. 3. Jika Anda mengonfigurasi lebih dari satu akun konsumen, masukkan akun dan alamat email untuk setiap akun konsumen tambahan. 4. Simpan dan tutup file `terraform.tfvars`.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui file main.tf.	Selesaikan langkah-langkah ini hanya jika Anda menerapkan solusi ini ke lebih dari satu akun konsumen. Jika Anda menerapkan solusi ini hanya ke satu akun konsumen, tidak diperlukan modifikasi file ini. 1. Buka file `main.tf`. 2. Untuk setiap akun konsumen tambahan, buat modul baru yang didasarkan pada `consumer_account_A` modul di template. Untuk setiap akun konsumen, untuk`provider`, nilainya harus sesuai dengan alias yang Anda masukkan dalam `provider.tf` file. 3. Simpan dan tutup file `main.tf`.	DevOps insinyur

Terapkan solusi dengan menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan solusinya.	Di CLI Terraform, masukkan perintah berikut untuk menyebarkan AWS sumber daya di akun pembuat dan konsumen:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	1. Masukkan perintah berikut untuk menginisialisasi Terraform.
	`terraform init`
	2. Masukkan perintah berikut untuk memvalidasi konfigurasi Terraform.
	`terraform validate`
	3. Masukkan perintah berikut untuk membuat rencana eksekusi Terraform.
	`terraform plan`
	4. Tinjau perubahan konfigurasi dalam paket Terraform dan konfirmasikan bahwa Anda ingin menerapkan perubahan ini.
	5. Masukkan perintah berikut untuk menyebarkan sumber daya.
	`terraform apply`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi identitas alamat email.	Saat Anda menerapkan paket Terraform, Terraform membuat identitas alamat email untuk setiap akun konsumen di Amazon SES. Sebelum pemberitahuan dapat dikirim ke alamat email tersebut, Anda harus memverifikasi alamat email tersebut. Untuk petunjuk, lihat Memverifikasi identitas alamat email dalam dokumentasi Amazon SES.	AWS Umum

Validasi penyebaran sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan di akun pembuat.	1. Masuk ke akun pembuat. 2. Di bilah navigasi, konfirmasikan bahwa sedang melihat target Wilayah AWS. Jika Anda berada di Wilayah lain, pilih nama Wilayah yang ditampilkan saat ini, lalu pilih Wilayah target. 3. Buka Konsol DynamoDB. 4. Di panel navigasi, pilih Tabel. 5. Dalam daftar tabel, validasi bahwa `AmiShare` tabel ada. 6. Buka Konsol Lambda.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Di panel navigasi, pilih Fungsi. 8. Dalam daftar fungsi, validasi bahwa `ami-share` fungsi tersebut ada. 9. Buka konsol IAM. 10. Di panel navigasi, pilih Peran. 11. Dalam daftar peran, validasi bahwa `external-ddb-role` peran tersebut ada. 12. Buka konsol EventBridge . 13. Di panel navigasi, pilih Aturan. 14. Dalam daftar aturan, validasi bahwa `modify_image_attribute_event` aturan ada. 15. Buka konsol Amazon SES. 16. Di panel navigasi, pilih Identitas Terverifikasi. 17. Dalam daftar identitas, validasi bahwa identitas alamat email telah terdaftar dan diverifikasi untuk setiap akun konsumen.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penyebaran di akun konsumen.	1. Masuk ke akun konsumen. 2. Di bilah navigasi, konfirmasikan bahwa sedang melihat target Wilayah AWS. Jika Anda berada di Wilayah lain, pilih nama Wilayah yang ditampilkan saat ini, lalu pilih Wilayah target. 3. Buka Konsol DynamoDB. 4. Di panel navigasi, pilih Tabel. 5. Dalam daftar tabel, validasi bahwa `Mapping` tabel ada. 6. Buka Konsol Lambda. 7. Di panel navigasi, pilih Fungsi. 8. Dalam daftar fungsi, validasi bahwa `ami-deregister-function` fungsi `ami-usage-function` dan ada. 9. Buka konsol EventBridge . 10. Di panel navigasi, pilih Aturan. 11. Dalam daftar aturan, validasi bahwa `ami_usage_events` dan `ami_deregister_events` aturan ada.	DevOps insinyur

Validasi pemantauan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat AMI di akun pembuat.	1. Di akun pembuat, buat AMI pribadi. Untuk petunjuknya, lihat Membuat AMI dari EC2 instans Amazon. 2. Bagikan AMI baru dengan salah satu akun konsumen. Untuk petunjuk, lihat Berbagi AMI dengan spesifik Akun AWS.	DevOps insinyur
Gunakan AMI di akun konsumen.	Di akun konsumen, gunakan AMI bersama untuk membuat EC2 instance Amazon atau templat peluncuran. Untuk petunjuknya, lihat Bagaimana cara meluncurkan EC2 instans Amazon dari AMI kustom (AWS Re:Post Knowledge Center) atau Membuat template peluncuran untuk grup Auto Scaling (dokumentasi Amazon Auto EC2 Scaling).	DevOps insinyur
Validasi pemantauan dan peringatan.	1. Masuk ke akun pembuat. 2. Buka EC2 konsol Amazon. 3. Di panel navigasi, pilih AMIs. 4. Pilih AMI dalam daftar, lalu pilih Tindakan, Edit izin AMI.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	5. Di bagian Akun bersama, pilih akun konsumen, lalu pilih Hapus yang dipilih. 6. Pilih Simpan perubahan. 7. Validasi bahwa alamat email target yang Anda tetapkan untuk akun konsumen menerima pemberitahuan bahwa pembagian dibatalkan untuk AMI.	

(Opsional) Hentikan pemantauan bersama AMIs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus sumber daya.	1. Masukkan perintah berikut untuk menghapus sumber daya yang digunakan oleh pola ini dan hentikan pemantauan bersama AMIs. `terraform destroy` 2. Konfirmasikan `destroy` perintah dengan memasukkan `yes`.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
Saya tidak menerima peringatan email.	Mungkin ada beberapa alasan mengapa email Amazon SES tidak dikirim. Periksa hal-hal berikut: 1. Di bagian Epik, gunakan epik penyebaran sumber daya Validasi untuk mengonfirmasi bahwa infrastruktur telah disediakan dengan benar di semua. Akun AWS 2. Validasi peristiwa fungsi Lambda di Amazon CloudWatch Logs. Untuk petunjuk, lihat Menggunakan CloudWatch konsol di dokumentasi Lambda. Konfirmasikan bahwa tidak ada masalah izin, seperti penolakan eksplisit dalam kebijakan berbasis identitas atau sumber daya apa pun. Untuk informasi selengkapnya, lihat Logika evaluasi kebijakan dalam dokumentasi IAM. 3. Di Amazon SES, validasi bahwa status identitas alamat email telah Diverifik asi. Untuk informasi selengkapnya, lihat Memverifikasi identitas alamat email.

Sumber daya terkait

AWS dokumentasi

- Membangun fungsi Lambda dengan Python (dokumentasi Lambda)
- Buat AMI (EC2 dokumentasi Amazon)
- Bagikan AMI dengan spesifik Akun AWS (EC2 dokumentasi Amazon)
- Batalkan pendaftaran AMI Anda (dokumentasi Amazon) EC2

Dokumentasi Terraform

- Instal Terraform
- Konfigurasi Backend Terraform
- Penyedia AWS Terraform
- Unduhan biner Terraform

Lihat detail snapshot EBS untuk akun atau organisasi AWS

Arun Chandapillai dan Parag Nagwekar, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana Anda dapat secara otomatis membuat laporan sesuai permintaan dari semua snapshot Amazon Elastic Block Store (Amazon EBS) di akun Amazon Web Services (AWS) atau unit organisasi (OU) di AWS Organizations.

Amazon EBS adalah easy-to-use layanan penyimpanan blok berkinerja tinggi yang dapat diskalakan yang dirancang untuk Amazon Elastic Compute Cloud (Amazon). EC2 Volume EBS menyediakan penyimpanan yang tahan lama dan persisten yang dapat Anda lampirkan ke EC2 instans Anda. Anda dapat menggunakan volume EBS sebagai penyimpanan utama untuk data Anda dan mengambil point-in-time cadangan volume EBS Anda dengan membuat snapshot. Anda dapat menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI) Interface (AWS CLI) untuk melihat detail snapshot EBS tertentu. Pola ini menyediakan cara terprogram untuk mengambil informasi tentang semua snapshot EBS di akun AWS atau OU Anda.

Anda dapat menggunakan skrip yang disediakan oleh pola ini untuk menghasilkan file nilai yang dipisahkan koma (CSV) yang memiliki informasi berikut tentang setiap snapshot: ID akun, ID snapshot, ID volume dan ukuran, tanggal snapshot diambil, ID instance, dan deskripsi. Jika snapshot EBS Anda diberi tag, laporan tersebut juga menyertakan atribut pemilik dan tim.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- AWS CLI versi 2 diinstal dan dikonfigurasi
- Peran AWS Identity and Access Management (IAM) dengan izin yang sesuai (izin akses untuk akun tertentu atau untuk semua akun di OU jika Anda berencana menjalankan skrip dari AWS Organizations)

Arsitektur

Diagram berikut menunjukkan alur kerja skrip yang menghasilkan laporan snapshot EBS sesuai permintaan yang tersebar di beberapa akun AWS dalam OU.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- Amazon Elastic Block Store (Amazon EBS) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans. EC2
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Organizations adalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.

Kode

Kode untuk aplikasi sampel yang digunakan dalam pola ini tersedia di GitHub, di aws-ebs-snapshots-awsorganizationsrepositori. Ikuti petunjuk di bagian selanjutnya untuk menggunakan file sampel.

Epik

Unduh skripnya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh skrip Python.	Unduh skrip GetSnapshotDetailsAllAccountsOU.py dari GitHub repositori.	AWS Umum

Dapatkan detail snapshot EBS untuk akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip Python.	Jalankan perintah :	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`python3 getsnapsh otinfo.py --file <output-file>.csv --region <region-name>`
	
	where `<output-file>` mengacu pada file keluaran CSV tempat Anda ingin informasi tentang snapshot EBS ditempatkan, dan `<region-name>` merupakan Wilayah AWS tempat snapshot disimpan. Contoh:
	
	`python3 getsnapsh otinfo.py --file snapshots.csv --region us-east-1`	

Dapatkan detail snapshot EBS untuk organisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip Python.	Jalankan perintah : `python3 getsnapsh otinfo.py --file <output-file>.csv --role <IAM-role> --region <region-name>` where `<output-file>` mengacu pada file keluaran CSV tempat Anda ingin informasi tentang snapshot	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	EBS ditempatkan, `<IAM-role>` adalah peran yang menyediakan izin untuk mengakses AWS Organizations, dan merupakan `<region-name>` Wilayah AWS tempat snapshot disimpan. Contoh: `python3 getsnapshotinfo.py --file snapshots.csv --role <IAM role> --region us-west-2`	

Sumber daya terkait

- Dokumentasi Amazon EBS
- Tindakan Amazon EBS
- Referensi API Amazon EBS
- Meningkatkan kinerja Amazon EBS
- Sumber daya Amazon EBS
- Harga snapshot EBS

Informasi tambahan

Jenis snapshot EBS

Amazon EBS menyediakan tiga jenis snapshot, berdasarkan kepemilikan dan akses:

- Dimiliki oleh Anda - Secara default, hanya Anda yang dapat membuat volume dari snapshot yang Anda miliki.
- Snapshot publik — Anda dapat membagikan snapshot secara publik dengan semua akun AWS lainnya. Untuk membuat snapshot publik, Anda mengubah izin snapshot untuk membagikannya

dengan akun AWS yang Anda tentukan. Pengguna yang akan Anda otorisasi kemudian dapat menggunakan snapshot yang Anda bagikan dengan membuat volume EBS mereka sendiri, sementara snapshot asli Anda tetap tidak terpengaruh. Anda juga dapat membuat snapshot tidak terenkripsi tersedia untuk umum untuk semua pengguna AWS. Namun, Anda tidak dapat membuat snapshot terenkripsi tersedia untuk umum karena alasan keamanan. Cuplikan publik menimbulkan risiko keamanan yang signifikan karena kemungkinan mengekspos data pribadi dan sensitif. Kami sangat menyarankan agar tidak membagikan snapshot EBS Anda dengan semua akun AWS. Untuk informasi selengkapnya tentang berbagi snapshot, lihat dokumentasi AWS.

- Snapshot pribadi — Anda dapat membagikan snapshot secara pribadi dengan akun AWS individual yang Anda tentukan. Untuk membagikan snapshot secara pribadi dengan akun AWS tertentu, ikuti petunjuk dalam dokumentasi AWS, dan pilih Private untuk setelan izin. Pengguna yang Anda beri wewenang dapat menggunakan snapshot yang Anda bagikan untuk membuat volume EBS mereka sendiri, sementara snapshot asli Anda tetap tidak terpengaruh.

Ikhtisar dan prosedur

Tabel berikut menyediakan tautan ke informasi lebih lanjut tentang snapshot EBS, termasuk bagaimana Anda dapat menurunkan biaya volume EBS dengan menemukan dan menghapus snapshot yang tidak digunakan, dan mengarsipkan snapshot yang jarang diakses yang tidak memerlukan pengambilan yang sering atau cepat.

Untuk informasi tentang	Lihat
Snapshot, fitur-fiturnya, dan keterbatasannya	Buat snapshot Amazon EBS
Cara membuat snapshot	Konsol: Buat snapshot AWS CLI: perintah buat-snapshot Contoh: `aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description " volume snapshot"`
Menghapus snapshot (informasi umum)	Hapus snapshot Amazon EBS
Cara menghapus snapshot	Konsol: Hapus snapshot

	AWS CLI: perintah hapus-snapshot Contoh: `aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0`
Mengarsipkan snapshot (informasi umum)	Arsipkan snapshot Amazon EBS Arsip Snapshots Amazon EBS (posting blog)
Cara mengarsipkan snapshot	Konsol: Arsipkan snapshot AWS CLI: perintah modify-snapshot-tier
Cara mengambil snapshot yang diarsipkan	Konsol: Kembalikan snapshot yang diarsipkan AWS CLI: perintah restore-snapshot-tier
Harga snapshot	Harga Amazon EBS

FAQ

Berapa periode arsip minimum?

Periode arsip minimum adalah 90 hari.

Berapa lama waktu yang dibutuhkan untuk memulihkan snapshot yang diarsipkan?

Diperlukan waktu hingga 72 jam untuk memulihkan snapshot yang diarsipkan dari tingkat arsip ke tingkat standar, tergantung pada ukuran snapshot.

Apakah snapshot yang diarsipkan penuh snapshot?

Snapshot yang diarsipkan selalu merupakan snapshot penuh.

Snapshot mana yang dapat diarsipkan pengguna?

Anda hanya dapat mengarsipkan snapshot yang Anda miliki di akun Anda.

Bisakah Anda mengarsipkan snapshot volume perangkat root dari Amazon Machine Image (AMI) yang terdaftar?

Tidak, Anda tidak dapat mengarsipkan snapshot volume perangkat root dari AMI terdaftar.

Apa pertimbangan keamanan untuk berbagi snapshot?

Saat Anda membagikan snapshot, Anda memberi orang lain akses ke semua data pada snapshot. Bagikan snapshot hanya dengan orang yang Anda percayai dengan data Anda.

Bagaimana Anda membagikan snapshot dengan Wilayah AWS lain?

Snapshot dibatasi untuk Wilayah tempatnya dibuat. Untuk berbagi snapshot dengan Wilayah lain, salin snapshot ke Wilayah tersebut, lalu bagikan salinannya.

Bisakah Anda berbagi snapshot yang dienkripsi?

Anda tidak dapat membagikan snapshot yang dienkripsi dengan kunci terkelola AWS default. Anda dapat berbagi snapshot yang dienkripsi hanya dengan kunci yang dikelola pelanggan. Saat membagikan snapshot terenkripsi, Anda juga harus membagikan kunci terkelola pelanggan yang digunakan untuk mengenkripsi snapshot.

Bagaimana dengan snapshot yang tidak terenkripsi?

Anda dapat membagikan snapshot yang tidak terenkripsi secara publik.

Lebih banyak pola

- Mengotomatiskan pembuatan akun dengan menggunakan Akselerator Zona Landing di AWS
- Mengotomatiskan operasi AWS infrastruktur dengan menggunakan Amazon Bedrock
- Mengotomatiskan penilaian sumber daya AWS
- Inventarisasi AWS sumber daya secara otomatis di beberapa akun dan Wilayah
- Mengotomatiskan portofolio AWS Service Catalog dan penerapan produk dengan menggunakan AWS CDK
- Otomatiskan manajemen pipeline dinamis untuk menerapkan solusi perbaikan terbaru di lingkungan Gitflow dengan menggunakan dan AWS Service Catalog AWS CodePipeline
- Otomatiskan konsumsi dan visualisasi metrik kustom Amazon MWAA di Amazon Managed Grafana dengan menggunakan Terraform
- Secara otomatis melampirkan kebijakan terkelola AWS untuk Systems Manager ke profil EC2 instans menggunakan Cloud Custodian dan AWS CDK
- Secara otomatis mengenkripsi volume Amazon EBS yang ada dan baru
- Bangun AWS landing zone yang mencakup MongoDB Atlas
- Memusatkan pemantauan dengan menggunakan Amazon CloudWatch Observability Access Manager
- Periksa EC2 instance untuk tag wajib saat peluncuran
- Bersihkan AWS Account Factory untuk sumber daya Terraform (AFT) dengan aman setelah kehilangan file status
- Buat definisi tugas Amazon ECS dan pasang sistem file pada EC2 instans menggunakan Amazon EFS
- Buat aturan AWS Config khusus dengan menggunakan AWS CloudFormation Guard kebijakan
- Sesuaikan nama peran default dengan menggunakan AWS CDK aspek dan lubang keluar
- Menyebarkan dan mengelola AWS Control Tower kontrol dengan menggunakan AWS CDK dan CloudFormation
- Terapkan dan kelola AWS Control Tower kontrol dengan menggunakan Terraform
- Menerapkan kode di beberapa Wilayah AWS menggunakan AWS CodePipeline, AWS CodeCommit, dan AWS CodeBuild
- Menerapkan aplikasi kontainer saat AWS IoT Greengrass V2 berjalan sebagai wadah Docker

- Aktifkan Amazon GuardDuty secara kondisional dengan menggunakan templat AWS CloudFormation
- Aktifkan pengarsipan DB2 log langsung ke Amazon S3 dalam database IBM Db2
- Mengekspor laporan identitas AWS IAM Identity Center dan tugasnya dengan menggunakan PowerShell
- Buat CloudFormation template AWS yang berisi aturan terkelola AWS Config menggunakan Troposfer
- Berikan instans SageMaker notebook akses sementara ke CodeCommit repositori di akun AWS lain
- Integrasikan Stonebranch Universal Controller dengan AWS Mainframe Modernisasi
- Luncurkan CodeBuild proyek di seluruh akun AWS menggunakan Step Functions dan fungsi proxy Lambda
- Kelola set AWS izin secara dinamis dengan menggunakan Terraform
- Migrasi aplikasi yang dihosting IIS ke Amazon dengan menggunakan appcmd.exe EC2
- Migrasikan sertifikat SSL Windows ke Application Load Balancer menggunakan ACM
- Pantau aktivitas pengguna root IAM
- Buat arsitektur IPAM Multi-wilayah yang hierarkis dengan menggunakan Terraform AWS
- Optimalkan penerapan tanpa server multi-akun dengan menggunakan alur kerja dan Tindakan AWS CDK GitHub
- Pertahankan ruang IP yang dapat dirutekan dalam desain VPC multi-akun untuk subnet non-beban kerja
- Menyediakan peran IAM dengan hak istimewa paling sedikit dengan menerapkan solusi mesin penjual otomatis peran
- Daftarkan beberapa Akun AWS dengan satu alamat email dengan menggunakan Amazon SES
- Hapus EC2 entri Amazon di Akun AWS seberang AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi
- Hapus EC2 entri Amazon dalam hal yang sama Akun AWS dari AWS Managed Microsoft AD dengan menggunakan AWS Lambda otomatisasi
- Jalankan beban kerja yang digerakkan oleh peristiwa dan terjadwal dalam skala besar dengan AWS Fargate
- Amankan data sensitif di CloudWatch Log dengan menggunakan Amazon Macie

- Mengirim pemberitahuan untuk instans database Amazon RDS for SQL Server dengan menggunakan server SMTP lokal dan Database Mail
- Siapkan dasbor pemantauan Grafana untuk AWS ParallelCluster
- Siapkan logging terpusat pada skala perusahaan dengan menggunakan Terraform
- Siapkan pemulihan bencana untuk SAP di IBM Db2 di AWS
- Merampingkan manajemen EC2 kepatuhan Amazon dengan agen Amazon Bedrock dan AWS Config
- Tandai lampiran Transit Gateway secara otomatis menggunakan AWS Organizations
- Gunakan kueri BMC Discovery untuk mengekstrak data migrasi untuk perencanaan migrasi
- Verifikasi praktik terbaik operasional untuk PCI DSS 4.0 dengan menggunakan AWS Config
- Lihat log dan metrik AWS Network Firewall dengan menggunakan Splunk
- Visualisasikan laporan kredensi IAM untuk semua akun AWS menggunakan Amazon Quick Sight

Pesan & komunikasi

Topik

- Otomatiskan konfigurasi RabbitMQ di Amazon MQ
- Tingkatkan kualitas panggilan di workstation agen di pusat kontak Amazon Connect
- Lebih banyak pola

Otomatiskan konfigurasi RabbitMQ di Amazon MQ

Yogesh Bhatia dan Afroz Khan, Amazon Web Services

Ringkasan

Amazon MQ adalah layanan broker pesan terkelola yang menyediakan kompatibilitas dengan banyak broker pesan populer. Menggunakan Amazon MQ dengan RabbitMQ menyediakan kluster RabbitMQ yang kuat yang dikelola di Amazon Web Services (AWS) Cloud dengan beberapa broker dan opsi konfigurasi. Amazon MQ menyediakan infrastruktur yang sangat tersedia, aman, dan dapat diskalakan, dan dapat memproses sejumlah besar pesan per detik dengan mudah. Beberapa aplikasi dapat menggunakan infrastruktur dengan host virtual, antrian, dan pertukaran yang berbeda. Namun, mengelola opsi konfigurasi ini atau membuat infrastruktur secara manual dapat memerlukan waktu dan usaha. Pola ini menjelaskan cara untuk mengelola konfigurasi untuk RabbitMQ dalam satu langkah, melalui satu file. Anda dapat menyematkan kode yang disediakan dengan pola ini dalam alat integrasi berkelanjutan (CI) seperti Jenkins atau Bamboo.

Anda dapat menggunakan pola ini untuk mengonfigurasi cluster RabbitMQ apa pun. Yang dibutuhkan hanyalah konektivitas ke cluster. Meskipun ada banyak cara lain untuk mengelola konfigurasi RabbitMQ, solusi ini membuat seluruh konfigurasi aplikasi dalam satu langkah, sehingga Anda dapat mengelola antrian dan detail lainnya dengan mudah.

Prasyarat dan batasan

Prasyarat

- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi untuk menunjuk ke akun AWS Anda (untuk petunjuk, lihat dokumentasi AWS CLI)
- Ansible diinstal, sehingga Anda dapat menjalankan pedoman untuk membuat konfigurasi
- rabbitmqadmin diinstal (untuk petunjuk, lihat dokumentasi RabbitMQ)
- Cluster RabbitMQ di Amazon MQ, dibuat dengan metrik Amazon yang sehat CloudWatch

Persyaratan tambahan

- Pastikan untuk membuat konfigurasi untuk host virtual dan pengguna secara terpisah dan bukan sebagai bagian dari JSON.
- Pastikan bahwa konfigurasi JSON adalah bagian dari repositori dan dikontrol versi.

- Versi CLI rabbitmqadmin harus sama dengan versi server RabbitMQ, jadi opsi terbaik adalah mengunduh CLI dari konsol RabbitMQ.
- Sebagai bagian dari pipeline, pastikan sintaks JSON divalidasi sebelum setiap proses dijalankan.

Versi produk

- AWS CLI versi 2.0
- Ansible versi 2.9.13
- rabbitmqadmin versi 3.9.13 (harus sama dengan versi server RabbitMQ)

Arsitektur

Tumpukan teknologi sumber

- Kluster RabbitMQ yang berjalan di mesin virtual (VM) lokal yang ada atau klaster Kubernetes (di tempat atau di cloud)

Tumpukan teknologi target

- Konfigurasi RabbitMQ otomatis di Amazon MQ untuk RabbitMQ

Arsitektur target

Ada banyak cara untuk mengkonfigurasi RabbitMQ. Pola ini menggunakan fungsionalitas konfigurasi impor, di mana satu file JSON berisi semua konfigurasi. File ini menerapkan semua pengaturan dan dapat dikelola oleh sistem kontrol versi seperti Bitbucket atau Git. Pola ini menggunakan Ansible untuk mengimplementasikan konfigurasi melalui CLI rabbitmqadmin.

Alat

Alat

- rabbitmqadmin adalah alat baris perintah untuk API berbasis HTTP RabbitMQ. Ini digunakan untuk mengelola dan memantau node dan cluster RabbitMQ.
- Ansible adalah alat sumber terbuka untuk mengotomatisasi aplikasi dan infrastruktur TI.

- AWS CLI memungkinkan Anda berinteraksi dengan layanan AWS dengan menggunakan perintah di shell baris perintah.

Layanan AWS

- Amazon MQ adalah layanan broker pesan terkelola yang memudahkan untuk mengatur dan mengoperasikan broker pesan di cloud.
- AWS CloudFormation membantu Anda mengatur infrastruktur AWS dan mempercepat penyediaan cloud dengan infrastruktur sebagai kode.

Kode

File konfigurasi JSON yang digunakan dalam pola ini dan contoh buku pedoman Ansible disediakan di lampiran.

Epik

Buat infrastruktur AWS Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat cluster RabbitMQ di AWS.	Jika Anda belum memiliki cluster RabbitMQ, Anda dapat menggunakan AWS CloudFormation untuk membuat tumpukan di AWS. Atau, Anda dapat menggunakan modul Cloudformation di Ansible untuk membuat tumpukan. Dengan pendekatan yang terakhir, Anda dapat menggunakan Ansible untuk kedua tugas: untuk membuat infrastruktur RabbitMQ dan mengelola konfigurasi.	AWS CloudFormation, Ansible

Buat Amazon MQ untuk konfigurasi RabbitMQ

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat file properti.	Unduh file konfigurasi JSON (`rabbitmqconfig.jso n`) di lampiran, atau ekspor dari konsol RabbitMQ. Ubah untuk mengonfigurasi antrian, pertukaran, dan binding. File konfigurasi ini menunjukkan hal berikut: - Membuat dua antrian: dan `sample-queue1` `sample-queue2` - Menciptakan dua pertukara n: `sample-exchange1` dan `sample-exchange2` - Menerapkan pengikatan antara antrian dan pertukaran Konfigurasi ini dilakukan di bawah host virtual root (/), seperti yang dipersyaratkan oleh rabbitmqadmin.	JSON
Ambil detail Amazon MQ untuk infrastruktur RabbitMQ.	Ambil detail berikut untuk infrastruktur RabbitMQ di AWS: • Nama broker • Tuan rumah RabbitMQ • Nama pengguna RabbitMQ (pengguna administr	AWS CLI, Amazon MQ

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ator yang dibuat selama pembuatan cluster) • Kata sandi RabbitMQ Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mengambil informasi ini. Detail ini memungkinkan buku pedoman Ansible untuk terhubung ke akun AWS Anda dan menggunakan cluster RabbitMQ untuk menjalankan perintah. **Important** Komputer yang menjalankan buku pedoman Ansible harus dapat mengakses akun AWS Anda, dan AWS CLI harus sudah dikonfigurasi, seperti yang dijelaskan di bagian Prasyarat.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat file hosts_var.	Buat `hosts_var` file untuk Ansible dan pastikan bahwa semua variabel didefinisikan dalam file. Pertimbangkan untuk menggunakan Ansible Vault untuk menyimpan kata sandi. Anda dapat mengonfigurasi `hosts_var` file sebagai berikut (ganti tanda bintang dengan informasi Anda):
	`RABBITMQ_HOST: "**********.mq.us-east-2.amazonaws.com"`
	`RABBITMQ_VHOST: "/"`
	`RABBITMQ_USERNAME: "admin"`
	`RABBITMQ_PASSWORD: "*******"`
		Ansible

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat buku pedoman Ansible.	Untuk contoh buku pedoman, lihat `ansible-rabbit-config.yaml` di lampiran. Unduh dan simpan file ini. Buku pedoman Ansible mengimpor dan mengelola semua konfigurasi RabbitMQ, seperti antrian, pertukaran, dan binding, yang diperlukan aplikasi. Ikuti praktik terbaik untuk buku pedoman Ansible, seperti mengamankan kata sandi. Gunakan Ansible Vault untuk enkripsi kata sandi, dan ambil kata sandi RabbitMQ dari file terenkripsi.	Ansible

Deploy konfigurasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan buku pedoman.	Jalankan playbook Ansible yang Anda buat di epik sebelumnya. `ansible-playbook ansible-rabbit-config.yaml` Anda dapat memverifikasi konfigurasi baru di konsol RabbitMQ.	RabbitMQ, Amazon MQ, Ansible

Sumber daya terkait

- Bermigrasi dari RabbitMQ ke Amazon MQ (postingan blog AWS)
- Alat Baris Perintah Manajemen (dokumentasi RabbitMQ)
- Membuat atau menghapus CloudFormation tumpukan AWS (dokumentasi Ansible)
- Memigrasi aplikasi berbasis pesan ke Amazon MQ untuk RabbitMQ (posting blog AWS)

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Tingkatkan kualitas panggilan di workstation agen di pusat kontak Amazon Connect

Ernest Ozdoba, Amazon Web Services

Ringkasan

Masalah kualitas panggilan adalah beberapa masalah yang paling sulit untuk dipecahkan di pusat kontak. Untuk menghindari masalah kualitas suara dan prosedur pemecahan masalah yang rumit, Anda harus mengoptimalkan lingkungan kerja dan pengaturan workstation agen Anda. Pola ini menjelaskan teknik pengoptimalan kualitas suara untuk workstation agen di pusat kontak Amazon Connect. Ini memberikan rekomendasi di bidang-bidang berikut:

- Penyesuaian lingkungan kerja. Lingkungan agen tidak mempengaruhi bagaimana suara ditransmisikan melalui jaringan, tetapi mereka memiliki efek pada kualitas panggilan.
- Pengaturan stasiun kerja agen. Konfigurasi perangkat keras dan jaringan untuk workstation pusat kontak memiliki efek signifikan pada kualitas panggilan.
- Pengaturan browser. Agen menggunakan browser web untuk mengakses situs web Amazon Connect Contact Control Panel (CCP) dan berkomunikasi dengan pelanggan, sehingga pengaturan browser dapat memengaruhi kualitas panggilan.

Komponen berikut juga dapat memengaruhi kualitas panggilan, tetapi berada di luar cakupan stasiun kerja dan tidak tercakup dalam pola ini:

- Lalu lintas mengalir ke Amazon Web Services (AWS) Cloud melalui AWS Direct Connect, VPN terowongan penuh, atau VPN split-tunnel
- Kondisi jaringan saat bekerja di dalam atau di luar kantor perusahaan
- Konektivitas Public Switched Telephone Network (PSTN)
- Perangkat pelanggan dan operator telepon
- Penyiapan infrastruktur desktop virtual (VDI)

Untuk informasi selengkapnya yang berkaitan dengan area ini, lihat Masalah Common Contact Control Panel (CCP) dan Gunakan Utilitas Uji Titik Akhir dalam dokumentasi Amazon Connect.

Prasyarat dan batasan

Prasyarat

- Headset dan workstation harus mematuhi persyaratan yang ditentukan dalam Panduan Administrator Amazon Connect.

Batasan

- Teknik pengoptimalan dalam pola ini berlaku untuk kualitas suara ponsel lunak. Mereka tidak berlaku saat Anda mengonfigurasi Amazon Connect CCP dalam mode telepon meja. Namun, Anda dapat menggunakan mode telepon meja jika pengaturan ponsel lunak Anda tidak memberikan kualitas suara yang dapat diterima untuk panggilan tersebut.

Versi produk

- Untuk browser dan versi yang didukung, lihat Panduan Administrator Amazon Connect.

Arsitektur

Pola ini bersifat arsitektur-agnostik karena menargetkan pengaturan stasiun kerja agen. Seperti yang ditunjukkan diagram berikut, jalur suara dari agen ke pelanggan dipengaruhi oleh headset, browser, sistem operasi, perangkat keras workstation, dan jaringan agen.

Di pusat kontak Amazon Connect, konektivitas audio pengguna dibuat dengan WebRTC. Suara dikodekan dengan codec audio interaktif Opus dan dienkripsi dengan Secure Real-time Transport Protocol (SRTP) dalam perjalanan. Arsitektur jaringan lain dimungkinkan, termasuk VPN, WAN/LAN pribadi, dan jaringan ISP.

Alat

- Amazon Connect Endpoint Test Utility - Utilitas ini memeriksa konektivitas jaringan dan pengaturan browser.
- Editor konfigurasi browser untuk pengaturan WebRTC:
  - Untuk Firefox: about:config
  - Untuk Chrome: chrome: //flags
- CCP Log Parser - Alat ini membantu Anda menganalisis log CCP untuk tujuan pemecahan masalah.

Epik

Sesuaikan lingkungan kerja

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kurangi kebisingan latar belakang.	Hindari lingkungan yang bising. Jika ini tidak memungkinkan, optimalkan lingkungan dengan tips kedap suara ini: • Menyerap kebisingan dengan menggunakan permukaan yang menghilangkan suara seperti gorden, karpet, dan perabotan lembut. • Blokir kebisingan dengan menempatkan penghalang di antara meja. • Pertimbangkan solusi peredam bising aktif (ANC) seperti generator white noise untuk membantu konsentrasi dan memastikan privasi, atau menggunakan headset peredam bising. • Cegah gema pada panggilan Anda. Ruang besar dan kosong dapat menciptakan efek gema atau memperkuat suara. Menutupi permukaan yang dapat memantulkan suara	Agen, Manajer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akan membantu mengurangi gema.	

Optimalkan pengaturan stasiun kerja agen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih headset yang tepat.	• Jika lingkungannya bising, pilih headset stereo. Mengarahkan suara ke kedua telinga membantu agen fokus dan mendengar pelanggan dengan lebih baik, dan mengurangi kebisingan secara keseluruhan dengan mengurangi kemungkinan agen untuk mengangkat suara mereka. • Hindari menggunakan pengeras suara atau audio komputer bawaan. Untuk kualitas terbaik, gunakan headset kabel yang didedikasikan untuk penggunaan pusat kontak. Headset nirkabel nyaman, tetapi mereka mungkin menjadi sumber penundaan audio tambahan dan penurunan kualitas audio karena gangguan radio dan transcoding.	Agen, Manajer

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan headset sebagaima na dimaksud.	• Aktifkan fitur peredam bising aktif dan peningkatan suara headset Anda jika tersedia. Cari pengaturan seperti ANC atau ANR. Untuk petunjuk cara mengaktif kan pengaturan ini, lihat panduan pengguna untuk headset Anda. • Sesuaikan mikrofon Anda sehingga Anda dapat berbicara langsung ke dalamnya. Posisi terbaik untuk mikrofon Anda tepat di bawah dagu Anda. Penempatan yang benar dapat membuat perbedaan 10 desibel (dB) di tingkat suara. Sebagian besar headset memungkin kan Anda memutar atau menekuk lengan mikrofon (boom), jadi penting untuk menyimpannya di tempat yang tepat saat Anda berbicara. • Beberapa headset dilengkapi dengan beberapa mikrofon dan fitur-fitur canggih seperti voice beamforming, yang membantu menangkap ucapan tanpa ledakan. Untuk memastikan bahwa	Agen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda menggunakan mikrofon utama sebagaima na dimaksud oleh pabrikan, lihat panduan pengguna untuk perangkat Anda.	
Periksa sumber daya workstati on.	Pastikan komputer agen Anda berkinerja baik. Jika mereka menggunakan aplikasi pihak ketiga yang mengkonsumsi sumber daya, komputer mereka mungkin tidak memenuhi persyarat an perangkat keras minimum untuk menjalankan CCP. Jika agen mengalami masalah kualitas panggilan, pastikan mereka memiliki cukup daya pemrosesan (CPU), ruang disk, bandwidth jaringan, dan memori yang tersedia untuk CCP. Agen harus menutup aplikasi dan tab yang tidak perlu untuk meningkatkan kinerja PKC dan kualitas panggilan.	Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pengaturan suara sistem operasi.	Pengaturan default untuk level mikrofon dan boost biasanya berfungsi dengan baik. Jika Anda menemukan bahwa suara keluar tenang atau mikrofon terlalu banyak mengambil, mungkin membantu untuk menyesuai kan pengaturan ini. Pengatura n mikrofon dapat ditemukan di konfigurasi suara sistem komputer Anda (Suara, Input di macOS, Properti Mikrofon di Windows). Anda dapat mengakses pengatura n lanjutan yang mungkin memengaruhi kualitas suara melalui alat sistem atau aplikasi pihak ketiga. Berikut adalah beberapa pengaturan yang dapat Anda periksa: • Laju sampel — Nilai ini menentukan berapa kali suara diperiksa per detik. Pengaturan default biasanya 44 atau 48 kilohertz (kHz). Nilai optimal untuk Amazon Connect adalah 48 kHz. Anda dapat menggunakan pengaturan browser untuk mengganti nilai default. Untuk informasi selengkapnya, lihat bagian pemecahan masalah pada	Agen, Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Panduan Administrator Amazon Connect. • Gain — Nilai ini menentuka n seberapa banyak mikrofon memperkuat suara. Jika Anda menaikkan gain, mikrofon Anda mungkin menangkap lebih banyak kebisingan latar belakang. • Kedalaman bit — Nilai resolusi digital ini menggambarkan berapa banyak tingkat amplitudo suara yang dikenali. Semakin tinggi kedalaman bit, semakin halus suara suara. Namun, banyak jaringan telepon tradision al menggunakan standar modulasi kode pulsa (PCM), yang hanya mendukung resolusi 8-bit. • Ambang batas terbuka — Ini adalah amplitudo suara minimal yang diambil mikrofon. Jika Anda mengalami masalah kualitas suara, coba pulihkan nilai ini ke setelan defaultny a sebelum menyelidiki lebih lanjut.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkapnya tentang pengaturan ini dan pengaturan lain yang dapat disesuaikan, lihat manual perangkat Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan jaringan kabel.	Biasanya, kabel ethernet memiliki latensi yang lebih rendah, sehingga lebih mudah untuk memberikan kualitas transmisi yang konsisten yang diperlukan untuk transmisi data suara. Kami merekomendasikan bandwidth 100 KB per panggilan minimal. • Jika agen bekerja dari rumah, kami sarankan kabel melalui koneksi nirkabel. Seharusnya tidak lebih dari 150 milidetik untuk mendengar pelanggan. Anda dapat mengakses uji latensi Amazon Connect dari Amazon Connect Endpoint Test Utility. Namun, utilitas ini mengukur penundaan dari browser ke Amazon Connect Regions, bukan ke pelanggan. Rekomendasi penundaan satu arah 150 milidetik mencegah agen dan pelanggan saling berbicara . Nilai diukur dari ujung ke ujung, dan setiap elemen menambahkan penundaan, termasuk bagian panggilan antara Wilayah Amazon Connect dan pelanggan.	Administrator jaringan, Agen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Jika agen bekerja dari kantor, Wi-Fi perusahaan dapat diterima selama parameter berada dalam kisaran yang disarankan, dan lalu lintas Protokol Transportasi Real-time (RTP) diprioritaskan.	
Perbarui driver perangkat keras.	Saat Anda menggunakan USB atau jenis headset lain yang memiliki firmware sendiri, kami sarankan Anda tetap memperbaruinya dengan versi terbaru. Headset sederhana yang menggunakan port tambahan menggunakan perangkat audio bawaan komputer, jadi pastikan driver perangkat keras sistem operasi mutakhir. Dalam kasus yang jarang terjadi, pembaruan driver audio dapat menyebabkan masalah audio, dan Anda mungkin perlu memutarnya kembali. Untuk informasi selengkapnya tentang mengubah versi firmware dan driver, lihat manual perangkat Anda.	Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hindari hub USB dan dongle.	Saat Anda menghubungkan headset, hindari perangkat tambahan seperti dongle, konverter tipe port, hub, dan kabel ekstensi. Perangkat ini dapat memengaruhi kualitas panggilan. Sambungkan perangkat Anda langsung ke port di komputer Anda.	Agen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa log PKC.	CCP Log Parser menyediakan cara mudah untuk memeriksa log aplikasi. 1. Unduh log PKC setelah panggilan. 2. Buka Parser Log PKT. 3. Seret dan lepas file log untuk mengunggah log untuk analisis. 4. Ketika log telah dianalisis, tab Snapshots & Log akan dipilih secara default. Pilih tab Metrik di sebelahnya untuk memeriksa wawasan. 5. Di bagian WebRTC Metrics - audio_input, periksa yang berikut ini: • Grafik Level Audio, untuk melihat apakah level audio yang Anda terima di atas 0. Ini menunjukkan bahwa audio diterima dari penelepon Anda. • Grafik Paket untuk setiap paket yang hilang. Jika grafik ini menunjukkan peningkatan yang signifikan, hubungi tim dukungan TI Anda. 6. Di bagian WebRTC Metrics - audio_output, periksa yang berikut ini:	Agen (keterampilan lanjutan)

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Grafik Level Audio, untuk mengonfirmasi bahwa audio telah dikirim keluar dari perangkat Anda. • Grafik Paket. Jika Anda melihat lonjakan kehilangan paket, laporkan ke tim dukungan TI Anda. • Grafik Jitter Buffer & RTT. Nilai waktu pulang pergi (RTT) di atas 300 akan memengaruhi pengalaman panggilan. Laporkan ini ke tim dukungan TI Anda.	

Optimalkan pengaturan browser

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kembalikan pengaturan WebRTC default.	WebRTC harus diaktifkan untuk melakukan panggilan telepon lunak dengan PKC. Kami menyarankan agar Anda menyimpan pengaturan default untuk fitur terkait WebRTC. • Di Chrome, Anda dapat mengatur bendera dengan menavigasi ke URL chrome: //flags. Ketik WebRTC di kotak pencarian untuk menemukan	Administrator

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	pengaturan yang dapat mengganggu CCP, dan atur ini ke Default. • Di Firefox, ketik about:config di address bar, lalu ketik WebRTC di kotak pencarian pada halaman konfigurasi. Pengaturan non-default muncul dalam teks tebal dan dapat diubah menjadi Default.	
Nonaktifkan ekstensi browser saat pemecahan masalah.	Beberapa ekstensi browser mungkin memengaruhi kualitas panggilan atau bahkan mencegah panggilan terhubung dengan benar. Gunakan jendela penyamaran atau mode pribadi di browser Anda, dan nonaktifkan semua ekstensi. Jika itu memecahkan masalah, tinjau ekstensi browser Anda dan cari add-on yang mencurigakan, atau nonaktifkan satu per satu.	Agen, Administrator
Periksa tingkat sampel browser.	Konfirmasikan bahwa input mikrofon Anda disetel ke laju sampel 48 kHz yang optimal. Untuk petunjuknya, lihat Panduan Administrator Amazon Connect.	Agen, Administrator

Sumber daya terkait

Jika Anda telah mengikuti langkah-langkah dalam pola ini tetapi Anda masih mengalami masalah dengan kualitas panggilan, lihat sumber daya berikut untuk tips pemecahan masalah.

- Tinjau masalah Contact Control Panel (CCP) yang umum.
- Periksa koneksi dengan Utilitas Uji Titik Akhir.
- Ikuti panduan pemecahan masalah untuk masalah lainnya.

Jika pemecahan masalah dan penyesuaian Anda tidak menyelesaikan masalah kualitas panggilan, akar penyebabnya mungkin berada di luar stasiun kerja Anda. Untuk pemecahan masalah lebih lanjut, hubungi tim dukungan TI Anda.

Lebih banyak pola

- Menguraikan monolit menjadi layanan mikro dengan menggunakan CQRS dan sumber acara
- Terapkan ChatOps solusi untuk mengelola hasil pemindaian SAST dengan menggunakan Amazon Q Developer dalam tindakan kustom aplikasi obrolan dan CloudFormation
- Integrasikan Amazon API Gateway dengan Amazon SQS untuk menangani REST asinkron APIs
- Merampingkan pengembangan dan penerapan bot Amazon Lex dengan menggunakan alur kerja otomatis

Strategi multi-akun

Topik

- Migrasi akun AWS anggota dari ke AWS Organizations AWS Control Tower
- Menyiapkan peringatan untuk penutupan akun terprogram di AWS Organizations
- Lebih banyak pola

Migrasi akun AWS anggota dari ke AWS Organizations AWS Control Tower

Rodolfo Jr. Cerrada, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memigrasikan Akun AWS dari AWS Organizations, di mana itu adalah akun anggota yang diatur oleh akun manajemen, ke. AWS Control Tower Dengan mendaftarkan akun AWS Control Tower, Anda dapat memanfaatkan kontrol dan fitur preventif dan detektif yang merampingkan tata kelola akun Anda. Anda mungkin juga ingin memigrasikan akun anggota Anda jika akun manajemen AWS Organizations Anda telah disusupi, dan Anda ingin memindahkan akun anggota ke organisasi baru yang diatur oleh. AWS Control TowerAWS Control Tower

AWS Control Tower menyediakan kerangka kerja yang menggabungkan dan mengintegrasikan kemampuan beberapa lainnya Layanan AWS, termasuk AWS Organizations, dan memastikan kepatuhan dan tata kelola yang konsisten di seluruh lingkungan multi-akun Anda. Dengan AWS Control Tower, Anda dapat mengikuti seperangkat aturan dan definisi yang ditentukan yang memperluas kemampuan AWS Organizations. Misalnya, Anda dapat menggunakan kontrol untuk memastikan bahwa log keamanan dan izin akses lintas akun yang diperlukan dibuat, dan tidak diubah.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Control Tower disiapkan di organisasi target Anda di AWS Organizations (untuk petunjuk, lihat Menyiapkan dalam AWS Control Tower dokumentasi)
- Kredensi administrator untuk AWS Control Tower (anggota grup) AWSControlTowerAdmins
- Kredensi administrator untuk sumbernya Akun AWS

Batasan

- Akun manajemen sumber di AWS Organizations harus berbeda dari akun manajemen target di AWS Control Tower.

Versi produk

- AWS Control Tower versi 2.3 (Februari 2020) atau lebih baru (lihat catatan rilis)

Arsitektur

Diagram berikut menggambarkan proses migrasi dan arsitektur referensi. Pola ini memigrasikan Akun AWS dari organisasi sumber ke organisasi target yang diatur oleh. AWS Control Tower

Proses pendaftaran terdiri dari langkah-langkah berikut:

1. Organisasi target mengirimkan undangan untuk akun untuk bergabung dengan organisasi.
2. Akun menerima undangan dan menjadi anggota organisasi target.
3. Akun terdaftar AWS Control Tower dan dipindahkan ke unit organisasi terdaftar (OU). (Kami menyarankan Anda memeriksa AWS Control Tower dasbor untuk mengonfirmasi pendaftaran.) Pada titik ini, semua kontrol yang diaktifkan di OU terdaftar berlaku.

Alat

Layanan AWS

- AWS Organizationsadalah layanan manajemen akun yang memungkinkan Anda untuk mengkonsolidasikan beberapa Akun AWS ke dalam satu entitas (organisasi) yang Anda buat dan kelola secara terpusat.
- AWS Control Towermengintegrasikan kemampuan layanan lain, termasuk,, dan AWS Organizations AWS IAM Identity Center AWS Service Catalog, untuk membantu Anda menegakkan dan mengelola aturan tata kelola untuk keamanan, operasi, dan kepatuhan dalam skala besar di semua organisasi dan akun Anda di. AWS Cloud

Epik

Undang akun untuk bergabung dengan organisasi baru AWS Control Tower

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke AWS Control Tower.	Masuk ke AWS Control Tower konsol sebagai administrator.	Administrator AWS Control Tower

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Saat ini, tidak ada cara langsung untuk memindahkan Akun AWS dari organisasi sumber ke organisasi dalam OU yang diatur oleh AWS Control Tower. Namun, Anda dapat memperluas AWS Control Tower tata kelola ke yang sudah ada Akun AWS saat Anda mendaftarkannya ke OU yang sudah diatur oleh. AWS Control Tower Itu sebabnya Anda harus masuk AWS Control Tower untuk langkah ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Undang akun anggota.	1. Masuk ke AWS Organizations konsol, dan arahkan ke Akun AWShalaman. 2. Pada Tambahkan Akun AWS halaman, pilih Undang yang sudah ada Akun AWS. 3. Lengkapi informasi akun, termasuk nomor akun 12 digit (tanpa tanda hubung) dan deskripsi dan tag opsional, lalu pilih Kirim undangan. **Important** Verifikasi bahwa tidak ada aplikasi atau konektivitas jaringan yang akan terpengaruh oleh transfer akun. Tindakan ini mengirimkan email undangan dengan tautan ke akun anggota. Ketika administrator akun mengikuti tautan dan menerima undangan, akun anggota muncul di Akun AWShalaman. Untuk informasi selengkapnya, lihat Mengelola undangan akun dalam	Administrator AWS Control Tower

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dokumentasi. AWS Organizations	
Uji aplikasi dan konektivitas.	Ketika akun anggota telah terdaftar ke organisasi baru, itu muncul di OU dalam root. Itu juga muncul di AWS Control Tower konsol, ditandai sebagai tidak terdaftar di akun, karena belum terdaftar di OU terdaftar. AWS Control Tower Verifikasi hal berikut: • Periksa AWS Control Tower dasbor untuk melihat apakah ada pelanggaran pagar pembatas. • Periksa konektivitas jaringan (VPN atau AWS Direct Connect) untuk memastikan tidak terpengaruh oleh transfer. • (Pemilik aplikasi) Uji aplikasi yang terkait dengan akun ini untuk memverifikasi bahwa mereka berjalan seperti yang diharapkan, dan dependensi tidak terpengaruh oleh transfer akun.	Administrator AWS Control Tower, Administrator akun anggota, Pemilik aplikasi

Siapkan akun untuk pendaftaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau kontrol dan perbaiki pelanggaran apa pun.	Tinjau kontrol yang ditentukan dalam OU target, terutama kontrol pencegahan, dan perbaiki pelanggaran apa pun. Sejumlah kontrol preventif wajib diaktifkan secara default saat Anda mengatur AWS Control Tower landing zone. Ini tidak dapat dinonaktifkan. Anda harus meninjau kontrol wajib ini dan memperbaiki akun anggota (secara manual atau dengan menggunakan skrip) sebelum Anda mendaftarkan akun. **Note** Kontrol preventif menjaga agar akun AWS Control Tower terdaftar tetap patuh dan mencegah pelanggaran kebijakan. Setiap pelanggaran kontrol pencegahan dapat mempengaruhi pendaftaran. Pelanggaran kontrol detektif muncul di AWS Control Tower	Administrator AWS Control Tower, Administrator akun anggota

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dasbor, jika terdeteksi, setelah pendaftaran berhasil. Mereka tidak mempengaruhi proses pendaftaran. Untuk informasi selengkapnya, lihat Tentang kontrol dalam AWS Control Tower dokumentasi.	
Periksa masalah konektivitas setelah memperbaiki pelanggaran kontrol.	Dalam beberapa kasus, Anda mungkin harus menutup port tertentu atau menonaktifkan layanan untuk memperbaiki pelanggaran kontrol. Pastikan aplikasi yang menggunakan port dan layanan tersebut diperbaiki sebelum Anda mendaftarkan akun.	Pemilik aplikasi

Daftarkan akun ke AWS Control Tower

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Masuk ke AWS Control Tower.	Masuk ke konsol AWS Control Tower tersebut. Gunakan kredenal masuk yang memiliki izin administratif untuk. AWS Control Tower Jangan gunakan kredensi pengguna root (akun manajemen) untuk mendaftarkan akun. AWS Organizations Ini	Administrator AWS Control Tower

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akan menampilkan pesan kesalahan.	
Daftarkan akun.	1. Dari halaman Account Factory di AWS Control Tower, pilih Daftarkan akun. 2. Isi detailnya, termasuk alamat email yang terkait dengan akun yang ingin Anda daftarkan, nama tampilan yang akan muncul AWS Control Tower, alamat email Pusat Identitas IAM, nama depan dan belakang pemilik akun, dan OU tempat Anda ingin mendaftarkan akun. Alamat email IAM Identity Center adalah alamat email pengguna pilihan Anda. Anda dapat menggunakan alamat email yang sama dengan email akun. 3. Pilih Daftarkan akun. Untuk informasi selengkapnya, lihat Tentang mendaftarkan akun yang ada di AWS Control Tower dokumentasi.	Administrator AWS Control Tower

Verifikasi akun setelah pendaftaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi akun.	Dari AWS Control Tower, pilih Akun. Akun yang baru saja Anda daftarkan memiliki status awal Mendaftar. Ketika pendaftaran selesai, statusnya berubah menjadi Terdaftar.	Administrator AWS Control Tower, Administrator akun anggota
Periksa pelanggaran kontrol.	Kontrol yang ditentukan dalam OU akan secara otomatis berlaku untuk akun anggota terdaftar. Pantau AWS Control Tower dasbor untuk pelanggaran dan perbaiki sesuai dengan itu. Untuk informasi selengkapnya, lihat Tentang kontrol dalam AWS Control Tower dokumentasi.	Administrator AWS Control Tower, Administrator akun anggota

Pemecahan masalah

Isu	Solusi
Anda menerima pesan galat: Terjadi kesalahan yang tidak diketahui. Coba lagi nanti, atau hubungi AWS Support.	Kesalahan ini terjadi ketika Anda menggunakan kredensi pengguna root (akun manajemen) AWS Control Tower untuk mendaftarkan akun baru. AWS Service Catalog tidak dapat memetakan Portofolio Account Factory atau produk ke pengguna root, yang menghasilkan pesan kesalahan. Untuk memperbaiki kesalahan ini, gunakan kredensil pengguna (administrator) non-root, akses penuh untuk mendaftarkan akun baru. Untuk informasi

Isu	Solusi
	selengkapnya tentang cara menetapkan akses administratif ke pengguna administratif, lihat Memulai dokumentasi Pusat Identitas IAM.
Halaman AWS Control Tower Aktivitas menampilkan aksi Get Catastrophic Drift.	Tindakan ini mencerminkan pemeriksaan drift layanan dan tidak menunjukkan masalah apa pun dengan AWS Control Tower pengaturan. Tidak ada tindakan yang diperlukan.

Sumber daya terkait

Dokumentasi

- Terminologi dan konsep (AWS Organizations dokumentasi)
- Apa itu AWS Control Tower? (AWS Control Tower dokumentasi)
- Menghapus akun anggota dari organisasi (AWS Organizations dokumentasi)
- Menyiapkan (AWS Control Tower dokumentasi)

Tutorial dan video

- AWS Control Tower lokakarya (bengkel mandiri)
- Apa itu AWS Control Tower? (video)
- Penyediaan Pengguna di AWS Control Tower(video)

Menyiapkan peringatan untuk penutupan akun terprogram di AWS Organizations

Richard Milner-Watts, Debojit Bhadra, dan Manav Yadav, Amazon Web Services

Ringkasan

CloseAccount API untuk AWS Organizations memungkinkan Anda menutup akun anggota dalam organisasi secara terprogram, tanpa harus masuk ke akun dengan kredensyal root. RemoveAccountFromOrganization API menarik akun keluar dari organisasi di AWS Organizations, sehingga menjadi akun mandiri.

Ini APIs berpotensi meningkatkan jumlah operator yang dapat menutup atau menghapus akun AWS. Semua pengguna yang memiliki akses ke organisasi melalui AWS Identity and Access Management (IAM) di akun manajemen AWS Organizations dapat memanggil ini APIs, jadi akses tidak terbatas pada pemilik email root akun dengan perangkat autentikasi multi-faktor (MFA) terkait.

Pola ini menerapkan peringatan ketika `CloseAccount` dan `RemoveAccountFromOrganization` APIs dipanggil, sehingga Anda dapat memantau kegiatan ini. Untuk peringatan, ia menggunakan topik Amazon Simple Notification Service (Amazon SNS). Anda juga dapat mengatur notifikasi Slack melalui webhook.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Organisasi di AWS Organizations
- Akses ke akun manajemen organisasi, di bawah akar organisasi, untuk membuat sumber daya yang diperlukan

Batasan

- Seperti yang dijelaskan dalam referensi AWS Organizations `CloseAccount` API, API hanya mengizinkan 10 persen akun anggota aktif ditutup dalam periode 30 hari bergulir.
- Ketika akun AWS ditutup, statusnya diubah menjadi SUSPEND. Selama 90 hari setelah transisi status ini, AWS Support dapat membuka kembali akun. Setelah 90 hari akun dihapus secara permanen.

- Pengguna yang memiliki akses ke akun manajemen AWS Organizations dan APIs mungkin juga memiliki izin untuk menonaktifkan peringatan ini. Jika perhatian utama adalah perilaku jahat alih-alih penghapusan yang tidak disengaja, pertimbangkan untuk melindungi sumber daya yang dibuat oleh pola ini dengan batas izin IAM.
- API memanggil `CloseAccount` dan `RemoveAccountFromOrganization` diproses di Wilayah AS Timur (Virginia N.) (`us-east-1`). Oleh karena itu, Anda harus menggunakan solusi ini untuk mengamati peristiwa. `us-east-1`

Arsitektur

Tumpukan teknologi target

- AWS Organizations
- AWS CloudTrail
- Amazon EventBridge
- AWS Lambda
- Amazon SNS

Arsitektur target

Diagram berikut menunjukkan arsitektur solusi untuk pola ini.

1. AWS Organizations memproses `RemoveAccountFromOrganization` permintaan `CloseAccount` atau permintaan.
2. Amazon EventBridge terintegrasi dengan AWS CloudTrail untuk mengirimkan peristiwa ini ke bus acara default.
3. EventBridge Aturan Amazon khusus cocok dengan permintaan AWS Organizations dan memanggil fungsi AWS Lambda.
4. Fungsi Lambda mengirimkan pesan ke topik SNS, yang pengguna dapat berlangganan untuk peringatan email atau pemrosesan lebih lanjut.
5. Jika pemberitahuan Slack diaktifkan, fungsi Lambda mengirimkan pesan ke webhook Slack.

Alat

Layanan AWS

- AWS CloudFormation menyediakan cara untuk memodelkan kumpulan AWS terkait dan sumber daya pihak ketiga, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya, dengan memperlakukan infrastruktur sebagai kode.
- Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge menerima peristiwa, indikator perubahan lingkungan, dan menerapkan aturan untuk merutekan acara ke target. Aturan mencocokkan peristiwa dengan target berdasarkan struktur peristiwa, yang disebut pola peristiwa, atau sesuai jadwal.
- AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya bila diperlukan dan menskalakan secara otomatis, dari beberapa permintaan setiap hari hingga ribuan setiap detik. Anda hanya membayar untuk waktu komputasi yang Anda konsumsi. Tidak ada biaya ketika kode Anda tidak berjalan.
- AWS Organizations membantu Anda mengelola dan mengatur lingkungan secara terpusat saat Anda menumbuhkan dan menskalakan sumber daya AWS Anda. Dengan AWS Organizations, Anda dapat membuat akun AWS baru secara terprogram dan mengalokasikan sumber daya, mengelompokkan akun untuk mengatur alur kerja Anda, menerapkan kebijakan ke akun atau grup untuk tata kelola, dan menyederhanakan penagihan dengan menggunakan satu metode pembayaran untuk semua akun Anda.
- AWS CloudTrail memantau dan mencatat aktivitas akun di seluruh infrastruktur AWS Anda, dan memberi Anda kendali atas tindakan penyimpanan, analisis, dan remediasi.
- Amazon Simple Notification Service (Amazon SNS) adalah layanan pesan yang dikelola sepenuhnya untuk komunikasi (A2A) application-to-application dan (A2P) application-to-person.

Alat-alat lainnya

- AWS Lambda Powertools untuk pustaka Python adalah seperangkat utilitas yang menyediakan fitur penelusuran, pencatatan, metrik, dan penanganan peristiwa untuk fungsi Lambda.

Kode

Kode untuk pola ini terletak di repositori GitHub AWS Account Closer Notifier.

Solusinya mencakup CloudFormation template yang menyebarkan arsitektur untuk pola ini. Ini menggunakan AWS Lambda Powertools untuk pustaka Python untuk menyediakan pencatatan dan penelusuran.

Epik

Menyebarkan arsitektur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan CloudFormation template untuk tumpukan solusi.	CloudFormation Template untuk pola ini ada di cabang utama GitHub repositori. Ini menyebarkan peran IAM, EventBridge aturan, fungsi Lambda, dan topik SNS. Untuk meluncurkan template: 1. Kloning GitHub repositori untuk mendapatkan salinan kode solusi. 2. Buka AWS Managemen t Console untuk akun manajemen AWS Organizat ions. 3. Pilih Wilayah AS Timur (Virginia Utara) (`us-east-1`), lalu buka CloudFormation konsol. 4. Buat tumpukan dengan menggunakan `account-c losure-notifier.ym l` template dan menentuka n nilai-nilai berikut:	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Nama tumpukan: `aws-account-closure-notifier-stack` • `ResourceP refix` parameter: `aws-account-closure-notifier` • `SlackNoti fication` parameter : Jika pemberitahuan Slack diperlukan, ubah pengaturan ini menjadi`true`. • `SlackWebhookEndpoi nt` parameter: Jika pemberitahuan Slack diperlukan, tentukan URL webhook. Untuk informasi selengkap nya tentang meluncurkan CloudFormation tumpukan, lihat dokumentasi AWS.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa solusi telah berhasil diluncurkan.	1. Tunggu CloudFormation tumpukan mencapai status CREATE_COMPLETE. 2. Buka EventBridge konsol dius-east-1 . 3. Verifikasi bahwa aturan baru telah dibuat dengan nama tersebutaws-account-closure-notifier-event-rule .	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berlangganan topik SNS.	(Opsional) Jika Anda ingin berlangganan topik SNS: 1. Buka konsol Amazon SNS di `us-east-1` , dan temukan topik bernama. `aws-account-closure-notifier-sns-topic` 2. Pilih nama topik, lalu pilih Buat langganan. 3. Untuk Protokol, pilih Email. 4. Untuk Endpoint, tentukan alamat email yang akan menerima notifikasi, lalu pilih Buat langganan. 5. Periksa kotak masuk email Anda untuk mengetahui pesan dari AWS Notifications. Gunakan tautan di email ini untuk mengonfirmasi langganan. Untuk informasi selengkapnya tentang menyiapkan notifikasi SNS, lihat dokumentasi Amazon SNS.	Administrator AWS

Verifikasi solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim acara pengujian ke bus acara default.	GitHub Repositori menyediakan contoh peristiwa yang dapat Anda kirim ke bus acara EventBridge default untuk pengujian. EventBridge Aturan juga bereaksi terhadap peristiwa yang menggunakan sumber `account.closure.notifier` peristiwa khusus. **Note** Anda tidak dapat menggunakan sumber CloudTrail acara untuk mengirim acara ini, karena tidak mungkin mengirim acara sebagai layanan AWS. Untuk mengirim acara pengujian: 1. Buka EventBridge konsol di `us-east-1` . 2. Di panel navigasi, di bawah Bus, pilih bus acara, lalu pilih bus acara default. 3. Pilih Kirim acara.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Untuk sumber Acara, masukkan `account.closure.notifier` . 5. Untuk tipe Detail, masukkan `AWS API Call via CloudTrail` . 6. Untuk detail Acara, salin dan tempel isi `tests/dummy-event.json` dari GitHub repositori ke dalam kotak teks. 7. Pilih Kirim untuk memulai alur kerja notifikasi.	
Verifikasi bahwa notifikasi email telah diterima.	Periksa kotak pesan yang berlangganan topik SNS untuk pemberitahuan. Anda harus menerima email dengan rincian akun yang ditutup dan prinsipal yang melakukan panggilan API.	Administrator AWS
Verifikasi bahwa pemberitahuan Slack telah diterima.	(Opsional) Jika Anda menentukan URL webhook untuk `SlackWebhookEndpoint` parameter saat Anda menerapkan CloudFormation template, periksa saluran Slack yang dipetakan ke webhook. Ini harus menampilkan pesan dengan detail akun yang ditutup dan prinsipal yang melakukan panggilan API.	Administrator AWS

Sumber daya terkait

- CloseAccount tindakan (referensi AWS Organizations API)
- RemoveAccountFromOrganization tindakan (referensi AWS Organizations API)
- AWS Lambda Powertools untuk Python

Lebih banyak pola

- Mengotomatiskan pembuatan akun dengan menggunakan Akselerator Zona Landing di AWS
- Mengotomatiskan penghapusan AWS CloudFormation tumpukan dan sumber daya terkait
- Otomatiskan manajemen pipeline dinamis untuk menerapkan solusi perbaikan terbaru di lingkungan Gitflow dengan menggunakan dan AWS Service Catalog AWS CodePipeline
- Membangun mesh data perusahaan dengan Amazon DataZone, AWS CDK, dan AWS CloudFormation
- Memusatkan pemantauan dengan menggunakan Amazon CloudWatch Observability Access Manager
- Mengatur set izin untuk beberapa akun dengan menggunakan Account Factory untuk Terraform
- Menerapkan strategi percabangan Gitflow untuk lingkungan multi-akun DevOps
- Menerapkan strategi percabangan GitHub Flow untuk lingkungan multi-akun DevOps
- Menerapkan strategi percabangan Trunk untuk lingkungan multi-akun DevOps
- Kelola set AWS izin secara dinamis dengan menggunakan Terraform
- Buat arsitektur IPAM Multi-wilayah yang hierarkis dengan menggunakan Terraform AWS
- Mengatur deteksi CloudFormation drift di organisasi multi-wilayah dan multi-akun
- Mengatur resolusi DNS untuk jaringan hybrid di lingkungan multi-akun AWS

Keamanan, identitas & kepatuhan

Topik

- Otomatiskan respons insiden dan forensik
- Secara otomatis mengaudit grup AWS keamanan yang memungkinkan akses dari alamat IP publik
- Secara otomatis memulihkan instans dan klaster Amazon RDS DB yang tidak terenkripsi
- Secara otomatis memvalidasi dan menerapkan kebijakan dan peran IAM dengan menggunakan CodePipeline, IAM Access Analyzer, dan makro AWS CloudFormation
- Integrasi dua arah AWS Security Hub CSPM dengan perangkat lunak Jira
- Buat pipeline untuk gambar kontainer yang diperkeras menggunakan EC2 Image Builder dan Terraform
- Memusatkan manajemen kunci akses IAM di AWS Organizations dengan menggunakan Terraform
- Periksa CloudFront distribusi Amazon untuk log akses, HTTPS, dan versi TLS
- Pilih alur autentikasi Amazon Cognito untuk aplikasi perusahaan
- Buat aturan AWS Config khusus dengan menggunakan AWS CloudFormation Guard kebijakan
- Buat laporan konsolidasi temuan keamanan Prowler dari beberapa Akun AWS
- Menyebarkan dan mengelola AWS Control Tower kontrol dengan menggunakan AWS CDK dan CloudFormation
- Terapkan dan kelola AWS Control Tower kontrol dengan menggunakan Terraform
- Terapkan Otomatisasi Keamanan untuk AWS WAF solusi dengan menggunakan Terraform
- Menerapkan pipeline yang secara bersamaan mendeteksi masalah keamanan di beberapa kiriman kode
- Menyebarkan kontrol akses berbasis atribut detektif untuk subnet publik dengan menggunakan AWS Config
- Menyebarkan kontrol akses berbasis atribut preventif untuk subnet publik
- Mendeteksi instans database Amazon RDS dan Aurora yang memiliki sertifikat CA kedaluwarsa
- Buat kebijakan IAM secara dinamis dengan IAM Access Analyzer dengan menggunakan Step Functions
- Aktifkan Amazon GuardDuty secara kondisional dengan menggunakan templat AWS CloudFormation
- Aktifkan enkripsi data transparan di Amazon RDS for SQL Server

- Memantau dan memulihkan penghapusan kunci AWS KMS yang dijadwalkan
- Identifikasi bucket Amazon S3 publik dengan menggunakan Security Hub AWS Organizations CSPM
- Menyerap dan menganalisis log keamanan AWS di Microsoft Sentinel
- Mengelola AWS Organizations kebijakan sebagai kode dengan menggunakan AWS CodePipeline dan Amazon Bedrock
- Kelola set AWS IAM Identity Center izin sebagai kode dengan menggunakan AWS CodePipeline
- Kelola kredensil menggunakan AWS Secrets Manager
- Pantau ElastiCache kluster Amazon untuk enkripsi saat istirahat
- Pantau aktivitas pengguna root IAM
- Kirim pemberitahuan saat pengguna IAM dibuat
- Mencegah akses internet di tingkat akun dengan menggunakan kebijakan kontrol layanan
- Mengekspor laporan identitas AWS IAM Identity Center dan tugasnya dengan menggunakan PowerShell
- Membatasi akses berdasarkan alamat IP atau geolokasi dengan menggunakan AWS WAF
- Pindai repositori Git untuk informasi sensitif dan masalah keamanan dengan menggunakan git-secret
- Amankan transfer file dengan menggunakan Transfer Family, Amazon Cognito, dan GuardDuty
- Amankan data sensitif di CloudWatch Log dengan menggunakan Amazon Macie
- Kirim peringatan dari AWS Network Firewall ke saluran Slack
- Kirim atribut khusus ke Amazon Cognito dan masukkan ke dalam token
- Sederhanakan manajemen sertifikat pribadi dengan menggunakan AWS Private CA dan AWS RAM
- Merampingkan manajemen EC2 kepatuhan Amazon dengan agen Amazon Bedrock dan AWS Config
- Perbarui kredensyal AWS CLI dari AWS IAM Identity Center dengan menggunakan PowerShell
- Gunakan Network Firewall untuk menangkap nama domain DNS dari Indikasi Nama Server untuk lalu lintas keluar
- Gunakan Terraform untuk mengaktifkan Amazon secara otomatis GuardDuty untuk organisasi
- Verifikasi praktik terbaik operasional untuk PCI DSS 4.0 dengan menggunakan AWS Config
- Lebih banyak pola

Otomatiskan respons insiden dan forensik

Lucas Kauffman dan Tomek Jakubowski, Amazon Web Services

Ringkasan

Pola ini menyebarkan serangkaian proses yang menggunakan AWS Lambda fungsi untuk menyediakan hal-hal berikut:

- Cara untuk memulai proses respons insiden dengan pengetahuan minimum
- Proses otomatis dan berulang yang selaras dengan Panduan Respons Insiden AWS Keamanan
- Pemisahan akun untuk mengoperasikan langkah-langkah otomatisasi, menyimpan artefak, dan menciptakan lingkungan forensik

Kerangka Respon Insiden Otomatis dan Forensik mengikuti proses forensik digital standar yang terdiri dari tahapan berikut:

1. Penahanan
2. Akuisisi
3. Pemeriksaan
4. Analisis

Anda dapat melakukan investigasi pada data statis (misalnya, memori yang diperoleh atau gambar disk) dan pada data dinamis yang hidup tetapi pada sistem yang terpisah.

Untuk detail selengkapnya, lihat bagian Informasi tambahan.

Prasyarat dan batasan

Prasyarat

- Dua Akun AWS:
  - Akun keamanan, yang bisa menjadi akun yang sudah ada, tetapi lebih disukai baru
  - Akun forensik, sebaiknya yang baru
- AWS Organizations mengatur
- Di akun anggota Organizations:

  - Peran Amazon Elastic Compute Cloud (Amazon EC2) harus memiliki akses Dapatkan dan Daftar ke Amazon Simple Storage Service (Amazon S3) dan dapat diakses oleh. AWS Systems Manager Kami merekomendasikan menggunakan peran yang `AmazonSSMManagedInstanceCore` AWS dikelola. Perhatikan bahwa peran ini akan secara otomatis dilampirkan ke EC2 instans Amazon saat respons insiden dimulai. Setelah respons selesai, AWS Identity and Access Management (IAM) akan menghapus semua hak atas instance.
  - Titik akhir virtual private cloud (VPC) di akun AWS anggota dan dalam Respons dan Analisis Insiden. VPCs Titik akhir tersebut adalah: S3 Gateway, EC2 Messages, SSM, dan SSM Messages.
- AWS Command Line Interface (AWS CLI) diinstal pada EC2 instans Amazon. Jika EC2 instans Amazon belum AWS CLI diinstal, akses internet akan diperlukan agar snapshot disk dan akuisisi memori berfungsi. Dalam hal ini, skrip akan menjangkau internet untuk mengunduh file AWS CLI instalasi dan akan menginstalnya pada instance.

Batasan

- Kerangka kerja ini tidak bermaksud menghasilkan artefak yang dapat dianggap sebagai bukti elektronik, dapat diajukan di pengadilan.
- Saat ini, pola ini hanya mendukung instance berbasis Linux yang berjalan pada arsitektur x86.

Arsitektur

Arsitektur target

Selain akun anggota, lingkungan target terdiri dari dua akun utama: akun Keamanan dan akun Forensik. Dua akun digunakan karena alasan berikut:

- Untuk memisahkannya dari akun pelanggan lain untuk mengurangi radius ledakan jika terjadi analisis forensik yang gagal
- Untuk membantu memastikan isolasi dan perlindungan integritas artefak yang sedang dianalisis
- Untuk menjaga kerahasiaan investigasi
- Untuk menghindari situasi di mana pelaku ancaman mungkin telah menggunakan semua sumber daya yang segera tersedia untuk Anda yang disusupi Akun AWS dengan menekan kuota layanan dan dengan demikian mencegah Anda membuat instance EC2 Amazon untuk melakukan penyelidikan.

Selain itu, memiliki akun Keamanan dan Forensik yang terpisah memungkinkan untuk membuat peran terpisah — Responder untuk memperoleh bukti dan Investigator untuk menganalisisnya. Setiap peran akan memiliki akses ke akun terpisah.

Diagram berikut hanya menunjukkan interaksi antara akun. Rincian setiap akun ditampilkan dalam diagram berikutnya, dan diagram lengkap dilampirkan.

Diagram berikut menunjukkan akun anggota.

1. Sebuah acara dikirim ke topik Slack Amazon Simple Notification Service (Amazon SNS).

Diagram berikut menunjukkan akun Keamanan.

2. Topik Amazon SNS di akun Keamanan memulai peristiwa Forensik.

Diagram berikut menunjukkan akun Forensik.

Akun Keamanan adalah tempat dua AWS Step Functions alur kerja utama dibuat untuk memori dan akuisisi gambar disk. Setelah alur kerja berjalan, mereka mengakses akun anggota yang memiliki EC2 instans Amazon yang terlibat dalam insiden, dan mereka memulai serangkaian fungsi Lambda yang akan mengumpulkan dump memori atau dump disk. Artefak tersebut kemudian disimpan di akun Forensik.

Akun Forensik akan menyimpan artefak yang dikumpulkan oleh alur kerja Step Functions di bucket Analysis artefak Amazon S3. Akun Forensik juga akan memiliki pipeline Amazon EC2 Image Builder yang membangun Amazon Machine Image (AMI) dari instance Forensik. Saat ini, gambar didasarkan pada SANS SIFT Workstation.

Proses build menggunakan Maintenance VPC, yang memiliki konektivitas ke internet. Gambar tersebut nantinya dapat digunakan untuk memutar EC2 instance Amazon untuk analisis artefak yang dikumpulkan di VPC Analisis.

VPC Analisis tidak memiliki konektivitas internet. Secara default, pola menciptakan tiga subnet analisis pribadi. Anda dapat membuat hingga 200 subnet, yang merupakan kuota untuk jumlah subnet dalam VPC, tetapi titik akhir VPC perlu menambahkan subnet tersebut untuk mengotomatiskan perintah yang sedang berjalan di dalamnya. AWS Systems Manager Session Manager

Dari perspektif praktik terbaik, kami sarankan untuk menggunakan AWS CloudTrail dan AWS Config melakukan hal berikut:

- Lacak perubahan yang dibuat di akun Forensik Anda
- Memantau akses dan integritas artefak yang disimpan dan dianalisis

Alur kerja

Diagram berikut menunjukkan langkah-langkah kunci dari alur kerja yang mencakup proses dan pohon keputusan dari saat sebuah instance dikompromikan hingga dianalisis dan terkandung.

1. Apakah `SecurityIncidentStatus` tag telah disetel dengan nilai Analyze? Jika ya, lakukan hal berikut:
   a. Lampirkan profil IAM yang benar untuk AWS Systems Manager dan Amazon S3.
   b. Kirim pesan Amazon SNS ke antrian Amazon SNS di Slack.
   c. Kirim pesan Amazon SNS ke antrian. `SecurityIncident`
   d. Memanggil mesin status Memori dan Disk Acquisition.
2. Apakah memori dan disk telah diperoleh? Jika tidak, ada kesalahan.
3. Tandai EC2 instance Amazon dengan `Contain` tag.
4. Lampirkan peran IAM dan grup keamanan untuk sepenuhnya mengisolasi instance.

Otomatisasi dan skala

Maksud dari pola ini adalah untuk memberikan solusi yang dapat diskalakan untuk melakukan respons insiden dan forensik di beberapa akun dalam satu organisasi. AWS Organizations

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS Command Line Interface (AWS CLI) adalah alat open source untuk berinteraksi dengan Layanan AWS melalui perintah di shell baris perintah Anda.

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Security Hub CSPMmemberikan pandangan komprehensif tentang keadaan keamanan Anda di AWS. Ini juga membantu Anda memeriksa AWS lingkungan Anda terhadap standar industri keamanan dan praktik terbaik.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar.

Kode

Untuk kode dan panduan implementasi dan penggunaan spesifik, lihat repositori GitHub Automated Incident Response and Forensics Framework.

Epik

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	CloudFormation Template ditandai 1 hingga 7 dengan	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kata pertama dari nama skrip yang menunjukkan di akun mana template perlu digunakan. Perhatikan bahwa urutan peluncuran CloudFormation template itu penting. • `1-forensic-AnalysisVPCnS3Buckets.yaml` : Diterapkan di akun forensik. Ini menciptakan bucket Amazon S3 dan VPC Analisis, dan mengaktifkan. CloudTrail • `2-forensic-MaintenanceVPCnEC2ImageBuilderPipeline.yaml` : Menyebarkan pipa VPC pemeliharaan dan pembuat gambar berdasarkan SANS SIFT. • `3-security_IR-Disk_Mem_automation.yaml` : Menyebarkan fungsi di akun keamanan yang memungkinkan akuisisi disk dan memori. • `4-security_LiME_Volatility_Factory.yaml` : Memulai fungsi build untuk mulai membuat modul memori berdasarkan AMI IDs yang diberikan. Perhatikan bahwa AMI	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	IDs berbeda Wilayah AWS. Kapan pun Anda membutuhkan modul memori baru, Anda dapat menjalankan kembali skrip ini dengan AMI IDs baru. Pertimbangkan untuk mengintegrasikan ini dengan pipeline pembuat AMI gambar emas Anda (jika digunakan di lingkungan Anda). • `5-member-IR-automation.yaml` : Menciptakan fungsi otomatisasi respons insiden anggota, yang memulai proses respons insiden. Ini memungkinkan berbagi volume Amazon Elastic Block Store (Amazon EBS) di seluruh akun, memposting otomatis ke saluran Slack selama proses respons insiden, memulai proses forensik, dan mengisolasi instance setelah proses selesai. • `6-forensic-artifact-s3-policies.yaml` : Setelah semua skrip diterapkan, skrip ini memperbaiki izin yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	diperlukan untuk semua interaksi lintas akun. • `7-security-IR-vpc.yaml` : Mengonfigurasi VPC yang digunakan untuk pemrosesan volume respons insiden. Untuk memulai kerangka respons insiden untuk EC2 instance Amazon tertentu, buat tag dengan kunci `SecurityIncidentStatus` dan nilainya`Analyze`. Ini akan memulai fungsi Lambda anggota yang secara otomatis akan memulai isolasi dan memori serta akuisisi disk.	
Operasikan kerangka kerja.	Fungsi Lambda juga akan menandai ulang aset di akhir (atau saat gagal) dengan. `Contain` Ini memulai penahanan, yang sepenuhnya mengisolasi instance dengan grup tanpa INBOUND/OUTBOUND keamanan dan dengan peran IAM yang melarang semua akses. Ikuti langkah-langkah di GitHub repositori.	Administrator AWS

Menyebarkan tindakan CSPM Security Hub kustom

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menerapkan tindakan CSPM Security Hub kustom dengan menggunakan template. CloudFormation	Untuk membuat tindakan kustom sehingga Anda dapat menggunakan daftar dropdown dari Security Hub CSPM, gunakan template. `Modules/SecurityHub Custom Actions/SecurityHubCustomActions.yaml` CloudFormation Kemudian ubah `IRAutomation` peran di setiap akun anggota untuk memungkinkan fungsi Lambda yang menjalankan tindakan untuk mengambil peran. `IRAutomation` Untuk informasi lebih lanjut, lihat GitHub repositori.	Administrator AWS

Sumber daya terkait

- AWS Panduan Respons Insiden Keamanan

Informasi tambahan

Dengan menggunakan lingkungan ini, tim Security Operations Center (SOC) dapat meningkatkan proses respons insiden keamanan mereka melalui hal-hal berikut:

- Memiliki kemampuan untuk melakukan forensik di lingkungan yang terpisah untuk menghindari kompromi sumber daya produksi yang tidak disengaja
- Memiliki proses standar, berulang, otomatis untuk melakukan penahanan dan analisis.

- Memberikan pemilik akun atau administrator kemampuan untuk memulai proses respons insiden dengan pengetahuan minimal tentang cara menggunakan tag
- Memiliki lingkungan yang terstandarisasi dan bersih untuk melakukan analisis insiden dan forensik tanpa kebisingan lingkungan yang lebih besar
- Memiliki kemampuan untuk membuat beberapa lingkungan analisis secara paralel
- Memfokuskan sumber daya SOC pada respons insiden, bukan pada pemeliharaan dan dokumentasi lingkungan forensik cloud
- Beralih dari proses manual menuju proses otomatis untuk mencapai skalabilitas
- Menggunakan CloudFormation template untuk konsistensi dan untuk menghindari tugas berulang

Selain itu, Anda menghindari penggunaan infrastruktur persisten, dan Anda membayar sumber daya saat Anda membutuhkannya.

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Secara otomatis mengaudit grup AWS keamanan yang memungkinkan akses dari alamat IP publik

Eugene Shifer dan Stephen DiCato, Amazon Web Services

Ringkasan

Sebagai praktik terbaik keamanan, sangat penting untuk meminimalkan paparan AWS sumber daya hanya pada apa yang mutlak diperlukan. Misalnya, server web yang melayani masyarakat umum perlu mengizinkan akses masuk dari internet, tetapi akses ke beban kerja lain harus dibatasi pada jaringan tertentu untuk mengurangi paparan yang tidak perlu. Grup keamanan di Amazon Virtual Private Cloud (Amazon VPC) adalah kontrol yang efektif untuk membantu Anda membatasi akses sumber daya. Namun, mengevaluasi grup keamanan dapat menjadi tugas yang rumit, terutama dalam arsitektur multi-akun. AWS Config aturan dan AWS Security Hub CSPM kontrol dapat membantu Anda mengidentifikasi grup keamanan yang mengizinkan akses dari internet publik (0.0.0.0/0) ke protokol komunikasi jaringan tertentu, seperti Secure Shell (SSH), HTTP, HTTPS, dan protokol desktop jarak jauh Windows (RDP). Namun, aturan dan kontrol ini tidak berlaku jika layanan berjalan pada port non-standar atau jika akses dibatasi untuk alamat IP publik tertentu. Misalnya, ini mungkin terjadi ketika layanan web dikaitkan dengan port TCP 8443, bukan port TCP standar 443. Ini mungkin juga terjadi ketika pengembang memiliki akses ke server dari jaringan rumah mereka, seperti untuk tujuan pengujian.

Untuk mengatasinya, Anda dapat menggunakan solusi infrastruktur sebagai kode (IAc) yang disediakan dalam pola ini untuk mengidentifikasi grup keamanan yang memungkinkan akses dari alamat IP non-pribadi (RFC 1918 yang tidak sesuai) ke beban kerja apa pun di atau organisasi Anda. Akun AWS AWS AWS CloudFormationTemplate menyediakan AWS Config aturan khusus, AWS Lambdafungsi, dan izin yang diperlukan. Anda dapat menerapkannya sebagai tumpukan dalam satu akun atau sebagai tumpukan yang disetel di seluruh organisasi, dikelola melalui AWS Organizations.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Pengalaman menggunakan GitHub
- Jika Anda menerapkan ke dalam satu Akun AWS:
  - Izin untuk membuat tumpukan CloudFormation

  - AWS Config diatur di akun target
  - (Opsional) Security Hub CSPM diatur di akun target
- Jika Anda menerapkan ke dalam AWS organisasi:
  - Izin untuk membuat set CloudFormation tumpukan
  - Security Hub CSPM diatur dengan integrasi AWS Organizations
  - AWS Config atur di akun tempat Anda menerapkan solusi ini
  - Menunjuk Akun AWS untuk menjadi administrator yang didelegasikan untuk AWS Config dan Security Hub CSPM

Batasan

- Jika Anda menerapkan ke akun individual yang tidak mengaktifkan CSPM Security Hub, Anda dapat menggunakannya AWS Config untuk mengevaluasi temuan.
- Jika Anda menerapkan ke organisasi yang tidak memiliki administrator yang didelegasikan untuk AWS Config dan CSPM Security Hub, Anda harus masuk ke akun anggota individu untuk melihat temuan.
- Jika Anda menggunakannya AWS Control Tower untuk mengelola dan mengatur akun di organisasi Anda, gunakan IAc dalam pola ini dengan menggunakan Kustomisasi untuk (CFCT). AWS Control Tower Menggunakan CloudFormation konsol akan membuat penyimpangan konfigurasi dari AWS Control Tower pagar pembatas dan mengharuskan Anda mendaftarkan ulang unit organisasi () OUs atau akun terkelola.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Menyebarkan menjadi individu Akun AWS

Diagram arsitektur berikut menunjukkan penyebaran sumber AWS daya dalam satu Akun AWS. Anda menyediakan sumber daya dengan menggunakan CloudFormation templat langsung melalui CloudFormation konsol. Jika Security Hub CSPM diaktifkan, Anda dapat melihat hasilnya di salah satu AWS Config atau Security Hub CSPM. Jika Security Hub CSPM tidak diaktifkan, Anda dapat melihat hasilnya hanya di. AWS Config

Diagram menunjukkan alur kerja berikut:

1. Anda membuat CloudFormation tumpukan. Ini menyebarkan fungsi Lambda dan AWS Config aturan. Aturan dan fungsi diatur dengan izin AWS Identity and Access Management (IAM) yang diperlukan untuk mempublikasikan evaluasi sumber daya dan log AWS Config .
2. AWS Config Aturan beroperasi dalam mode evaluasi detektif dan memanggil fungsi Lambda setiap 24 jam.
3. Fungsi Lambda menilai grup keamanan dan mengirimkan pembaruan ke. AWS Config
4. Security Hub CSPM menerima semua temuan. AWS Config
5. Anda dapat melihat temuan di Security Hub CSPM atau di AWS Config, tergantung pada layanan yang telah Anda siapkan di akun.

Menyebarkan ke dalam organisasi AWS

Diagram berikut menunjukkan penyebaran pola di beberapa akun yang dikelola melalui AWS Organizations dan AWS Control Tower. Anda menyebarkan CloudFormation template melalui CFCT. Hasil penilaian terpusat di Security Hub CSPM di akun administrator yang didelegasikan. Bagian AWS CodePipeline alur kerja diagram menunjukkan langkah-langkah latar belakang yang terjadi selama penyebaran CFCT.

Diagram menunjukkan alur kerja berikut:

1. Di akun manajemen, Anda mengunggah file terkompresi (ZIP) dari template IAc ke bucket Amazon Simple Storage Service (Amazon S3) S3 yang digunakan oleh CFCT.
2. Pipeline CFCT membuka ritsleting file, menjalankan pemeriksaan cfn-nag (GitHub), dan menerapkan template sebagai kumpulan tumpukan. CloudFormation
3. Bergantung pada konfigurasi yang Anda tentukan dalam file manifes CFCT, CloudFormation StackSets menyebarkan tumpukan ke akun individual atau ditentukan. OUs Ini menyebarkan fungsi Lambda dan aturan di AWS Config akun target. Aturan dan fungsi diatur dengan izin IAM yang diperlukan untuk mempublikasikan evaluasi sumber daya dan log AWS Config .
4. AWS Config Aturan beroperasi dalam mode evaluasi detektif dan memanggil fungsi Lambda setiap 24 jam.
5. Fungsi Lambda menilai grup keamanan dan mengirimkan pembaruan ke. AWS Config
6. AWS Config meneruskan semua temuan ke Security Hub CSPM.

7. Temuan CSPM Security Hub dikumpulkan dalam akun administrator yang didelegasikan.
8. Anda dapat melihat temuan gabungan di CSPM Security Hub di akun administrator yang didelegasikan.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu. AWS Config Aturan menentukan pengaturan konfigurasi ideal Anda untuk sumber daya, dan AWS Config dapat mengevaluasi apakah AWS sumber daya Anda mematuhi ketentuan dalam aturan.
- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif. Kustomisasi untuk AWS Control Tower (CFCT) membantu Anda menyesuaikan AWS Control Tower landing zone dan tetap selaras dengan praktik terbaik. AWS Kustomisasi untuk solusi ini diimplementasikan melalui CloudFormation templat dan kebijakan kontrol AWS Organizations layanan () SCPs.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Security Hub CSPMmemberikan pandangan komprehensif tentang keadaan keamanan Anda di AWS. Ini juga membantu Anda memeriksa AWS lingkungan Anda terhadap standar industri keamanan dan praktik terbaik.

Alat lainnya

- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Deteksi grup keamanan rentan.

Praktik terbaik

Kami menyarankan Anda mematuhi praktik terbaik dalam sumber daya berikut:

- Praktik Terbaik untuk Unit Organisasi dengan AWS Organizations (AWS Cloud Operations & Migrations Blog)
- Panduan untuk Mendirikan Yayasan Awal menggunakan AWS Control Tower on AWS (Perpustakaan AWS Solusi)
- Panduan untuk membuat dan memodifikasi AWS Control Tower sumber daya (AWS Control Tower dokumentasi)
- Pertimbangan penyebaran CFCT (dokumentasi)AWS Control Tower
- Terapkan izin hak istimewa paling sedikit (dokumentasi IAM)

Epik

Tinjau CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan strategi penyebaran Anda.	Tinjau solusi dan kode untuk menentukan strategi penyebaran untuk AWS lingkungan Anda. Tentukan apakah Anda menerapkan ke dalam satu akun atau AWS organisasi.	Pemilik aplikasi, AWS Umum
Kloning repositori.	Masukkan perintah berikut untuk mengkloning repositori Deteksi grup keamanan rentan: `git clone https://github.com/aws-samp`	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`les/detect-public-security-groups.git`	
Validasi versi Python.	1. Arahkan ke direktori tingkat atas di repositori kloning: `cd detect-public-security-groups` 2. Buka Security-Group-Public-Assesment.yaml. 3. Di `SgPublicAccessCheckLambdaFunction` sumber daya, konfirmasikan bahwa versi Python kompatibel dengan target Anda. Wilayah AWS Secara default, fungsi ini menggunakan Python 3.12. Untuk informasi selengkapnya, lihat AWS Lambda menambahkan dukungan untuk Python 3.12. Jika perlu, perbarui versi Python. 4. Simpan dan tutup Security-Group-Public-Assesment.yaml.	Administrator AWS, Pengembang aplikasi

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	Menyebarkan CloudFormation template ke AWS lingkunga	Pengembang aplikasi, administrator AWS, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n Anda. Lakukan salah satu tindakan berikut: • Jika Anda menerapkan ke satu Akun AWS, ikuti petunjuk dalam Membuat tumpukan. • Jika Anda menerapkan ke organisasi yang tidak dikelola oleh AWS Control Tower, ikuti petunjuk di Buat kumpulan tumpukan. • Jika Anda menerapka n ke organisasi yang dikelola oleh AWS Control Tower, lihat petunjuk di Membangun kustomisasi Anda sendiri.	
Verifikasi penyebaran.	Di CloudFormation konsol, verifikasi bahwa kumpulan tumpukan atau tumpukan telah berhasil diterapkan.	Administrator AWS, Pemilik aplikasi

Tinjau temuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat temuan AWS Config aturan.	Di Security Hub CSPM, lakukan hal berikut untuk melihat daftar temuan individu: 1. Buka konsol CSPM Security Hub.	Administrator AWS, administr ator sistem AWS, administr ator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di panel navigasi, pilih Temuan. 3. Di kotak Tambahkan filter, tambahkan filter berikut: • Status kepatuhan adalah `FAILED` • Judul adalah `SgPublicAccessCheck` 4. Pilih Terapkan. Di Security Hub CSPM, lakukan hal berikut untuk melihat daftar total temuan yang dikelompokkan berdasarkan: Akun AWS 1. Buka konsol CSPM Security Hub. 2. Pada panel navigasi, silakan pilih Wawasan. 3. Pilih Buat wawasan. 4. Untuk memilih atribut pengelompokan untuk wawasan: a. Pilih kotak pencarian untuk menampilkan opsi filter. b. Pilih Grup berdasarkan. c. Pilih AwsAccountId. d. Pilih Terapkan. 5. Di kotak Tambahkan filter, tambahkan filter berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Judul adalah `SgPublicAccessCheck` • Status kepatuhan adalah `FAILED` 6. Pilih Buat wawasan. 7. Masukkan nama Insight, lalu pilih Buat wawasan. Di AWS Config, untuk melihat daftar temuan, ikuti petunjuk dalam Melihat Informasi Kepatuhan dan Hasil Evaluasi dalam AWS Config dokumentasi.	

Pemecahan masalah

Isu	Solusi
Pembuatan atau penghapusan set CloudFormation tumpukan gagal.	Ketika AWS Control Tower dikerahkan, itu memberlakukan pagar pembatas yang diperlukan dan mengasumsikan kontrol atas agregator dan aturan. AWS Config Ini termasuk mencegah perubahan langsung melalui CloudFormation. Untuk menerapkan atau menghapus CloudFormation template ini dengan benar, termasuk semua sumber daya terkait, Anda harus menggunakan CFCT.
CFCT gagal menghapus CloudFormation template.	Jika CloudFormation template tetap ada bahkan setelah membuat perubahan yang diperlukan dalam file manifes dan menghapus file template, konfirmasikan bahwa file manifes

Isu	Solusi
	berisi `enable_stack_set_deletion` parameter dan nilainya disetel ke`false`. Untuk informasi selengkapnya, lihat Menghapus kumpulan tumpukan dalam dokumentasi CFCT.

Sumber daya terkait

- AWS Config Aturan Kustom (AWS Config dokumentasi)

Secara otomatis memulihkan instans dan klaster Amazon RDS DB yang tidak terenkripsi

Ajay Rawat dan Josh Joy, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara memulihkan instans dan kluster DB Amazon Relational Database Service (Amazon RDS) yang tidak terenkripsi secara otomatis di Amazon Web Services () AWS Systems Manager dengan menggunakan,AWS runbook, dan (AWS Config) key. AWS Key Management Service AWS KMS

Instans RDS DB terenkripsi memberikan lapisan perlindungan data tambahan dengan mengamankan data Anda dari akses tidak sah ke penyimpanan yang mendasarinya. Anda dapat menggunakan enkripsi Amazon RDS untuk meningkatkan perlindungan data aplikasi Anda yang digunakan di AWS Cloud, dan untuk memenuhi persyaratan kepatuhan untuk enkripsi saat istirahat. Anda dapat mengaktifkan enkripsi untuk instans RDS DB saat Anda membuatnya, tetapi tidak setelah dibuat. Namun, Anda dapat menambahkan enkripsi ke instans RDS DB yang tidak terenkripsi dengan membuat snapshot dari instans DB Anda, dan kemudian membuat salinan terenkripsi dari snapshot itu. Anda kemudian dapat memulihkan instans DB dari snapshot terenkripsi untuk mendapatkan salinan terenkripsi dari instans DB asli Anda.

Pola ini digunakan Aturan AWS Config untuk mengevaluasi instans dan cluster RDS DB. Ini menerapkan remediasi dengan menggunakan AWS Systems Manager runbook, yang menentukan tindakan yang akan dilakukan pada sumber daya Amazon RDS yang tidak sesuai, AWS KMS dan kunci untuk mengenkripsi snapshot DB. Kemudian memberlakukan kebijakan kontrol layanan (SCPs) untuk mencegah pembuatan instance dan cluster DB baru tanpa enkripsi.

Kode untuk pola ini disediakan di GitHub.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- File dari repositori kode GitHub sumber untuk pola ini diunduh ke komputer Anda
- Instans atau cluster RDS DB yang tidak terenkripsi
- AWS KMS Kunci yang ada untuk mengenkripsi instans dan cluster RDS DB

- Akses untuk memperbarui kebijakan sumber daya kunci KMS
- AWS Config diaktifkan di Anda Akun AWS (lihat Memulai dengan AWS Config dalam AWS dokumentasi)

Batasan

- Anda dapat mengaktifkan enkripsi untuk instans RDS DB hanya ketika Anda membuatnya, bukan setelah dibuat.
- Anda tidak dapat memiliki replika baca terenkripsi dari instans DB yang tidak dienkripsi atau replika baca yang tidak dienkripsi dari instans DB terenkripsi.
- Anda tidak dapat memulihkan cadangan atau snapshot yang tidak terenkripsi ke instans DB terenkripsi.
- Enkripsi Amazon RDS tersedia untuk sebagian besar kelas instans DB. Untuk daftar pengecualian, lihat Mengenkripsi sumber daya Amazon RDS di dokumentasi Amazon RDS.
- Untuk menyalin snapshot terenkripsi dari satu Wilayah AWS ke yang lain, Anda harus menentukan kunci KMS di tujuan. Wilayah AWS Ini karena kunci KMS khusus untuk tempat Wilayah AWS mereka dibuat.
- Snapshot sumber tetap terenkripsi selama proses penyalinan. Amazon RDS menggunakan enkripsi amplop untuk melindungi data selama proses penyalinan. Untuk informasi selengkapnya, lihat Enkripsi amplop dalam AWS KMS dokumentasi.
- Anda tidak dapat menghapus enkripsi instans DB terenkripsi. Namun, Anda dapat mengekspor data dari instans DB terenkripsi dan mengimpor data ke instans DB yang tidak terenkripsi.
- Anda harus menghapus kunci KMS hanya ketika Anda yakin bahwa Anda tidak perlu menggunakannya lagi. Jika Anda tidak yakin, pertimbangkan untuk menonaktifkan kunci KMS alih-alih menghapusnya. Anda dapat mengaktifkan kembali kunci KMS yang dinonaktifkan jika Anda perlu menggunakannya lagi nanti, tetapi Anda tidak dapat memulihkan kunci KMS yang dihapus.
- Jika Anda tidak memilih untuk mempertahankan cadangan otomatis, cadangan otomatis Anda yang Wilayah AWS sama dengan instans DB akan dihapus. Cadangan tersebut tidak dapat dipulihkan setelah Anda menghapus instans DB.
- Pencadangan otomatis Anda dipertahankan untuk periode retensi yang ditetapkan pada instans DB pada saat Anda menghapusnya. Periode retensi yang ditetapkan ini terjadi baik ketika Anda memilih untuk membuat snapshot DB akhir maupun tidak.
- Jika remediasi otomatis diaktifkan, solusi ini mengenkripsi semua database yang memiliki kunci KMS yang sama.

Arsitektur

Diagram berikut menggambarkan arsitektur untuk CloudFormation implementasi. Perhatikan bahwa Anda juga dapat menerapkan pola ini dengan menggunakan AWS Cloud Development Kit (AWS CDK).

Alat

Alat

- CloudFormationmembantu Anda mengatur AWS sumber daya secara otomatis. Ini memungkinkan Anda untuk menggunakan file template untuk membuat dan menghapus koleksi sumber daya bersama-sama sebagai satu unit (tumpukan).
- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak untuk mendefinisikan infrastruktur cloud Anda dalam kode dan menyediakannya dengan menggunakan bahasa pemrograman yang sudah dikenal.

Layanan dan fitur AWS

- AWS Configmelacak konfigurasi AWS sumber daya Anda dan hubungannya dengan sumber daya Anda yang lain. Itu juga dapat mengevaluasi AWS sumber daya tersebut untuk kepatuhan. Layanan ini menggunakan aturan yang dapat dikonfigurasi untuk mengevaluasi AWS sumber daya terhadap konfigurasi yang diinginkan. Anda dapat menggunakan seperangkat aturan AWS Config terkelola untuk skenario kepatuhan umum, atau Anda dapat membuat aturan sendiri untuk skenario kustom. Jika AWS sumber daya ditemukan tidak sesuai, Anda dapat menentukan tindakan remediasi melalui AWS Systems Manager runbook dan secara opsional mengirim peringatan melalui topik Amazon Simple Notification Service (Amazon SNS). Dengan kata lain, Anda dapat mengaitkan tindakan remediasi dengan Aturan AWS Config dan memilih untuk menjalankannya secara otomatis untuk mengatasi sumber daya yang tidak sesuai tanpa intervensi manual. Jika sumber daya masih tidak sesuai setelah remediasi otomatis, Anda dapat mengatur aturan untuk mencoba remediasi otomatis lagi.
- Amazon Relational Database Service (Amazon RDS) memudahkan pengaturan, pengoperasian, dan skala database relasional di cloud. Blok bangunan dasar Amazon RDS adalah instans DB, yang merupakan lingkungan database terisolasi di AWS Cloud. Amazon RDS menyediakan pilihan jenis instans yang dioptimalkan agar sesuai dengan kasus penggunaan database relasional yang berbeda. Jenis instans terdiri dari berbagai kombinasi CPU, memori, penyimpanan, dan kapasitas

jaringan dan memberi Anda fleksibilitas untuk memilih campuran sumber daya yang tepat untuk database Anda. Setiap jenis instans mencakup beberapa ukuran instans, memungkinkan Anda untuk menskalakan basis data sesuai dengan persyaratan beban kerja target Anda.

- AWS Key Management Service (AWS KMS) adalah layanan terkelola yang memudahkan Anda membuat dan mengontrol AWS KMS keys, yang mengenkripsi data Anda. Kunci KMS adalah representasi logis dari kunci root. Kunci KMS mencakup metadata, seperti ID kunci, tanggal pembuatan, deskripsi, dan status kunci.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- Kebijakan kontrol layanan (SCPs) menawarkan kontrol pusat atas izin maksimum yang tersedia untuk semua akun di organisasi Anda. SCPs membantu Anda memastikan bahwa akun Anda tetap berada dalam pedoman kontrol akses organisasi Anda. SCPs tidak memengaruhi pengguna atau peran di akun manajemen. SCP tersebut hanya mempengaruhi akun anggota di organisasi Anda. Kami sangat menyarankan agar Anda tidak melampirkan SCPs ke akar organisasi Anda tanpa menguji secara menyeluruh dampak kebijakan terhadap akun. Sebagai gantinya, buat unit organisasi (OU) yang dapat Anda pindahkan akun Anda menjadi satu per satu, atau setidaknya dalam jumlah kecil, untuk memastikan bahwa Anda tidak secara tidak sengaja mengunci pengguna dari layanan utama.

Kode

Kode sumber dan template untuk pola ini tersedia di GitHub repositori. Pola ini menyediakan dua opsi implementasi: Anda dapat menerapkan CloudFormation template untuk membuat peran remediasi yang mengenkripsi instans dan cluster RDS DB, atau menggunakan. AWS CDK Repositori memiliki folder terpisah untuk dua opsi ini.

Bagian Epics memberikan step-by-step instruksi untuk menyebarkan template. CloudFormation Jika Anda ingin menggunakan AWS CDK, ikuti instruksi dalam `README.md` file di GitHub repositori.

Praktik terbaik

- Aktifkan enkripsi data baik saat istirahat maupun dalam perjalanan.
- Aktifkan AWS Config di semua akun dan Wilayah AWS.
- Rekam perubahan konfigurasi ke semua jenis sumber daya.
- Putar kredensial IAM Anda secara rutin.

- Manfaatkan penandaan AWS Config, yang membuat lebih mudah untuk mengelola, mencari, dan memfilter sumber daya.

Epik

Membuat peran remediasi IAM dan runbook Systems Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh CloudFormation template.	Unduh `unencrypted-to-encrypted-rds.template.json` file dari GitHub repositori.	DevOps insinyur
Buat CloudFormation tumpukan.	1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol. 2. Luncurkan `unencrypted-to-encrypted-rds.template.json` template untuk membuat tumpukan baru. Untuk informasi selengkapnya tentang penerapan templat, lihat CloudFormation dokumentasi.	DevOps insinyur
Tinjau CloudFormation parameter dan nilai.	1. Tinjau detail tumpukan dan perbarui nilai berdasarkan persyaratan lingkungan Anda. 2. Pilih Buat tumpukan untuk menyebarkan template.	DevOps insinyur
Tinjau sumber daya.	Ketika tumpukan telah dibuat, statusnya berubah menjadi	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	CREATE_COMPLETE. Tinjau sumber daya yang dibuat (peran IAM, runbook Systems Manager) di konsol. CloudFormation	

Perbarui kebijakan AWS KMS utama

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui kebijakan kunci KMS Anda.	1. Pastikan bahwa alias kunci `alias/RDSEncryptionAtRestKMSAlias` ada. 2. Pernyataan kebijakan utama harus mencakup peran remediasi IAM. (Periksa sumber daya yang dibuat oleh CloudFormation template yang Anda gunakan di epik sebelumnya.) 3. Dalam kebijakan utama berikut, perbarui bagian yang dicetak tebal agar sesuai dengan akun Anda dan peran IAM yang dibuat. `{ "Sid": "Allow access through RDS for all principals in the account that`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`are authorized to use RDS", "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam:: <your-AWS-account-ID>":role/<your-IAM-remediation-role>" }, "Action": ["kms:Encrypt", "kms:Decrypt", "kms:ReEncrypt*", "kms:GenerateDataKey*", "kms:CreateGrant", "kms:ListGrants", "kms:DescribeKey"], "Resource": "*", "Condition": { "StringEquals": { "kms:ViaService": "rds.us-east-1.amazonaws.com", "kms:CallerAccount": "<your-AWS-account-ID>" } } }`	

Temukan dan pulihkan sumber daya yang tidak sesuai

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat sumber daya yang tidak sesuai.	1. Untuk melihat daftar sumber daya yang tidak sesuai, buka konsol.AWS Config 2. Di panel navigasi, pilih Aturan, lalu pilih `rds-storage-encrypted` aturan. Sumber daya yang tidak sesuai yang tercantum di AWS Config konsol adalah instance, bukan cluster. Otomatisasi remediasi mengenkripsi instance dan cluster, dan membuat instance yang baru dienkripsi atau cluster yang baru dibuat. Namun, pastikan untuk tidak secara bersamaan memulihkan beberapa instance yang termasuk dalam cluster yang sama. Sebelum Anda memulihkan instans atau volume RDS DB apa pun, pastikan instans RDS DB tidak digunakan. Konfirmasikan bahwa tidak ada operasi penulisan yang terjadi saat snapshot sedang dibuat, untuk memastika	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n bahwa snapshot berisi data asli. Pertimbangkan untuk menerapkan jendela pemeliharaan di mana remediasi akan berjalan.	
Memulihkan sumber daya yang tidak patuh.	1. Ketika Anda siap dan jendela pemeliharaan berlaku, pilih sumber daya untuk diperbaiki, dan kemudian pilih Remediate. Kolom status Action sekarang harus menampilkan eksekusi Action antrian. 2. Lihat kemajuan dan status remediasi di Systems Manager. Buka konsol System Manager. Di panel navigasi, pilih Otomasi, lalu pilih ID eksekusi otomatisasi yang sesuai untuk melihat detail lebih lanjut.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi bahwa instans RDS DB tersedia.	Setelah otomatisasi selesai, instans RDS DB yang baru dienkripsi akan tersedia. Instans RDS DB terenkripsi akan memiliki awalan `encrypted` diikuti dengan nama aslinya. Misalnya, jika nama instans RDS DB yang tidak terenkripsi adalah`database-1` , instans RDS DB yang baru dienkripsi akan menjadi. `encrypted-database-1`	DevOps insinyur
Mengakhiri instance yang tidak terenkripsi.	Setelah remediasi selesai dan sumber daya yang baru dienkripsi telah divalidasi, Anda dapat menghentikan instance yang tidak terenkripsi. Pastikan untuk mengonfirmasi bahwa sumber daya yang baru dienkripsi cocok dengan sumber daya yang tidak terenkripsi sebelum Anda menghentikan sumber daya apa pun.	DevOps insinyur

Menegakkan SCPs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menegakkan SCPs.	Menegakkan SCPs untuk mencegah instans dan cluster DB dibuat tanpa enkripsi	Insinyur keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di masa depan. Gunakan `rds_encrypted.json` file yang disediakan di GitHub repositori untuk tujuan ini, dan ikuti instruksi dalam dokumentasi.AWS	

Sumber daya terkait

Referensi

- Menyiapkan AWS Config
- AWS Config aturan kustom
- AWS KMS konsep
- AWS Systems Manager dokumen
- Kebijakan kontrol layanan

Alat

- CloudFormation
- AWS Cloud Development Kit (AWS CDK)

Panduan dan pola

- Aktifkan kembali secara otomatis AWS CloudTrail dengan menggunakan aturan remediasi khusus di AWS Config

Informasi tambahan

Bagaimana cara AWS Config kerjanya?

Saat Anda menggunakan AWS Config, pertama kali menemukan AWS sumber daya yang didukung yang ada di akun Anda dan menghasilkan item konfigurasi untuk setiap sumber daya. AWS Config juga menghasilkan item konfigurasi saat konfigurasi sumber daya berubah, dan menyimpan catatan

historis item konfigurasi sumber daya Anda sejak Anda memulai perekam konfigurasi. Secara default, AWS Config membuat item konfigurasi untuk setiap sumber daya yang didukung di file Wilayah AWS. Jika Anda tidak AWS Config ingin membuat item konfigurasi untuk semua sumber daya yang didukung, Anda dapat menentukan jenis sumber daya yang ingin dilacak.

Bagaimana AWS Config dan Aturan AWS Config terkait dengan AWS Security Hub CSPM?

AWS Security Hub CSPM adalah layanan keamanan dan kepatuhan yang menyediakan manajemen postur keamanan dan kepatuhan sebagai layanan. Ia menggunakan AWS Config dan Aturan AWS Config sebagai mekanisme utamanya untuk mengevaluasi konfigurasi sumber AWS daya. Aturan AWS Config juga dapat digunakan untuk mengevaluasi konfigurasi sumber daya secara langsung. Lainnya Layanan AWS, seperti AWS Control Tower dan AWS Firewall Manager, juga digunakan Aturan AWS Config.

Secara otomatis memvalidasi dan menerapkan kebijakan dan peran IAM dengan menggunakan CodePipeline, IAM Access Analyzer, dan makro AWS CloudFormation

Helton Ribeiro dan Guilherme Simoes, Amazon Web Services

Ringkasan

Pola ini menjelaskan langkah-langkah dan menyediakan kode untuk membuat pipeline penerapan yang memungkinkan tim pengembangan Anda membuat kebijakan dan peran AWS Identity and Access Management (IAM) di akun Amazon Web Services (AWS) Anda. Pendekatan ini membantu organisasi Anda mengurangi biaya overhead untuk tim operasional Anda dan mempercepat proses penyebaran. Ini juga membantu pengembang Anda untuk membuat peran dan kebijakan IAM yang kompatibel dengan tata kelola dan kontrol keamanan yang ada.

Pendekatan pola ini menggunakan AWS Identity and Access Management Access Analyzer untuk memvalidasi kebijakan IAM yang ingin Anda lampirkan ke peran IAM dan menggunakan AWS CloudFormation untuk menerapkan peran IAM. Namun, alih-alih mengedit file CloudFormation template AWS secara langsung, tim pengembangan Anda membuat kebijakan dan peran IAM berformat JSON. CloudFormation Makro AWS mengubah file kebijakan berformat JSON ini menjadi tipe sumber daya CloudFormation AWS IAM sebelum memulai penerapan.

Pipeline penerapan (`RolesPipeline`) memiliki tahapan sumber, validasi, dan penerapan. Selama tahap sumber, tim pengembangan Anda mendorong file JSON yang berisi definisi peran dan kebijakan IAM ke repositori AWS. CodeCommit AWS CodeBuild kemudian menjalankan skrip untuk memvalidasi file-file tersebut dan menyalinnya ke bucket Amazon Simple Storage Service (Amazon S3). Karena tim pengembangan Anda tidak memiliki akses langsung ke file CloudFormation template AWS yang disimpan dalam bucket S3 terpisah, mereka harus mengikuti proses pembuatan dan validasi file JSON.

Terakhir, selama fase penerapan, AWS CodeDeploy menggunakan CloudFormation tumpukan AWS untuk memperbarui atau menghapus kebijakan dan peran IAM dalam akun.

> **Important**
>
> Alur kerja pola ini adalah bukti konsep (POC) dan kami menyarankan Anda hanya menggunakannya di lingkungan pengujian. Jika Anda ingin menggunakan pendekatan pola

ini di lingkungan produksi, lihat Praktik terbaik keamanan di IAM dalam dokumentasi IAM dan buat perubahan yang diperlukan pada peran IAM dan layanan AWS Anda.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Bucket S3 baru atau yang sudah ada untuk `RolesPipeline` pipa. Pastikan kredenal akses yang Anda gunakan memiliki izin untuk mengunggah objek ke bucket ini.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang ini, lihat Menginstal, memperbarui, dan menghapus instalan AWS CLI dalam dokumentasi AWS CLI.
- AWS Serverless Application Model (AWS SAM) CLI, diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang ini, lihat Menginstal AWS SAM CLI di dokumentasi AWS SAM.
- Python 3, diinstal pada mesin lokal Anda. Untuk informasi lebih lanjut tentang ini, lihat dokumentasi Python.
- Klien Git, diinstal dan dikonfigurasi.
- GitHub `IAM roles pipeline`Repositori, dikloning ke mesin lokal Anda.
- Kebijakan dan peran IAM berformat JSON yang ada. Untuk informasi lebih lanjut tentang ini, lihat ReadMefile di repositori Github`IAM roles pipeline`.
- Tim pengembang Anda tidak boleh memiliki izin untuk mengedit AWS CodePipeline CodeBuild, dan CodeDeploy sumber daya solusi ini.

Batasan

- Alur kerja pola ini adalah bukti konsep (POC) dan kami menyarankan Anda hanya menggunakannya di lingkungan pengujian. Jika Anda ingin menggunakan pendekatan pola ini di lingkungan produksi, lihat Praktik terbaik keamanan di IAM dalam dokumentasi IAM dan buat perubahan yang diperlukan pada peran IAM dan layanan AWS Anda.

Arsitektur

Diagram berikut menunjukkan kepada Anda cara memvalidasi dan menerapkan peran dan kebijakan IAM secara otomatis ke akun dengan menggunakan CodePipeline, IAM Access Analyzer, dan makro AWS. CloudFormation

Diagram menunjukkan alur kerja berikut:

1. Pengembang menulis file JSON yang berisi definisi untuk kebijakan dan peran IAM. Pengembang mendorong kode ke CodeCommit repositori dan CodePipeline kemudian memulai pipeline. `RolesPipeline`
2. CodeBuild memvalidasi file JSON dengan menggunakan IAM Access Analyzer. Jika ada keamanan atau temuan terkait kesalahan, proses penyebaran dihentikan.
3. Jika tidak ada keamanan atau temuan terkait kesalahan, file JSON dikirim ke bucket S3. `RolesBucket`
4. CloudFormation Makro AWS yang diimplementasikan sebagai fungsi AWS Lambda kemudian membaca file JSON dari bucket dan mengubahnya menjadi `RolesBucket` tipe sumber daya CloudFormation AWS IAM.
5. CloudFormation Stack AWS yang telah ditentukan sebelumnya menginstal, memperbarui, atau menghapus kebijakan dan peran IAM di akun.

Otomatisasi dan skala

CloudFormation Template AWS yang secara otomatis menerapkan pola ini disediakan di repositori pipeline peran GitHub IAM.

Alat

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- IAM Access Analyzer membantu Anda mengidentifikasi sumber daya di organisasi dan akun Anda, seperti bucket S3 atau peran IAM, yang dibagikan dengan entitas eksternal. Ini membantu Anda mengidentifikasi akses yang tidak diinginkan ke sumber daya dan data Anda.

- AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka yang membantu Anda membangun aplikasi tanpa server di AWS Cloud.

Kode

Kode sumber dan template untuk pola ini tersedia di repositori pipeline peran GitHub IAM.

Epik

Kloning repositori

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori sampel.	Kloning repositori pipeline peran GitHub IAM ke mesin lokal Anda.	Pengembang aplikasi, AWS Umum

Menyebarkan pipa RolesPipeline

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan pipa.	1. Arahkan ke direktori yang berisi repositori kloning. 2. **Important** Jalankan `make deploy bucket=<bucket_name>` perintah.: Anda harus mengganti `<bucket_name>` dengan nama bucket untuk bucket S3 yang ada.	Pengembang aplikasi, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Jalankan `aws codepipeline get-pipeline --name RolesPipeline` perintah untuk memeriksa apakah penerapan Anda berhasil.	
Kloning repositori pipeline.	1. CloudFormation Tumpukan `RolesPipeline` AWS membuat `roles-pipeline-repo` CodeCommit repositori. 2. Masuk ke AWS Management Console, buka CodeCommit konsol AWS, lalu salin URL CodeCommit repositori untuk mengkloningnya ke mesin lokal Anda. Untuk informasi selengkapnya tentang ini, lihat Connect ke CodeCommit repositori AWS di dokumentasi CodeCommit AWS.	Pengembang aplikasi, AWS Umum

Uji RolesPipeline pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji RolesPipeline pipeline dengan kebijakan dan peran IAM yang valid.	1. Buat file JSON untuk kebijakan dan peran IAM Anda. Anda dapat menggunakan sampel dalam `role-example`	Pengembang aplikasi, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	direktori dari GitHub `IAM roles pipeline` repositori. 2. **Important** Tentukan kebijakan dan peran IAM Anda dengan konfigurasi yang diperlukan. : Pastikan Anda mengikuti format yang dijelaskan dalam `ReadMe` file dari GitHub `IAM roles pipeline` repositori. 3. Dorong modifikasi ke dalam `roles-pipeline-repo` CodeCommit repositori. 4. Verifikasi implementasi `RolesPipeline` pipa. 5. Pastikan bahwa kebijakan dan peran IAM diterapkan dengan benar di akun. 6. Validasi jika ada batas izin yang terkait dengan kebijakan atau peran IAM. Untuk informasi selengkapnya tentang ini, lihat Batas izin untuk entitas IAM dalam dokumentasi IAM.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji RolesPipeline pipeline dengan kebijakan dan peran IAM yang tidak valid.	1. Ubah `roles-pipeline-repo` CodeCommit repositori dan sertakan peran atau kebijakan IAM yang tidak valid. Misalnya, Anda dapat menggunakan tindakan yang tidak ada atau versi kebijakan IAM yang tidak valid. 2. Verifikasi implementasi pipeline. IAM Access Analyzer menghentikan pipeline selama tahap validasi jika mendeteksi kebijakan atau peran IAM yang tidak valid.	Pengembang aplikasi, AWS Umum

Bersihkan sumber daya Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersiaplah untuk pembersihan.	Kosongkan bucket S3 dan kemudian jalankan perintah. `destroy`	Pengembang aplikasi, AWS Umum
Hapus RolesStack tumpukan.	1. `RolesPipeline` Pipeline membuat CloudFormation tumpukan `RolesStack` AWS yang menerapkan kebijakan dan peran IAM. Anda harus menghapus tumpukan ini sebelum menghapus `RolesPipeline` pipeline.	Pengembang aplikasi, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Masuk ke AWS Management Console, buka CloudFormation konsol AWS, lalu pilih `RolesStack` tumpukan dan pilih Hapus.	
Hapus RolesPipeline tumpukan.	Untuk menghapus CloudFormation tumpukan `RolesPipeline` AWS, ikuti instruksi dari ReadMefile di repositori GithubIAM `roles pipeline`.	Pengembang aplikasi, AWS Umum

Sumber daya terkait

- IAM Access Analyzer - Validasi kebijakan (AWS News Blog)
- Menggunakan CloudFormation makro AWS untuk melakukan pemrosesan khusus pada templat (CloudFormation dokumentasi AWS)
- Membangun fungsi Lambda dengan Python (dokumentasi AWS Lambda)

Integrasi dua arah AWS Security Hub CSPM dengan perangkat lunak Jira

Joaquin Rinaudo, Amazon Web Services

Ringkasan

Solusi ini mendukung integrasi dua arah antara AWS Security Hub CSPM dan Jira. Dengan menggunakan solusi ini, Anda dapat secara otomatis dan manual membuat dan memperbarui tiket Jira dari temuan CSPM Security Hub. Tim keamanan dapat menggunakan integrasi ini untuk memberi tahu tim pengembang tentang temuan keamanan parah yang memerlukan tindakan.

Solusinya memungkinkan Anda untuk:

- Pilih Security Hub CSPM yang mengontrol secara otomatis membuat atau memperbarui tiket di Jira.
- Di konsol CSPM Security Hub, gunakan tindakan kustom CSPM Security Hub untuk meningkatkan tiket secara manual di Jira.
- Secara otomatis menetapkan tiket di Jira berdasarkan Akun AWS tag yang ditentukan dalam. AWS Organizations Jika tag ini tidak didefinisikan, penerima tugas default digunakan.
- Secara otomatis menekan temuan CSPM Security Hub yang ditandai sebagai positif palsu atau risiko yang diterima di Jira.
- Secara otomatis menutup tiket JIRA ketika temuan terkait diarsipkan di Security Hub CSPM.
- Buka kembali tiket Jira saat temuan CSPM Security Hub terulang kembali.

Alur kerja Jira

Solusinya menggunakan alur kerja Jira khusus yang memungkinkan pengembang mengelola dan mendokumentasikan risiko. Saat masalah bergerak melalui alur kerja, integrasi dua arah memastikan bahwa status tiket Jira dan temuan CSPM Security Hub disinkronkan di seluruh alur kerja di kedua layanan. Alur kerja ini adalah turunan dari Alur Kerja SecDevOps Risiko oleh Dinis Cruz, berlisensi di bawah Lisensi Apache versi 2.0. Sebaiknya tambahkan kondisi alur kerja Jira sehingga hanya anggota tim keamanan Anda yang dapat mengubah status tiket.

Untuk contoh tiket Jira yang dibuat secara otomatis oleh solusi ini, lihat bagian Informasi tambahan dari pola ini.

Prasyarat dan batasan

Prasyarat

- Jika Anda ingin menerapkan solusi ini di lingkungan multi-akun AWS :
  - Lingkungan multi-akun Anda aktif dan dikelola oleh AWS Organizations.
  - Security Hub CSPM diaktifkan pada Anda. Akun AWS
  - Di AWS Organizations, Anda telah menetapkan akun administrator CSPM Security Hub.
  - Anda memiliki peran lintas akun AWS Identity and Access Management (IAM) yang memiliki `AWSOrganizationsReadOnlyAccess` izin ke akun manajemen. AWS Organizations
  - (Opsional) Anda telah menandai Anda Akun AWS dengan`SecurityContactID`. Tag ini digunakan untuk menetapkan tiket Jira ke kontak keamanan yang ditentukan.
- Jika Anda ingin menerapkan solusi ini dalam satu Akun AWS:
  - Anda memiliki yang aktif Akun AWS.
  - Security Hub CSPM diaktifkan pada Anda. Akun AWS
- Sebuah contoh Pusat Data Jira

> **Important**
>
> Solusi ini mendukung penggunaan Jira Cloud. Namun, Jira Cloud tidak mendukung mengimpor alur kerja XHTML, jadi Anda perlu membuat ulang alur kerja secara manual di Jira. Anda dapat menemukan transisi dan status di GitHub repositori.

- Izin administrator di Jira
- Salah satu token Jira berikut:
  - Untuk Jira Enterprise, token akses pribadi (PAT). Untuk informasi selengkapnya, lihat Menggunakan Token Akses Pribadi (dukungan Atlassian).
  - Untuk Jira Cloud, token API Jira. Untuk informasi selengkapnya, lihat Mengelola token API (dukungan Atlassian).

Arsitektur

Bagian ini menggambarkan arsitektur solusi dalam berbagai skenario, seperti ketika pengembang dan insinyur keamanan memutuskan untuk menerima risiko atau memutuskan untuk memperbaiki masalah.

Skenario 1: Pengembang mengatasi masalah ini

1. Security Hub CSPM menghasilkan temuan terhadap kontrol keamanan tertentu, seperti yang ada dalam standar Praktik Terbaik Keamanan AWS Dasar.
2. CloudWatch Acara Amazon yang terkait dengan temuan dan `CreateJIRA` tindakan memulai suatu AWS Lambda fungsi.
3. Fungsi Lambda menggunakan file konfigurasi dan `GeneratorId` bidang temuan untuk mengevaluasi apakah itu harus meningkatkan temuan.
4. Fungsi Lambda menentukan temuan harus ditingkatkan, ia memperoleh tag `SecurityContactID` akun dari AWS Organizations dalam akun manajemen. AWS ID ini dikaitkan dengan pengembang dan digunakan sebagai ID penerima hak untuk tiket Jira.
5. Fungsi Lambda menggunakan kredenal yang disimpan AWS Secrets Manager untuk membuat tiket di Jira. Jira memberi tahu pengembang.
6. Pengembang membahas temuan keamanan yang mendasarinya dan, di Jira, mengubah status tiket ke`TEST FIX`.
7. Security Hub CSPM memperbarui temuan sebagai`ARCHIVED`, dan acara baru dihasilkan. Peristiwa ini menyebabkan fungsi Lambda menutup tiket Jira secara otomatis.

Skenario 2: Pengembang memutuskan untuk menerima risiko

1. Security Hub CSPM menghasilkan temuan terhadap kontrol keamanan tertentu, seperti yang ada dalam standar Praktik Terbaik Keamanan AWS Dasar.
2. CloudWatch Peristiwa yang terkait dengan temuan dan `CreateJIRA` tindakan memulai fungsi Lambda.
3. Fungsi Lambda menggunakan file konfigurasi dan `GeneratorId` bidang temuan untuk mengevaluasi apakah itu harus meningkatkan temuan.
4. Fungsi Lambda menentukan temuan harus ditingkatkan, ia memperoleh tag `SecurityContactID` akun dari AWS Organizations dalam akun manajemen. AWS ID ini dikaitkan dengan pengembang dan digunakan sebagai ID penerima hak untuk tiket Jira.
5. Fungsi Lambda menggunakan kredensi yang disimpan di Secrets Manager untuk membuat tiket di Jira. Jira memberi tahu pengembang.
6. Pengembang memutuskan untuk menerima risiko dan, di Jira, mengubah status tiket ke`AWAITING RISK ACCEPTANCE`.

7. Insinyur keamanan meninjau permintaan dan menemukan pembenaran bisnis yang tepat. Insinyur keamanan mengubah status tiket Jira menjadi`ACCEPTED RISK`. Ini menutup tiket Jira.
8. Acara CloudWatch harian memulai fungsi penyegaran Lambda, yang mengidentifikasi tiket Jira yang ditutup dan memperbarui temuan CSPM Security Hub terkait sebagai. `SUPPRESSED`

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- Amazon CloudWatch Events membantu Anda memantau peristiwa sistem untuk AWS sumber daya Anda dengan menggunakan aturan untuk mencocokkan peristiwa dan meruteknya ke fungsi atau aliran.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Secrets Managermembantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- AWS Security Hub CSPMmemberikan pandangan komprehensif tentang keadaan keamanan Anda di AWS. Ini juga membantu Anda memeriksa AWS lingkungan Anda terhadap standar industri keamanan dan praktik terbaik.

Repositori kode

Kode untuk pola ini tersedia di GitHub, di repositori aws-securityhub-jira-software-integration. Ini termasuk kode sampel dan alur kerja Jira untuk solusi ini.

Epik

Konfigurasikan Jira

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Impor alur kerja.	Sebagai administrator di Jira, impor `issue-workflow.xml` file ke instance Pusat Data Jira Anda. Jika Anda menggunakan Jira Cloud, Anda perlu membuat alur kerja sesuai dengan file `assets/jira-cloud-transitions.png` dan`assets/jira-cloud-status.png` . File dapat ditemukan di repositori aws-securityhub-jira-software-integration di. GitHub Untuk petunjuk, lihat Menggunakan XMLuntuk membuat alur kerja (dokumentasi Jira).	Administrator Jira
Aktifkan dan tetapkan alur kerja.	Alur kerja tidak aktif sampai Anda menetapkannya ke skema alur kerja. Anda kemudian menetapkan skema alur kerja untuk sebuah proyek. 1. Untuk proyek Anda, pastikan Anda telah mengidentifikasi skema jenis masalah untuk proyek tersebut. Anda dapat	Administrator Jira

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	membuat jenis masalah baru atau memilih dari yang sudah ada, sepertiBug. 2. Tetapkan alur kerja yang diimpor ke skema alur kerja sesuai dengan instruksi di Aktifkan alur kerja (dokumentasi JIRA). 3. Tetapkan skema alur kerja ke proyek sesuai dengan instruksi di Kaitkan skema alur kerja dengan proyek (dokumentasi Jira).	

Siapkan parameter solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan parameter solusi.	1. Di folder conf, buka`params_prod.shfile` . 2. Berikan nilai untuk parameter berikut: • `ORG_ACCOUNT_ID` — ID akun untuk akun AWS Organizations manajemen Anda. Solusinya membaca tag akun dan menetapkan tiket ke kontak keamanan tertentu yang ditentukan dalam Akun AWS tag tersebut.	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `ORG_ROLE`— Nama peran IAM yang digunakan untuk mengakses akun AWS Organizations manajemen. Peran ini harus memiliki `OrganizationsReadOnlyAccess` izin. • `EXTERNAL_ID` — Parameter opsional jika Anda menggunakan ID eksternal untuk mengasumsikan peran IAM yang ditentukan dalam`ORG_ROLE`. Untuk informasi selengkapnya, lihat Cara menggunakan ID eksternal (dokumentasi IAM). • `JIRA_DEFAULT_ASSIGNEE` — Ini adalah penerima tugas default Jira untuk semua masalah keamanan. Penerima tugas default ini digunakan jika akun tidak ditandai dengan benar atau peran tidak dapat diasumsikan. • `JIRA_INSTANCE` — Alamat HTTPS untuk titik akhir Jira Anda dalam format berikut: `team-`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`<team-id>.atlassian.net/` • `JIRA_PROJECT_KEY` — Nama kunci proyek Jira yang digunakan untuk membuat tiket, seperti `SEC` atau`TEST`. Proyek ini pasti sudah ada di Jira. • `ISSUE_TYPE` — Nama skema jenis masalah yang ditetapkan untuk proyek di Jira, seperti Bug atau`Security Issue`. • `REGIONS`— Daftar Wilayah AWS kode tempat Anda ingin menerapkan solusi ini, seperti`eu-west-1` . 3. Simpan dan tutup file parameter solusi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Identifikasi temuan yang ingin Anda otomatiskan.	1. Buka konsol CSPM Security Hub. 2. Di panel navigasi CSPM Security Hub, pilih Temuan. 3. Pilih judul temuan. 4. Pilih ID temuan. Ini menampilkan JSON lengkap untuk temuan. 5. Di JSON, salin string di `GeneratorId` lapangan. Nilai ini ada di AWS Security Finding Format (ASFF). Misalnya, `aws-foundational-security-best-practices/v/1.0.0/S3.1` sesuai dengan temuan dari kontrol keamanan Pengaturan Akses Publik Blok S3.1 S3 harus diaktifkan. 6. Ulangi langkah-langkah ini sampai Anda menyalin semua `GeneratorID` nilai untuk temuan apa pun yang ingin Anda otomatiskan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan temuan ke file konfigurasi.	1. Dalam src/code, buka file. `config.jsonconfig` 2. Rekatkan `Generator ID` nilai yang Anda ambil di cerita sebelumnya ke dalam `default` parameter , dan gunakan koma untuk memisahkan setiap ID. 3. Simpan dan tutup file konfigurasi . Contoh kode berikut menunjukkan otomatisasi `aws-foundational-security-best-practices/v/1.0.0/SNS.1` dan `aws-foundational-security-best-practices/v/1.0.0/S3.1` temuan. `{ "Controls" : { "eu-west-1": [ "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.22" ], "default": [ aws-foundational-security-best-practices/v/1.0.0/SNS.1,`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>aws-foundational-
	security-best-p
	ractices/v/1.0.0/S3.1
	]
	}
	}</pre>
	Note
	Anda dapat memilih untuk mengotoma tiskan temuan yang berbeda untuk masing-masing Wilayah AWS. Praktik yang baik untuk membantu mencegah temuan duplikat adalah memilih satu Wilayah untuk mengotomatiskan pembuatan kontrol yang terkait dengan IAM.	

Menyebarkan integrasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan integrasi.	Di terminal baris perintah, masukkan perintah berikut: <pre>./deploy.sh prod</pre>	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kredensi Jira ke Secrets Manager.	1. Buka konsol Secrets Manager. 2. Di bawah Rahasia, pilih Simpan rahasia baru. 3. Untuk Tipe rahasia, pilih Tipe rahasia lainnya. 4. Jika Anda menggunak an Jira Enterprise, untuk pasangan kunci/nilai, lakukan hal berikut: • Di baris pertama, masukkan `auth` di kotak kunci, lalu masukkan `token_auth` di kotak nilai. • Tambahkan baris kedua, masukkan `token` di kotak kunci, lalu masukkan token akses pribadi Anda di kotak nilai. Jika Anda menggunakan Jira Cloud, untuk pasangan kunci/nilai, lakukan hal berikut: • Di baris pertama, masukkan `auth` di kotak kunci, lalu masukkan `basic_auth` di kotak nilai. • Tambahkan baris kedua, masukkan `token`	Administrator sistem AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	di kotak kunci, lalu masukkan token API Anda di kotak nilai. • Tambahkan baris ketiga, masukkan `email` di kotak kunci, lalu masukkan alamat email Anda di kotak nilai. 5. Pilih Berikutnya. 6. Untuk nama Rahasia`Jira-Token` , masukkan, dan kemudian di bagian bawah halaman, pilih Berikutnya. 7. Pada halaman rotasi Rahasia, tetap Nonaktifk an rotasi otomatis, dan kemudian di bagian bawah halaman, pilih Berikutnya. 8. Pada halaman Ulasan, tinjau detail rahasia, lalu pilih Store.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat tindakan kustom CSPM Security Hub.	1. Untuk masing-masing Wilayah AWS, di AWS Command Line Interface (AWS CLI), gunakan create-action-targetperintah untuk membuat tindakan kustom CSPM Security Hub bernama. `CreateJiraIssue` `aws securityhub create-action-target --name "CreateJiraIssue" \ --description "Create ticket in JIRA" \ --id "CreateJiraIssue" --region $<aws-region>` 2. Buka konsol CSPM Security Hub. 3. Di panel navigasi CSPM Security Hub, pilih Temuan. 4. Dalam daftar temuan, pilih temuan yang ingin Anda eskalasi. 5. Di menu Tindakan, pilih`CreateJiraIssue` .	Administrator sistem AWS

Sumber daya terkait

- AWS Konektor Manajemen Layanan untuk Manajemen Layanan Jira

- AWS Standar Praktik Terbaik Keamanan Dasar

Informasi tambahan

Contoh tiket Jira

Ketika temuan CSPM Security Hub tertentu terjadi, solusi ini secara otomatis membuat tiket Jira. Tiket termasuk informasi berikut:

- Judul - Judul mengidentifikasi masalah keamanan dalam format berikut:

```
AWS Security Issue :: <AWS account ID> :: <Security Hub finding title>
```

- Deskripsi — Bagian deskripsi tiket menjelaskan kontrol keamanan yang terkait dengan temuan, menyertakan tautan ke temuan di konsol CSPM Security Hub, dan memberikan deskripsi singkat tentang cara menangani masalah keamanan dalam alur kerja Jira.

Berikut ini adalah contoh tiket Jira yang dibuat secara otomatis.

judul	AWS Masalah Keamanan:: 012345678912:: Lambda.1 Kebijakan fungsi Lambda harus melarang akses publik.
Deskripsi	Apa masalahnya? Kami mendeteksi temuan keamanan dalam Akun AWS 012345678912 yang menjadi tanggung jawab Anda. Kontrol ini memeriksa apakah kebijakan AWS Lambda fungsi yang dilampirkan pada sumber daya Lambda melarang akses publik. Jika kebijakan fungsi Lambda mengizinkan akses publik, kontrol gagal. <Link to Security Hub CSPM finding> Apa yang harus saya lakukan dengan tiket? • Akses akun dan verifikasi konfigurasi. Akui mengerjakan tiket dengan memindahk

annya ke “Dialokasikan untuk Perbaikan”. Setelah diperbaiki, dipindahkan ke perbaikan pengujian sehingga keamanan memvalidasi masalah ditangani.

- Jika Anda berpikir risiko harus diterima, pindahkan ke “Menunggu Penerimaan Risiko”. Ini akan membutuhkan peninjauan oleh insinyur keamanan.
- Jika Anda berpikir positif palsu, transisik an ke “Tandai sebagai Positif Palsu”. Ini akan ditinjau oleh insinyur keamanan dan reopened/closed karenanya.

Buat pipeline untuk gambar kontainer yang diperkeras menggunakan EC2 Image Builder dan Terraform

Mike Saintcross dan Andrew Ranes, Amazon Web Services

Ringkasan

Pola ini membangun pipeline EC2 Image Builder yang menghasilkan image container dasar Amazon Linux 2 yang diperkeras. Terraform digunakan sebagai alat infrastruktur sebagai kode (IAc) untuk mengonfigurasi dan menyediakan infrastruktur yang digunakan untuk membuat gambar kontainer yang diperkeras. Resepnya membantu Anda menyebarkan image container Amazon Linux 2 berbasis Docker yang telah dikeraskan menurut Red Hat Enterprise Linux (RHEL) 7 STIG Versi 3 Release 7 - Medium. (Lihat STIG-Build-Linux-Medium versi 2022.2.1 di bagian komponen STIG Linux dari dokumentasi EC2 Image Builder.) Ini disebut sebagai gambar wadah emas.

Build ini mencakup dua EventBridge aturan Amazon. Satu aturan memulai pipeline image container ketika temuan Amazon Inspector High atau Critical sehingga gambar yang tidak aman diganti. Aturan ini mengharuskan pemindaian yang ditingkatkan Amazon Inspector dan Amazon Elastic Container Registry (Amazon ECR) Registry ECR) agar diaktifkan. Aturan lain mengirimkan pemberitahuan ke antrian Amazon Simple Queue Service (Amazon SQS) setelah push gambar berhasil ke repositori Amazon ECR, untuk membantu Anda menggunakan gambar kontainer terbaru.

> **Note**
>
> Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs.

Prasyarat dan batasan

Prasyarat

- Akun AWS tempat Anda dapat menerapkan infrastruktur.
- AWS Command Line Interface (AWS CLI) diinstal untuk menyetel kredensi AWS Anda untuk penerapan lokal.
- Terraform diunduh dan diatur dengan mengikuti instruksi dalam dokumentasi Terraform.
- Git (jika Anda menyediakan dari mesin lokal).

- Peran dalam akun AWS yang dapat Anda gunakan untuk membuat sumber daya AWS.
- Semua variabel didefinisikan dalam file.tfvars. Atau, Anda dapat menentukan semua variabel saat menerapkan konfigurasi Terraform.

Batasan

- Solusi ini menciptakan infrastruktur Amazon Virtual Private Cloud (Amazon VPC) yang mencakup gateway NAT dan gateway internet untuk konektivitas internet dari subnet pribadinya. Anda tidak dapat menggunakan titik akhir VPC, karena proses bootstrap oleh AWS Task Orchestrator dan Executor () menginstal AWSTOE AWS CLI versi 2 dari internet.

Versi produk

- Amazon Linux 2
- AWS CLI versi 1.1 atau yang lebih baru

Arsitektur

Tumpukan teknologi target

Pola ini menciptakan 43 sumber daya, termasuk:

- Dua bucket Amazon Simple Storage Service (Amazon S3): satu untuk file komponen pipeline dan satu untuk akses server dan log aliran Amazon VPC
- Repositori ECR Amazon
- Virtual Private Cloud (VPC) yang berisi subnet publik, subnet pribadi, tabel rute, gateway NAT, dan gateway internet
- Pipeline, resep, dan komponen EC2 Image Builder
- Gambar wadah
- Kunci AWS Key Management Service (AWS KMS) untuk enkripsi gambar
- Antrian SQS
- Tiga peran: satu untuk menjalankan pipeline EC2 Image Builder, satu profil instance untuk EC2 Image Builder, dan satu untuk EventBridge aturan
- Dua EventBridge aturan

Struktur modul Terraform

Untuk kode sumber, lihat GitHub repositori Terraform Image EC2 Builder Container Hardening Pipeline.

```
### components.tf
### config.tf
### dist-config.tf
### files
#   ###assumption-policy.json
### hardening-pipeline.tfvars
### image.tf
### infr-config.tf
### infra-network-config.tf
### kms-key.tf
### main.tf
### outputs.tf
### pipeline.tf
### recipes.tf
### roles.tf
### sec-groups.tf
### trigger-build.tf
### variables.tf
```

Rincian modul

- `components.tf`berisi sumber daya unggah Amazon S3 untuk mengunggah konten direktori. `/files` Anda juga dapat menambahkan file YAMM komponen kustom secara modular di sini juga.
- `/files`berisi `.yml` file yang menentukan komponen yang digunakan dalam`components.tf`.
- `image.tf`berisi definisi untuk sistem operasi gambar dasar. Di sinilah Anda dapat memodifikasi definisi untuk pipeline gambar dasar yang berbeda.
- `infr-config.tf`dan `dist-config.tf` berisi sumber daya untuk infrastruktur AWS minimum yang diperlukan untuk memutar dan mendistribusikan gambar.
- `infra-network-config.tf`berisi infrastruktur VPC minimum untuk menyebarkan gambar kontainer ke dalam.
- `hardening-pipeline.tfvars`berisi variabel Terraform yang akan digunakan pada waktu penerapan.
- `pipeline.tf`membuat dan mengelola pipeline EC2 Image Builder di Terraform.

- `recipes.tf`adalah tempat Anda dapat menentukan campuran komponen yang berbeda untuk membuat resep wadah.
- `roles.tf`berisi definisi kebijakan AWS Identity and Access Management (IAM) untuk profil instans Amazon Elastic Compute Cloud EC2 (Amazon) dan peran penerapan pipeline.
- `trigger-build.tf`berisi EventBridge aturan dan sumber daya antrian SQS.

Arsitektur target

Diagram menggambarkan alur kerja berikut:

1. EC2 Image Builder membuat image kontainer dengan menggunakan resep yang ditentukan, yang menginstal pembaruan sistem operasi dan menerapkan RHEL Medium STIG ke image dasar Amazon Linux 2.
2. Gambar yang dikeraskan dipublikasikan ke registri ECR Amazon pribadi, dan EventBridge aturan mengirim pesan ke antrian SQS ketika gambar telah berhasil diterbitkan.
3. Jika Amazon Inspector dikonfigurasi untuk pemindaian yang disempurnakan, Amazon Inspector memindai registri Amazon ECR.
4. Jika Amazon Inspector menghasilkan temuan tingkat keparahan Kritis atau Tinggi untuk gambar, EventBridge aturan akan memicu pipeline Image Builder untuk berjalan lagi dan memublikasikan EC2 gambar yang baru dikeraskan.

Otomatisasi dan skala

- Pola ini menjelaskan cara menyediakan infrastruktur dan membangun pipeline di komputer Anda. Namun, ini dimaksudkan untuk digunakan dalam skala. Alih-alih menerapkan modul Terraform secara lokal, Anda dapat menggunakannya di lingkungan multi-akun, seperti AWS Control Tower dengan Account Factory untuk lingkungan Terraform. Dalam hal ini, Anda harus menggunakan bucket S3 status backend untuk mengelola file status Terraform alih-alih mengelola status konfigurasi secara lokal.
- Untuk penggunaan yang diskalakan, terapkan solusi ke satu akun pusat, seperti akun Layanan Bersama atau Layanan Umum, dari Control Tower atau model akun landing zone, dan berikan izin kepada akun konsumen untuk mengakses repositori Amazon ECR dan kunci AWS KMS. Untuk informasi lebih lanjut tentang pengaturan, lihat artikel re:Post Bagaimana cara mengizinkan akun sekunder untuk mendorong atau menarik gambar di repositori gambar Amazon ECR saya? Misalnya, di mesin penjual otomatis akun atau Account Factory untuk Terraform, tambahkan izin

ke setiap baseline akun atau baseline penyesuaian akun untuk menyediakan akses ke repositori Amazon ECR dan kunci enkripsi tersebut.

- Setelah pipeline image container di-deploy, Anda dapat memodifikasinya dengan menggunakan fitur EC2 Image Builder seperti komponen, yang membantu Anda mengemas lebih banyak komponen ke dalam build Docker.
- Kunci AWS KMS yang digunakan untuk mengenkripsi image kontainer harus dibagikan di seluruh akun tempat gambar dimaksudkan untuk digunakan.
- Anda dapat menambahkan dukungan untuk gambar lain dengan menduplikasi seluruh modul Terraform dan memodifikasi atribut berikut: `recipes.tf`
  - Ubah `parent_image = "amazonlinux:latest"` ke jenis gambar lain.
  - Ubah `repository_name` untuk menunjuk ke repositori Amazon ECR yang ada. Ini membuat pipeline lain yang menerapkan jenis gambar induk yang berbeda ke repositori Amazon ECR Anda yang ada.

Alat

Alat

- Terraform (penyediaan IAc)
- Git (jika penyediaan secara lokal)
- AWS CLI versi 1 atau versi 2 (jika disediakan secara lokal)

Kode

Kode untuk pola ini ada di GitHub repositori Terraform Image EC2 Builder Container Hardening Pipeline. Untuk menggunakan kode sampel, ikuti instruksi di bagian selanjutnya.

Epik

Penyediaan infrastruktur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan kredensil lokal.	Siapkan kredensi sementara AWS Anda.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	1. Lihat apakah AWS CLI diinstal:
	`$ aws --version`
	`aws-cli/1.16.249 Python/3.6.8...`
	• Versi AWS CLI harus 1.1 atau lebih baru.
	• Jika perintah tidak ditemukan, instal AWS CLI.
	2. Jalankan `aws configure` dan berikan nilai-nilai berikut:
	`$ aws configure`
	`AWS Access Key ID [**********xxxx]: <Your AWS access key ID>`
	`AWS Secret Access Key [**********xxxx]: <Your AWS secret access key>`
	`Default region name: [us-east-1]: <Your desired Region for deployment>`
	`Default output format [None]: <Your desired output format>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori.	1. Kloning repositori yang disediakan dengan pola ini. Anda dapat menggunakan HTTPS atau Secure Shell (SSH).
	
	HTTPS:
	
	`git clone https://github.com/aws-samples/terraform-ec2-image-builder-container-hardening-pipeline`
	
	SSH:
	
	`git clone git@github.com:aws-samples/terraform-ec2-image-builder-container-hardening-pipeline.git`
	2. Arahkan ke direktori lokal Anda yang berisi solusi ini:
	
	`cd terraform-ec2-image-builder-container-hardening-pipeline`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui variabel.	Perbarui variabel dalam `hardening-pipeline.tfvars` file agar sesuai dengan lingkungan Anda dan konfigurasi yang Anda inginkan. Anda harus menyediakan milik Anda sendiri`account_id`. Namun, Anda juga harus memodifikasi variabel lainnya agar sesuai dengan penerapan yang Anda inginkan. Semua variabel diperlukan.
	
	`account_id = "<DEPLOYMENT-ACCOUNT-ID>"`
	`aws_region = "us-east-1"`
	`vpc_name = "example-hardening-pipeline-vpc"`
	`kms_key_alias = "image-builder-container-key"`
	`ec2_iam_role_name = "example-hardening-instance-role"`
	`hardening_pipeline_role_name = "example-hardening-pipeline-role"`
	`aws_s3_ami_resources_bucket = "example-hardening-ami-resources-bucket-0123"`
	`image_name = "example-hardening-al2-container-image"`
		AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>ecr_name = "example-hardening-container-repo"
	recipe_version = "1.0.0"
	ebs_root_vol_size = 10</pre>
	Berikut adalah deskripsi dari masing-masing variabel:
	• `account_id` Nomor akun AWS yang ingin Anda gunakan solusinya.
	• `aws_region` Wilayah AWS tempat Anda ingin menerapkan solusinya.
	• `vpc_name`- Nama untuk infrastruktur VPC Anda.
	• `kms_key_alias` - Nama kunci AWS KMS yang akan digunakan oleh konfigura si infrastruktur EC2 Image Builder.
	• `ec2_iam_role_name` - Nama untuk peran yang akan digunakan sebagai profil EC2 instance.
	• `hardening_pipeline _role_name` - Nama untuk peran yang akan digunakan untuk menyebark an pipa pengerasan.
	• `aws_s3_ami_resourc es_bucket` - Nama untuk bucket S3 yang akan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menampung semua file yang diperlukan untuk membangun pipeline dan gambar kontainer. • `image_name` - Nama gambar kontainer. Nilai ini harus antara 3 dan 50 karakter dan harus berisi karakter alfanumerik dan tanda hubung saja. • `ecr_name`- Nama registri Amazon ECR untuk menyimpan gambar kontainer. • `recipe_version` - Versi resep gambar. Nilai default adalah 1.0.0. • `ebs_root_vol_size` Ukuran (dalam gigabyte) volume root Amazon Elastic Block Store (Amazon EBS). Nilai default adalah 10 gigabyte.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi Terraform.	Setelah memperbarui nilai variabel, Anda dapat menginisi alisasi direktori konfigura si Terraform. Menginisi alisasi direktori konfigurasi mengunduh dan menginstal penyedia AWS, yang ditentuka n dalam konfigurasi. `terraform init` Anda akan melihat pesan yang mengatakan Terraform telah berhasil diinisialisasi dan mengidentifikasi versi penyedia yang diinstal.	AWS DevOps
Menyebarkan infrastruktur dan membuat gambar kontainer.	Gunakan perintah berikut untuk menginisialisasi, memvalidasi, dan menerapkan modul Terraform ke lingkunga n dengan menggunakan variabel yang ditentukan dalam file Anda: `.tfvars` `terraform init && terraform validate && terraform apply -var-file *.tfvars -auto-approve`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan wadahnya.	Anda dapat membuat versi baru resep kontainer setelah EC2 Image Builder menerapkan pipeline dan resep awal. Anda dapat menambahkan salah satu dari 31+ komponen yang tersedia dalam EC2 Image Builder untuk menyesuaikan build container. Untuk informasi selengkapnya, lihat bagian Komponen Buat versi baru resep kontainer di dokumentasi EC2 Image Builder.	Administrator AWS

Validasi sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penyediaan infrastruktur AWS.	Setelah Anda berhasil menyelesaikan `apply` perintah Terraform pertama Anda, jika Anda menyediakan secara lokal, Anda akan melihat cuplikan ini di terminal mesin lokal Anda: `Apply complete! Resources: 43 added, 0 changed, 0 destroyed.`	AWS DevOps
Validasi sumber daya infrastruktur AWS individual.	Untuk memvalidasi sumber daya individual yang digunakan, jika Anda	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menyediakan secara lokal, Anda dapat menjalankan perintah berikut: `terraform state list` Perintah ini mengembalikan daftar 43 sumber daya.	

Hapus sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus infrastruktur dan gambar kontainer.	Setelah selesai bekerja dengan konfigurasi Terraform , Anda dapat menjalank an perintah berikut untuk menghapus sumber daya: `terraform init && terraform validate && terraform destroy -var-file *.tfvars -auto-approve`	AWS DevOps

Pemecahan masalah

Isu	Solusi
Kesalahan memvalidasi kredensi penyedia	Saat Anda menjalankan Terraform `apply` atau `destroy` perintah dari mesin lokal Anda, Anda mungkin mengalami kesalahan yang mirip dengan berikut ini:

Isu	Solusi
	`Error: configuring Terraform AWS Provider: error validating provider credentials: error calling sts:GetCallerIdentity: operation error STS: GetCallerIdentity, https response error StatusCode: 403, RequestID: 123456a9-fbc1-40ed-b8d8-513d0133ba7f, api error InvalidClientTokenId: The security token included in the request is invalid.` Kesalahan ini disebabkan oleh kedaluwarsa token keamanan untuk kredensil yang digunakan dalam konfigurasi mesin lokal Anda. Untuk mengatasi kesalahan, lihat Mengatur dan melihat setelan konfigurasi dalam dokumentasi AWS CLI.

Sumber daya terkait

- Pipa Pengerasan Kontainer EC2 Image Builder Terraform (repositori) GitHub
- EC2 Dokumentasi Image Builder
- AWS Control Tower Account Factory untuk Terraform (postingan blog AWS)
- Bucket S3 status backend (dokumentasi Terraform)
- Menginstal atau memperbarui versi terbaru AWS CLI (dokumentasi AWS CLI)
- Unduh Terraform

Memusatkan manajemen kunci akses IAM di AWS Organizations dengan menggunakan Terraform

Aarti Rajput, Chintamani Aphale, T.V.R.L.Phani Kumar Dadi, Pratap Kumar Nanda, Pradip kumar Pandey, dan Mayuri Shinde, Amazon Web Services

Ringkasan

Menegakkan aturan keamanan untuk kunci dan kata sandi adalah tugas penting bagi setiap organisasi. Salah satu aturan penting adalah memutar kunci AWS Identity and Access Management (IAM) secara berkala untuk menegakkan keamanan. Kunci akses AWS umumnya dibuat dan dikonfigurasi secara lokal setiap kali tim ingin mengakses AWS dari AWS Command Line Interface (AWS CLI) atau dari aplikasi di luar AWS. Untuk menjaga keamanan yang kuat di seluruh organisasi, kunci keamanan lama harus diubah atau dihapus setelah persyaratan dipenuhi atau secara berkala. Proses mengelola rotasi kunci di beberapa akun dalam suatu organisasi memakan waktu dan membosankan. Pola ini membantu Anda mengotomatiskan proses rotasi dengan menggunakan Account Factory for Terraform (AFT) dan layanan AWS.

Pola ini memberikan manfaat ini:

- Mengelola kunci akses IDs dan kunci akses rahasia Anda di semua akun di organisasi Anda dari lokasi pusat.
- Secara otomatis memutar variabel `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` lingkungan.
- Menegakkan perpanjangan jika kredensi pengguna disusupi.

Pola ini menggunakan Terraform untuk menerapkan fungsi AWS Lambda, aturan EventBridge Amazon, dan peran IAM. EventBridge Aturan berjalan secara berkala dan memanggil fungsi Lambda yang mencantumkan semua kunci akses pengguna berdasarkan kapan mereka dibuat. Fungsi Lambda tambahan membuat ID kunci akses baru dan kunci akses rahasia, jika kunci sebelumnya lebih tua dari periode rotasi yang Anda tentukan (misalnya, 45 hari), dan beri tahu administrator keamanan dengan menggunakan Amazon Simple Notification Service (Amazon SNS) dan Amazon Simple Email Service (Amazon SES). Rahasia dibuat di AWS Secrets Manager untuk pengguna tersebut, kunci akses rahasia lama disimpan di Secrets Manager, dan izin untuk mengakses kunci lama dikonfigurasi. Untuk memastikan bahwa kunci akses lama tidak lagi digunakan, itu dinonaktifkan setelah periode tidak aktif (misalnya, 60 hari, yang akan menjadi 15 hari setelah kunci diputar dalam

contoh kami). Setelah periode buffer tidak aktif (misalnya, 90 hari, atau 45 hari setelah kunci diputar dalam contoh kami), kunci akses lama dihapus dari AWS Secrets Manager. Untuk detail arsitektur dan alur kerja, lihat bagian Arsitektur.

Prasyarat dan batasan

- Sebuah landing zone untuk organisasi Anda yang dibangun menggunakan AWS Control Tower (versi 3.1 atau yang lebih baru)
- Account Factory for Terraform (AFT) dikonfigurasi dengan tiga akun:
  - Akun manajemen organisasi mengelola seluruh organisasi dari lokasi pusat.
  - Akun manajemen AFT menghosting pipeline Terraform dan menyebarkan infrastruktur ke dalam akun penerapan.
  - Akun penyebaran menyebarkan solusi lengkap ini dan mengelola kunci IAM dari lokasi pusat.
- Terraform versi 0.15.0 atau yang lebih baru untuk menyediakan infrastruktur di akun penerapan.
- Alamat email yang dikonfigurasi di Amazon Simple Email Service (Amazon SES).
- (Disarankan) Untuk meningkatkan keamanan, gunakan solusi ini di dalam subnet pribadi (akun penyebaran) dalam virtual private cloud (VPC). Anda dapat memberikan detail VPC dan subnet saat Anda menyesuaikan variabel (lihat Menyesuaikan parameter untuk pipeline kode di bagian Epik).

Arsitektur

Repositori AFT

Pola ini menggunakan Account Factory for Terraform (AFT) untuk membuat semua sumber daya AWS yang diperlukan dan pipeline kode untuk menyebarkan sumber daya di akun penerapan. Pipeline kode berjalan di dua repositori:

- Kustomisasi global berisi kode Terraform yang akan berjalan di semua akun yang terdaftar di AFT.
- Kustomisasi akun berisi kode Terraform yang akan berjalan di akun penerapan.

Detail sumber daya

CodePipeline Pekerjaan AWS membuat sumber daya berikut di akun penerapan:

- EventBridge Aturan AWS dan aturan yang dikonfigurasi

- `account-inventory`Fungsi Lambda
- `IAM-access-key-rotation`Fungsi Lambda
- `Notification`Fungsi Lambda
- Bucket Amazon Simple Storage Service (Amazon S3) yang berisi template email
- Kebijakan IAM yang diperlukan

Arsitektur

Diagram ini menggambarkan hal sebagai berikut:

1. EventBridge Aturan memanggil fungsi `account-inventory` Lambda setiap 24 jam.
2. Fungsi `account-inventory` Lambda menanyakan AWS Organizations untuk daftar semua akun AWS, nama akun IDs, dan email akun.
3. Fungsi `account-inventory` Lambda memulai fungsi `IAM-access-key-auto-rotation` Lambda untuk setiap akun AWS dan meneruskan metadata ke dalamnya untuk pemrosesan tambahan.
4. Fungsi `IAM-access-key-auto-rotation` Lambda menggunakan peran IAM yang diasumsikan untuk mengakses akun AWS. Skrip Lambda menjalankan audit terhadap semua pengguna dan kunci akses IAM mereka di akun.
5. Ambang batas rotasi kunci IAM (periode rotasi) dikonfigurasi sebagai variabel lingkungan saat fungsi `IAM-access-key-auto-rotation` Lambda diterapkan. Jika periode rotasi diubah, fungsi `IAM-access-key-auto-rotation` Lambda di-deploy kembali dengan variabel lingkungan yang diperbarui. Anda dapat mengonfigurasi parameter untuk mengatur periode rotasi, periode tidak aktif untuk kunci lama, dan buffer tidak aktif setelah itu kunci lama akan dihapus (lihat Menyesuaikan parameter untuk pipa kode di bagian Epik).
6. Fungsi `IAM-access-key-auto-rotation` Lambda memvalidasi usia kunci akses berdasarkan konfigurasinya. Jika usia kunci akses IAM belum melebihi periode rotasi yang Anda tentukan, fungsi Lambda tidak akan mengambil tindakan lebih lanjut.
7. Jika usia kunci akses IAM telah melebihi periode rotasi yang Anda tentukan, fungsi `IAM-access-key-auto-rotation` Lambda membuat kunci baru dan memutar kunci yang ada.
8. Fungsi Lambda menyimpan kunci lama di Secrets Manager dan membatasi izin kepada pengguna yang kunci aksesnya menyimpang dari standar keamanan. Fungsi Lambda juga membuat kebijakan berbasis sumber daya yang memungkinkan hanya prinsipal IAM yang ditentukan untuk mengakses dan mengambil rahasia.

9. Fungsi `IAM-access-key-rotation` Lambda memanggil fungsi Lambda`Notification`.
10. Fungsi `Notification` Lambda menanyakan bucket S3 untuk template email dan secara dinamis menghasilkan pesan email dengan metadata aktivitas yang relevan.
11. Fungsi `Notification` Lambda memanggil Amazon SES untuk tindakan lebih lanjut.
12. Amazon SES mengirimkan email ke alamat email pemilik akun dengan informasi yang relevan.

Alat

Layanan AWS

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya. Patern ini membutuhkan peran dan izin IAM.
- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Secrets Manager membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram.
- Amazon Simple Email Service (Amazon SES) membantu Anda mengirim dan menerima email dengan menggunakan alamat email dan domain Anda sendiri.

Alat-alat lainnya

- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Instruksi dan kode untuk pola ini tersedia di repositori rotasi kunci akses GitHub IAM. Anda dapat menerapkan kode di akun penerapan pusat AWS Control Tower untuk mengelola rotasi kunci dari lokasi pusat.

Praktik terbaik

- Untuk IAM, lihat praktik terbaik keamanan dalam dokumentasi IAM.

- Untuk rotasi kunci, lihat panduan untuk memperbarui kunci akses dalam dokumentasi IAM.

Epik

Siapkan file sumber

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kloning repositori.	1. Kloning repositori rotasi GitHub kunci akses IAM:
	`$ git clone https://github.com/aws-samples/centralized-iam-key-management-aws-organizations-terraform.git`
	2. Konfirmasikan bahwa salinan lokal repositori Anda berisi tiga folder:
	`$ cd Iam-Access-keys-Rotation`
	`$ ls`
	`org-account-customization`
	`global-account-customization`
	`account-customization`
		DevOps insinyur

Konfigurasikan akun

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan akun bootstrapping.	Sebagai bagian dari proses bootstrap AFT, Anda harus	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	memiliki folder yang dipanggil `aft-bootstrap` di mesin lokal Anda.
	
	1. Salin semua file Terraform secara manual dari GitHub org-account-customization folder lokal Anda ke folder Anda `aft-bootstrap` .
	2. Jalankan perintah Terraform untuk mengonfigurasi peran lintas akun global di akun manajemen AWS Control Tower:
	
	`$ cd aft-bootstrap`
	`$ terraform init`
	`$ terraform apply –auto-approve`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan penyesuaian global.	Sebagai bagian dari pengaturan folder AFT, Anda harus memiliki folder yang dipanggil `aft-global-customizations` di mesin lokal Anda.
	
	1. Salin semua file Terraform secara manual dari GitHub global-account-customizationfolder lokal Anda ke folder Anda`aft-global-customizations/terraform .`
	2. Dorong kode ke AWS CodeCommit:
	
	`$ git add *`
	`$ git commit -m "message"`
	`$ git push`
		DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Konfigurasikan kustomisasi akun.	Sebagai bagian dari pengaturan folder AFT, Anda telah menjadi folder yang dipanggil `aft-account-customizations` di mesin lokal Anda.
	1. Buat folder dengan nomor akun penjual Anda.
	2. Salin manual semua file Terraform secara manual dari folder GitHub kustomisasi akun lokal Anda ke folder Anda. `aft-account-customizations/<vended account>/terraform`
	3. Dorong kode ke AWS CodeCommit:
	`$ git add *`
	`$ git commit -m "message"`
	`$ git push`
		DevOps insinyur

Sesuaikan parameter untuk pipa kode

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan parameter pipeline kode non-Terraform untuk semua akun.	Buat file yang disebut `input.auto.tfvars` di `aft-global-customizations/terraform/` folder dan berikan data input	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang diperlukan. Lihat file di GitHub repositori untuk nilai default.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan parameter pipeline kode untuk akun penerapan.	Buat file yang dipanggil `input.auto.tfvars` di `aft-account-customizations/<AccountName>/terraform/` folder dan tekan kode ke AWS CodeCommit. Mendorong kode ke AWS CodeCommit secara otomatis memulai pipeline kode. Tentukan nilai untuk parameter berdasarkan persyaratan organisasi Anda, termasuk yang berikut ini (lihat file di repositori Github untuk nilai default): • `s3_bucket_name` — Nama ember unik untuk template email. • `s3_bucket_prefix` — Nama folder di dalam ember S3. • `admin_email_address` — Alamat email untuk administrator yang harus menerima pemberitahuan. • `org_list_account` — Nomor akun akun manajemen. • `rotation_period` — Jumlah hari setelah itu kunci	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	harus diputar dari aktif ke tidak aktif. • `inactive_period` — Jumlah hari setelah tombol yang diputar harus dinonaktifkan. Nilai ini harus lebih besar dari nilai`rotation_period` . • `inactive_buffer` — Masa tenggang antara rotasi dan penonaktifan kunci. • `recovery_grace_period` — Masa tenggang antara penonaktifan dan penghapusan kunci. • `dry_run_flag` — Setel ke true jika Anda ingin mengirim pemberitahuan ke administrator untuk tujuan pengujian, tanpa memutar kunci. • `store_secrets_in_central_account` — Setel ke true jika Anda ingin menyimpan rahasia di akun penerapan. Jika variabel diatur ke false (default), rahasia akan disimpan di akun anggota. • `credential_replication_region` — Wilayah AWS tempat Anda ingin	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menerapkan fungsi Lambda dan bucket S3 untuk template email. • `run_lambda_in_vpc` — Setel ke true untuk menjalankan fungsi Lambda di dalam VPC. • `vpc_id`— ID VPC dari akun penerapan, jika Anda ingin menjalankan fungsi Lambda di dalam VPC. • `vpc_cidr`— Rentang CIDR untuk akun penerapan. • `subnet_id` — Subnet IDs untuk akun penyebaran. • `create_smtp_endpoint` — Setel ke true jika Anda ingin mengaktifkan titik akhir email.	

Validasi rotasi kunci

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi solusinya.	1. Dari AWS Management Console, masuk ke akun penerapan. 2. Buka konsol IAM dan periksa apakah kredensil pengguna (kunci akses IDs dan kunci rahasia) sedang diputar seperti yang ditentukan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Setelah kunci IAM diputar, konfirmasikan hal berikut: • Nilai lama disimpan di AWS Secrets Manager. • Nama rahasianya ada dalam format`Account_<account ID>_User_<username>_AccessKey` . • Pengguna yang Anda tentukan dalam `admin_email_address` parameter menerima pemberitahuan email tentang rotasi kunci.	

Perpanjang solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan tanggal pemberitahuan email.	Jika Anda ingin mengirim notifikasi email pada hari tertentu sebelum menonaktifkan tombol akses, Anda dapat memperbarui fungsi `IAM-access-key-auto-rotation` Lambda dengan perubahan tersebut: 1. Tentukan variabel yang disebut`notify-period` . 2. Tambahkan `if` kondisi `main.py` sebelum Anda menonaktifkan kunci:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`If (keyage>rotation-period-notify-period){ send_to_notifier(context, aws_account_id, account_name, resource_owner, resource_actions[resource_owner], dryrun, config.emailTemplateAudit) }`	

Pemecahan masalah

Isu	Solusi
Pekerjaan `account-inventory` Lambda gagal `AccessDenied` saat mendaftar akun.	Jika mengalami masalah ini, Anda harus memvalidasi izin: 1. Masuk ke akun yang baru dijual, buka CloudWatch konsol Amazon, lalu lihat grup CloudWatch `/aws/lambda/account-inventory-lambda` log. 2. Di CloudWatch log terbaru, identifikasi nomor akun yang menyebabkan masalah akses ditolak. 3. Masuk ke akun manajemen AWS Control Tower dan konfirmasikan bahwa peran `allow-list-account` telah dibuat. 4. Jika peran tidak ada, jalankan kembali kode Terraform dengan menggunakan perintah. `terraform apply`

Isu	Solusi
	5. Pilih tab Akun Tepercaya dan validasi bahwa akun yang sama dipercaya.

Sumber daya terkait

- Praktik yang Direkomendasikan Terraform (dokumentasi Terraform)
- Praktik terbaik keamanan dalam IAM (dokumentasi IAM)
- Praktik terbaik untuk rotasi kunci (dokumentasi IAM)

Periksa CloudFront distribusi Amazon untuk log akses, HTTPS, dan versi TLS

SaiJeevan Devireddy dan Bijesh Bal, Amazon Web Services

Ringkasan

Pola ini memeriksa CloudFront distribusi Amazon untuk memastikan bahwa ia menggunakan HTTPS, menggunakan Transport Layer Security (TLS) versi 1.2 atau yang lebih baru, dan memiliki akses logging diaktifkan. CloudFront adalah layanan yang disediakan oleh Amazon Web Services (AWS) yang mempercepat distribusi konten web statis dan dinamis Anda, seperti.html, .css, .js, dan file gambar, kepada pengguna Anda. CloudFront mengirimkan konten Anda melalui jaringan pusat data di seluruh dunia yang disebut lokasi tepi. Saat pengguna meminta konten yang Anda sajikan CloudFront, permintaan akan diarahkan ke lokasi tepi yang memberikan latensi terendah (penundaan waktu), sehingga konten dikirimkan dengan performa terbaik.

Pola ini menyediakan fungsi AWS Lambda yang dimulai saat Amazon CloudWatch Events mendeteksi panggilan CloudFront CreateDistributionAPI,, atau. CreateDistributionWithTagsUpdateDistribution Logika kustom dalam fungsi Lambda mengevaluasi semua CloudFront distribusi yang dibuat atau diperbarui di akun AWS. Ini mengirimkan pemberitahuan pelanggaran dengan menggunakan Amazon Simple Notification Service (Amazon SNS) jika mendeteksi pelanggaran berikut:

- Pemeriksaan global:
  - Sertifikat khusus tidak menggunakan TLS versi 1.2
  - Logging dinonaktifkan untuk distribusi
- Pemeriksaan asal:
  - Origin tidak dikonfigurasi dengan TLS versi 1.2
  - Komunikasi dengan asal diperbolehkan pada protokol selain HTTPS
- Pemeriksaan perilaku:
  - Komunikasi perilaku default diperbolehkan pada protokol selain HTTPS
  - Komunikasi perilaku khusus diizinkan pada protokol selain HTTPS

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Alamat email tempat Anda ingin menerima pemberitahuan pelanggaran

Batasan

- Kontrol keamanan ini tidak memeriksa distribusi Cloudfront yang ada kecuali pembaruan telah dilakukan untuk distribusi.
- CloudFront dianggap sebagai layanan global dan tidak terikat dengan Wilayah AWS tertentu. Namun, Amazon CloudWatch Logs dan AWS Cloudtrail API logging untuk layanan global terjadi di Wilayah AS Timur (Virginia N.). `us-east-1` Oleh karena itu, kontrol keamanan untuk ini CloudFront harus digunakan dan dipelihara di`us-east-1`. Penerapan tunggal ini memantau semua distribusi untuk. CloudFront Jangan menerapkan kontrol keamanan di Wilayah AWS lainnya. (Penerapan di Wilayah lain akan mengakibatkan kegagalan untuk memulai CloudWatch Acara dan fungsi Lambda, dan tidak ada pemberitahuan SNS.)
- Solusi ini telah melalui pengujian ekstensif dengan distribusi konten CloudFront web. Ini tidak mencakup distribusi streaming real-time messaging protocol (RTMP).

Arsitektur

Tumpukan teknologi target

- Fungsi Lambda
- Topik SNS
- EventBridge Aturan Amazon

Arsitektur target

Otomatisasi dan skala

- Jika Anda menggunakan AWS Organizations, Anda dapat menggunakan AWS Cloudformation StackSets untuk menerapkan template terlampir di beberapa akun yang ingin Anda pantau.

Alat

Layanan AWS

- AWS CloudFormation — CloudFormation adalah layanan yang membantu Anda memodelkan dan menyiapkan sumber daya AWS dengan menggunakan infrastruktur sebagai kode.
- Amazon EventBridge — EventBridge menghadirkan aliran data real-time dari aplikasi Anda sendiri, aplikasi perangkat lunak sebagai layanan (SaaS), dan layanan AWS, merutekan data tersebut ke target seperti fungsi Lambda.
- AWS Lambda — Lambda mendukung menjalankan kode tanpa menyediakan atau mengelola server.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang sangat skalabel yang dapat digunakan untuk berbagai solusi penyimpanan, termasuk situs web, aplikasi seluler, cadangan, dan danau data.
- Amazon SNS — Amazon SNS mengkoordinasikan dan mengelola pengiriman atau pengiriman pesan antara penerbit dan klien, termasuk server web dan alamat email. Pelanggan menerima semua pesan yang dipublikasikan ke topik tempat mereka berlangganan, dan semua pelanggan ke suatu topik menerima pesan yang sama.

Kode

Kode terlampir meliputi:

- File zip yang berisi kode Lambda (index.py)
- CloudFormation Template (file.yl) yang Anda jalankan untuk menyebarkan kode Lambda

Epik

Unggah kontrol keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat bucket S3 untuk kode Lambda.	Di konsol Amazon S3, buat bucket S3 dengan nama unik yang tidak mengandung garis miring di depan. Nama bucket S3 unik secara global, dan namespace dibagikan oleh semua akun AWS. Bucket S3 Anda harus berada di Wilayah	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tempat Anda berencana untuk menerapkan kode Lambda.	
Unggah kode Lambda ke bucket S3.	Unggah kode Lambda (file cloudfront_ssl_log_lambda.zip) yang disediakan di bagian Lampiran ke bucket S3 yang Anda buat di langkah sebelumnya.	Arsitek awan

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	Di CloudFormation konsol AWS, di Wilayah AWS yang sama dengan bucket S3, terapkan CloudFormation template (cloudfront-ssl-logging.yl) yang disediakan di bagian Lampiran.	Arsitek awan
Tentukan nama bucket S3.	Untuk parameter S3 Bucket, tentukan nama bucket S3 yang Anda buat di epik pertama.	Arsitek awan
Tentukan nama kunci Amazon S3 untuk file Lambda.	Untuk parameter Kunci S3, tentukan lokasi Amazon S3 dari file kode Lambda .zip di bucket S3 Anda. Jangan sertakan garis miring utama (misalnya, Anda dapat memasukkan lambda.zip atau controls/lambda.zip).	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan alamat email pemberitahuan.	Untuk parameter email Pemberitahuan, berikan alamat email tempat Anda ingin menerima pemberita huan pelanggaran.	Arsitek awan
Tentukan tingkat logging.	Untuk parameter level Lambda Logging, tentukan level logging untuk fungsi Lambda Anda. Pilih salah satu nilai berikut: • INFO untuk mendapatkan pesan informasi terperinci tentang kemajuan aplikasi. • ERROR untuk mendapatkan informasi tentang peristiwa kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. • PERINGATAN untuk mendapatkan informasi tentang situasi yang berpotensi berbahaya.	Arsitek awan

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Ketika CloudFormation template telah berhasil digunakan, topik SNS baru dibuat dan pesan berlangganan dikirim ke	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	alamat email yang Anda berikan. Anda harus mengonfir masi langganan email ini untuk menerima pemberita huan pelanggaran.	

Sumber daya terkait

- CloudFormation Informasi AWS
- Membuat tumpukan di CloudFormation konsol AWS (CloudFormation dokumentasi)
- CloudFront pencatatan (CloudFront dokumentasi)
- Informasi Amazon S3
- Informasi AWS Lambda

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Pilih alur autentikasi Amazon Cognito untuk aplikasi perusahaan

Michael Daehnert dan Fabian Jahnke, Amazon Web Services

Ringkasan

Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler. Ini menawarkan fitur bermanfaat untuk otentikasi identitas federasi. Untuk memulai dan menjalankannya, arsitek teknis perlu memutuskan bagaimana mereka ingin menggunakan fitur-fitur tersebut.

Amazon Cognito mendukung beberapa aliran untuk permintaan otentikasi. Alur ini menentukan bagaimana pengguna Anda dapat memverifikasi identitas mereka. Keputusan tentang alur otentikasi mana yang akan digunakan bergantung pada persyaratan spesifik aplikasi Anda dan dapat menjadi rumit. Pola ini membantu Anda memutuskan alur otentikasi mana yang paling cocok untuk aplikasi perusahaan Anda. Ini mengasumsikan bahwa Anda sudah memiliki pengetahuan dasar tentang Amazon Cognito, OpenID Connect (OIDC), dan federasi, dan ini memandu Anda melalui detail tentang aliran otentikasi federasi yang berbeda.

Solusi ini ditujukan untuk pengambil keputusan teknis. Ini membantu Anda memahami alur otentikasi yang berbeda dan memetakannya ke persyaratan aplikasi Anda. Prospek teknis harus mengumpulkan wawasan yang diperlukan untuk memulai integrasi Amazon Cognito. Karena organisasi perusahaan terutama berfokus pada federasi SAMP, pola ini mencakup deskripsi untuk kumpulan pengguna Amazon Cognito dengan federasi SAMP.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Peran dan izin AWS Identity and Access Management (IAM) dengan akses penuh ke Amazon Cognito
- (Opsional) Akses ke penyedia identitas Anda (IDP), seperti Microsoft Entra ID, Active Directory Federation Service (AD FS), atau Okta
- Keahlian tingkat tinggi untuk aplikasi Anda
- Pengetahuan dasar tentang Amazon Cognito, OpenID Connect (OIDC), dan federasi

Batasan

- Pola ini berfokus pada kumpulan pengguna Amazon Cognito dan penyedia identitas. Untuk informasi tentang kumpulan identitas Amazon Cognito, lihat bagian Informasi tambahan.

Arsitektur

Gunakan tabel berikut untuk membantu Anda memilih alur otentikasi. Informasi lebih lanjut tentang setiap aliran disediakan di bagian ini.

Apakah Anda memerlukan machine-to-machine otentikasi?	Apakah aplikasi Anda aplikasi berbasis web tempat frontend dirender di server?	Apakah aplikasi Anda aplikasi satu halaman (SPA) atau aplikasi frontend berbasis seluler?	Apakah aplikasi Anda memerlukan token penyegaran untuk fitur “tetap masuk”?	Apakah frontend menawarkan mekanisme pengalihan berbasis browser?	Aliran Amazon Cognito yang direkomendasikan
Ya	Tidak	Tidak	Tidak	Tidak	Alur Kredensial Klien
Tidak	Ya	Tidak	Ya	Ya	Alur Kode Otorisasi
Tidak	Tidak	Ya	Ya	Ya	Alur Kode Otorisasi dengan Kunci Bukti untuk Pertukaran Kode (PKCE)
Tidak	Tidak	Tidak	Tidak	Tidak	Alur Kata Sandi Pemilik Sumber Daya*

* Alur Kata Sandi Pemilik Sumber Daya harus digunakan hanya jika benar-benar diperlukan. Untuk informasi selengkapnya, lihat bagian alur Kata Sandi Pemilik Sumber Daya dalam pola ini.

Alur Kredensial Klien

Aliran Kredensial Klien adalah yang terpendek dari aliran Amazon Cognito. Ini harus digunakan jika sistem atau layanan berkomunikasi satu sama lain tanpa interaksi pengguna. Sistem permintaan menggunakan ID klien dan rahasia klien untuk mengambil token akses. Karena kedua sistem bekerja tanpa interaksi pengguna, tidak diperlukan langkah persetujuan tambahan.

Diagram ini menggambarkan hal sebagai berikut:

1. Aplikasi 1 mengirimkan permintaan otentikasi dengan ID klien dan rahasia klien ke titik akhir Amazon Cognito, dan mengambil token akses.
2. Aplikasi 1 menggunakan token akses ini untuk setiap panggilan berikutnya ke Aplikasi 2.
3. Aplikasi 2 memvalidasi token akses dengan Amazon Cognito.

Aliran ini harus digunakan:

- Untuk komunikasi antar aplikasi tanpa interaksi pengguna

Aliran ini tidak boleh digunakan:

- Untuk komunikasi apa pun di mana interaksi pengguna dimungkinkan

Alur Kode Otorisasi

Alur Kode Otorisasi adalah untuk otentikasi berbasis web klasik. Dalam alur ini, backend menangani semua pertukaran dan penyimpanan token. Klien berbasis browser tidak melihat token yang sebenarnya. Solusi ini digunakan untuk aplikasi yang ditulis dalam kerangka kerja seperti .NET Core, Jakarta Faces, atau Jakarta Server Pages (JSP).

Alur Kode Otorisasi adalah aliran berbasis pengalihan. Klien harus dapat berinteraksi dengan browser web atau klien serupa. Klien dialihkan ke server otentikasi dan mengautentikasi terhadap server ini. Jika klien berhasil mengautentikasi, itu dialihkan kembali ke server.

Diagram ini menggambarkan hal sebagai berikut:

1. Klien mengirimkan permintaan ke server web.
2. Server web mengarahkan klien ke Amazon Cognito dengan menggunakan kode status HTTP 302. Klien secara otomatis mengikuti pengalihan ini ke login iDP yang dikonfigurasi.
3. IdP memeriksa sesi browser yang ada di sisi iDP. Jika tidak ada, pengguna menerima prompt untuk mengautentikasi dengan memberikan nama pengguna dan kata sandi mereka. IDP merespons dengan token SAMP ke Amazon Cognito.
4. Amazon Cognito mengembalikan kesuksesan dengan token web JSON (JWT), khususnya token kode. Server web memanggil /oauth2/token untuk menukar token kode dengan token akses. Server web mengirimkan ID klien dan rahasia klien ke Amazon Cognito untuk validasi.
5. Token akses digunakan untuk setiap panggilan berikutnya ke aplikasi lain.
6. Aplikasi lain memvalidasi token akses dengan Amazon Cognito.

Aliran ini harus digunakan:

- Jika pengguna dapat berinteraksi dengan browser web atau klien. Kode aplikasi dijalankan dan dirender di server untuk memastikan bahwa tidak ada rahasia yang terpapar ke browser.

Aliran ini tidak boleh digunakan:

- Untuk aplikasi satu halaman (SPAs) atau aplikasi seluler karena mereka dirender pada klien dan tidak boleh menggunakan rahasia klien.

Alur Kode Otorisasi dengan PKCE

Aliran Kode Otorisasi dengan Kunci Bukti untuk Pertukaran Kode (PKCE) harus digunakan untuk aplikasi satu halaman dan aplikasi seluler. Ini adalah penerus aliran Implisit dan lebih aman karena menggunakan PKCE. PKCE adalah perpanjangan dari hibah kode otorisasi OAuth 2.0 untuk klien publik. PKCE menjaga terhadap penebusan kode otorisasi yang dicegat.

Diagram ini menggambarkan hal sebagai berikut:

1. Aplikasi ini membuat pemverifikasi kode dan tantangan kode. Ini adalah nilai unik yang didefinisikan dengan baik yang dikirim ke Amazon Cognito untuk referensi di masa mendatang.
2. Aplikasi memanggil titik akhir /oauth2/authorization dari Amazon Cognito. Ini secara otomatis mengarahkan pengguna ke login iDP yang dikonfigurasi.

3. IdP memeriksa sesi yang ada. Jika tidak ada, pengguna menerima prompt untuk mengautentikasi dengan memberikan nama pengguna dan kata sandi mereka. IDP merespons dengan token SAMP ke Amazon Cognito.
4. Setelah Amazon Cognito mengembalikan kesuksesan dengan token kode, server web memanggil /oauth2/token untuk menukar token kode dengan token akses.
5. Token akses digunakan untuk setiap panggilan berikutnya ke aplikasi lain.
6. Aplikasi lain memvalidasi token akses dengan Amazon Cognito.

Aliran ini harus digunakan:

- Untuk SPAs atau aplikasi seluler

Aliran ini tidak boleh digunakan:

- Jika backend aplikasi menangani otentikasi

Alur Kata Sandi Pemilik Sumber Daya

Alur Kata Sandi Pemilik Sumber Daya ditujukan untuk aplikasi tanpa kemampuan pengalihan. Itu dibangun dengan membuat formulir login di aplikasi Anda sendiri. Login diperiksa di Amazon Cognito melalui panggilan CLI atau SDK alih-alih mengandalkan aliran pengalihan. Federasi tidak dimungkinkan dalam alur otentikasi ini karena federasi memerlukan pengalihan berbasis browser.

Diagram ini menggambarkan hal sebagai berikut:

1. Pengguna memasukkan kredensialnya pada formulir login yang disediakan oleh aplikasi.
2. AWS Command Line Interface (AWS CLI) melakukan admin-initiated-authpanggilan ke Amazon Cognito.

> **Note**
>
> Atau, Anda dapat menggunakan AWS SDKs alih-alih AWS CLI.

3. Amazon Cognito mengembalikan token akses.
4. Token akses digunakan untuk setiap panggilan berikutnya ke aplikasi lain.
5. Aplikasi lain memvalidasi token akses dengan Amazon Cognito.

Aliran ini harus digunakan:

- Saat memigrasikan klien yang ada yang menggunakan logika autentikasi langsung (seperti otentikasi akses dasar atau otentikasi akses intisari) ke OAuth dengan mengonversi kredensi yang disimpan menjadi token akses

Aliran ini tidak boleh digunakan:

- Jika Anda ingin menggunakan identitas federasi
- Jika aplikasi Anda mendukung pengalihan

Alat

Layanan AWS

- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.

Alat-alat lainnya

- Debugger token web JSON (JWT) adalah alat validasi JWT berbasis web.

Epik

Nilai aplikasi Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan persyaratan otentikasi.	Nilai aplikasi Anda sesuai dengan persyaratan otentikasi spesifik Anda.	Pengembang aplikasi, Arsitek aplikasi
Sejajarkan persyaratan dengan alur otentikasi.	Di bagian Arsitektur, gunakan tabel keputusan dan penjelasan dari setiap alur untuk memilih alur autentikasi Amazon Cognito Anda.	Pengembang aplikasi, AWS Umum, Arsitek aplikasi

Siapkan kumpulan pengguna Amazon Cognito

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat pengguna.	1. Masuk ke AWS Management Console, lalu buka konsol Amazon Cognito. 2. Buat kumpulan pengguna Cognito baru. Untuk petunjuknya, lihat kumpulan pengguna Amazon Cognito. 3. Perbarui pengaturan dan atribut kumpulan pengguna sesuai kebutuhan. Misalnya, tetapkan kebijakan kata sandi untuk kumpulan pengguna. Jangan membuat klien aplikasi.	AWS Umum
(Opsional) Konfigurasikan penyedia identitas.	1. Buat penyedia identitas SAMP di kumpulan pengguna Amazon Cognito. Untuk petunjuk, lihat Menambahkan dan mengelola penyedia identitas SAMP di kumpulan pengguna. 2. Konfigurasikan penyedia identitas SAMP pihak ketiga Anda untuk bekerja dengan federasi untuk kumpulan pengguna Amazon Cognito. Untuk informasi selengkap nya, lihat Mengonfigurasi	AWS Umum, Administrator Federasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	penyedia identitas SAMP pihak ketiga Anda. Jika Anda menggunakan AD FS, lihat Membangun Federasi AD FS untuk Aplikasi Web Anda menggunakan Kumpulan Pengguna Amazon Cognito (postingan blog AWS).	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat klien aplikasi.	1. Buat klien aplikasi untuk kumpulan pengguna. Untuk petunjuknya, lihat Membuat klien aplikasi. Perhatikan hal-hal berikut: • Ubah pengaturan sesuai kebutuhan, seperti kedaluwarsa token. • Jika alur otentikasi Anda tidak memerlukan rahasia klien, kosongkan kotak centang Hasilkan rahasia klien. 2. Pilih Pengaturan klien aplikasi untuk mengubah integrasinya ke login kumpulan pengguna (nama pengguna dan kata sandi) atau login gabungan melalui IDP berbasis SAMP. 3. Aktifkan IDP Anda dengan mendefinisikan URLs dan mendefinisikan OAuth alur atau cakupan sesuai kebutuhan.	AWS Umum

Integrasikan aplikasi dengan Amazon Cognito

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tukarkan detail integrasi Amazon Cognito.	Bergantung pada alur autentikasi Anda, bagikan informasi Amazon Cognito	Pengembang aplikasi, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dengan aplikasi, seperti ID kumpulan pengguna dan ID klien aplikasi.	
Menerapkan otentikasi Amazon Cognito.	Ini tergantung pada alur otentikasi yang Anda pilih, bahasa pemrograman Anda, dan kerangka kerja yang Anda gunakan. Untuk beberapa tautan untuk memulai, lihat bagian Sumber daya terkait.	Pengembang aplikasi

Sumber daya terkait

Dokumentasi AWS

- Alur otentikasi kumpulan pengguna
- Memverifikasi token web JSON
- Akses layanan AWS dari aplikasi ASP.NET Core menggunakan kumpulan identitas Amazon Cognito
- Kerangka kerja dan SDKs:
  - Amazon Amplify otentikasi
  - Contoh Penyedia Identitas Amazon Cognito (dokumentasi AWS SDK for Java 2.x)
  - Mengautentikasi pengguna dengan Amazon Cognito (dokumentasi AWS SDK for .NET)

Postingan blog AWS

- Otorisasi @Edge menggunakan cookie: Lindungi CloudFront konten Amazon Anda agar tidak diunduh oleh pengguna yang tidak diautentikasi
- Membangun Federasi AD FS untuk Aplikasi Web Anda menggunakan Kumpulan Pengguna Amazon Cognito

Mitra implementasi

- AWS Partners untuk solusi otentikasi

Informasi tambahan

FAQ

Mengapa aliran Implisit tidak digunakan lagi?

Sejak rilis framework OAuth 2.1, aliran Implicit ditandai sebagai usang karena alasan keamanan. Sebagai alternatif, silakan gunakan alur Kode Otorisasi dengan PKCE yang dijelaskan di bagian Arsitektur.

Bagaimana jika Amazon Cognito tidak menawarkan beberapa fungsionalitas yang saya butuhkan?

AWS Partners menawarkan integrasi yang berbeda untuk solusi otentikasi dan otorisasi. Untuk informasi selengkapnya, lihat AWS Partners untuk solusi autentikasi.

Bagaimana dengan aliran kumpulan identitas Amazon Cognito?

Kumpulan pengguna Amazon Cognito dan identitas federasi adalah untuk otentikasi. Kumpulan identitas Amazon Cognito digunakan untuk otorisasi akses sumber daya AWS dengan meminta kredensil AWS sementara. Token ID dan pertukaran token akses untuk kumpulan identitas tidak dibahas dalam pola ini. Untuk informasi selengkapnya, lihat Apa perbedaan antara kumpulan pengguna Amazon Cognito dan kumpulan identitas dan skenario Amazon Cognito Umum.

Langkah selanjutnya

Pola ini memberikan gambaran umum tentang alur otentikasi Amazon Cognito. Sebagai langkah selanjutnya, implementasi rinci untuk bahasa pemrograman aplikasi perlu dipilih. Beberapa penawaran bahasa SDKs dan kerangka kerja, yang dapat Anda gunakan dengan Amazon Cognito. Untuk referensi yang bermanfaat, lihat bagian Sumber daya terkait.

Buat aturan AWS Config khusus dengan menggunakan AWS CloudFormation Guard kebijakan

Andrew Lok, Nicole Brown, Kailash Havildar, dan Tanya Howell, Amazon Web Services

Ringkasan

AWS Configaturan membantu Anda mengevaluasi AWS sumber daya dan status konfigurasi targetnya. Ada dua jenis AWS Config aturan: dikelola dan kustom. Anda dapat membuat aturan kustom dengan AWS Lambda fungsi atau dengan AWS CloudFormation Guard(GitHub), policy-as-code bahasa.

Aturan yang dibuat dengan Guard memberikan kontrol yang lebih terperinci daripada aturan terkelola, dan biasanya lebih mudah dikonfigurasi daripada aturan Lambda yang sepenuhnya disesuaikan. Pendekatan ini memberi insinyur dan arsitek kemampuan untuk membangun aturan tanpa perlu mengetahui Python, NodeJS, atau Java, yang diperlukan untuk menerapkan aturan khusus melalui Lambda.

Pola ini menyediakan templat yang dapat diterapkan, contoh kode, dan pendekatan penerapan untuk membantu Anda mengadopsi aturan khusus dengan Guard. Dengan menggunakan pola ini, administrator dapat menggunakan AWS Config untuk membuat aturan kepatuhan khusus yang memiliki atribut item konfigurasi. Misalnya, pengembang dapat menggunakan kebijakan Guard terhadap item AWS Config konfigurasi untuk terus memantau status penerapan AWS dan AWS non-sumber daya, mendeteksi pelanggaran aturan, dan secara otomatis memulai remediasi.

Tujuan

Setelah membaca pola ini, Anda harus dapat:

- Memahami bagaimana kode kebijakan Guard berinteraksi dengan AWS Config layanan.
- Terapkan Skenario 1, yang merupakan aturan AWS Config khusus yang menggunakan sintaks Guard untuk memvalidasi kepatuhan untuk volume terenkripsi. Aturan ini memverifikasi bahwa drive sedang digunakan dan memverifikasi bahwa jenis drive adalah gp3.
- Terapkan Skenario 2, yang merupakan aturan AWS Config khusus yang menggunakan sintaks Guard untuk memvalidasi kepatuhan Amazon GuardDuty . Aturan ini memverifikasi bahwa GuardDuty perekam mengaktifkan Perlindungan Amazon Simple Storage Service (Amazon S3) dan Perlindungan Amazon Elastic Kubernetes Service (Amazon EKS).

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Config, atur di Anda Akun AWS

Batasan

- Aturan kustom penjaga hanya dapat menanyakan pasangan kunci-nilai dalam item konfigurasi target catatan JSON

Arsitektur

Anda menerapkan sintaks Guard ke AWS Config aturan sebagai kebijakan kustom. AWS Config menangkap JSON hierarkis dari masing-masing sumber daya yang ditentukan. JSON dari item AWS Config konfigurasi berisi pasangan kunci-nilai. Atribut ini digunakan dalam sintaks Guard sebagai variabel yang ditetapkan ke nilai yang sesuai.

Berikut ini adalah penjelasan dari sintaks Guard. Variabel dari item konfigurasi JSON digunakan dan ditambahkan dengan karakter. `%`

```
# declare variable
let <variable name> = <'value'>

# create rule and assign condition and policy
    rule <rule name> when
        <CI json key> == <"CI json value"> {
            <top level CI json key>.<next level CI json key> == %<variable name>
        }
```

Skenario 1: Volume Amazon EBS

Skenario 1 menerapkan aturan AWS Config kustom yang menggunakan sintaks Guard untuk memvalidasi kepatuhan untuk volume terenkripsi. Aturan ini memverifikasi bahwa drive sedang digunakan dan memverifikasi bahwa jenis drive adalah gp3.

Berikut ini adalah contoh item AWS Config konfigurasi untuk skenario 1. Ada tiga pasangan kunci-nilai dalam item konfigurasi ini yang digunakan sebagai variabel dalam kebijakan

Guard:`volumestatus`,`volumeencryptionstatus`, dan. `volumetype` Juga, `resourceType` kunci digunakan sebagai filter dalam kebijakan Penjaga.


```
{
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2023-01-15T19:04:45.402Z",
  "configurationItemStatus": "ResourceDiscovered",
  "configurationStateId": "4444444444444",
  "configurationItemMD5Hash": "",
  "arn": "arn:aws:ec2:us-west-2:111111111111:volume/vol-222222222222",
  "resourceType": "AWS::EC2::Volume",
  "resourceId": "vol-222222222222",
  "awsRegion": "us-west-2",
  "availabilityZone": "us-west-2b",
  "resourceCreationTime": "2023-01-15T19:03:22.247Z",
  "tags": {},
  "relatedEvents": [],
  "relationships": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-33333333333333333",
      "relationshipName": "Is attached to Instance"
    }
  ],
  "configuration": {
    "attachments": [
      {
        "attachTime": "2023-01-15T19:03:22.000Z",
        "device": "/dev/xvda",
        "instanceId": "i-33333333333333333",
        "state": "attached",
        "volumeId": "vol-222222222222",
        "deleteOnTermination": true,
        "associatedResource": null,
        "instanceOwningService": null
      }
    ],
    "availabilityZone": "us-west-2b",
    "createTime": "2023-01-15T19:03:22.247Z",
    "encrypted": false,
    "kmsKeyId": null,
    "outpostArn": null,
```

```
    "size": 8,
    "snapshotId": "snap-55555555555555555",
    "state": "in-use",
    "volumeId": "vol-222222222222",
    "iops": 100,
    "tags": [],
    "volumeType": "gp2",
    "fastRestored": null,
    "multiAttachEnabled": false,
    "throughput": null,
    "sseType": null
  },
  "supplementaryConfiguration": {}
}
```

Berikut ini adalah contoh menggunakan sintaks Guard untuk menentukan variabel dan aturan dalam skenario 1. Dalam contoh berikut:

- Tiga baris pertama mendefinisikan variabel dengan menggunakan `let` perintah. Mereka diberi nama dan nilai yang berasal dari atribut item konfigurasi.
- Blok `compliancecheck` aturan menambahkan dependensi kondisional when yang mencari pasangan `resourceType` kunci-nilai yang cocok. `AWS::EC2::Volume` Jika kecocokan ditemukan, aturan berlanjut melalui atribut JSON lainnya dan mencari kecocokan pada tiga kondisi berikut:`state`,`encrypted`, dan. `volumeType`

```
let volumestatus = 'available'
let volumetype = 'gp3'
let volumeencryptionstatus = true

    rule compliancecheck when
        resourceType == "AWS::EC2::Volume" {
            configuration.state == %volumestatus
            configuration.encrypted == %volumeencryptionstatus
            configuration.volumeType == %volumetype
        }
```

Untuk kebijakan kustom Guard lengkap yang mengimplementasikan aturan kustom ini, lihat awsconfig-guard-cft.yaml atau awsconfig-guard-tf-ec2vol.json di repositori kode. GitHub Untuk kode HashiCorp Terraform yang menerapkan kebijakan kustom ini di Guard, lihat awsconfig-guard-tf-example.json di repositori kode.

Skenario 2: GuardDuty kepatuhan

Skenario 2 menerapkan aturan AWS Config khusus yang menggunakan sintaks Guard untuk memvalidasi kepatuhan Amazon GuardDuty . Aturan ini memverifikasi bahwa GuardDuty perekam mengaktifkan Perlindungan Amazon S3 dan Perlindungan Amazon EKS. Ini juga memverifikasi bahwa GuardDuty temuan dipublikasikan setiap 15 menit. Skenario ini dapat diterapkan di semua Akun AWS dan Wilayah AWS dalam organisasi (in AWS Organizations).

Berikut ini adalah contoh item AWS Config konfigurasi untuk skenario 2. Ada tiga pasangan kunci-nilai dalam item konfigurasi ini yang digunakan sebagai variabel dalam kebijakan Guard:`FindingPublishingFrequency`,`S3Logs`, dan. `Kubernetes` Juga, `resourceType` kunci digunakan sebagai filter dalam kebijakan.

```
{
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2023-11-27T13:34:28.888Z",
  "configurationItemStatus": "OK",
  "configurationStateId": "7777777777777",
  "configurationItemMD5Hash": "",
  "arn": "arn:aws:guardduty:us-
west-2:111111111111:detector/66666666666666666666666666666666",
  "resourceType": "AWS::GuardDuty::Detector",
  "resourceId": "66666666666666666666666666666666",
  "resourceName": "66666666666666666666666666666666",
  "awsRegion": "us-west-2",
  "availabilityZone": "Regional",
  "resourceCreationTime": "2020-02-17T02:48:04.511Z",
  "tags": {},
  "relatedEvents": [],
  "relationships": [],
  "configuration": {
    "Enable": true,
    "FindingPublishingFrequency": "FIFTEEN_MINUTES",
    "DataSources": {
      "S3Logs": {
        "Enable": true
      },
      "Kubernetes": {
        "AuditLogs": {
          "Enable": true
        }
```

```
      }
    },

    "Id": "66666666666666666666666666666666",
    "Tags": []
  },
  "supplementaryConfiguration": {
    "CreatedAt": "2020-02-17T02:48:04.511Z"
  }
}
```

Berikut ini adalah contoh menggunakan sintaks Guard untuk menentukan variabel dan aturan dalam skenario 2. Dalam contoh berikut:

- Tiga baris pertama mendefinisikan variabel dengan menggunakan `let` perintah. Mereka diberi nama dan nilai yang berasal dari atribut item konfigurasi.
- Blok `compliancecheck` aturan menambahkan dependensi kondisional when yang mencari pasangan `resourceType` kunci-nilai yang cocok. `AWS::GuardDuty::Detector` Jika kecocokan ditemukan, aturan berlanjut melalui atribut JSON lainnya dan mencari kecocokan pada tiga kondisi berikut:`S3Logs.Enable`,`Kubernetes.AuditLogs.Enable`, dan. `FindingPublishingFrequency`

```
let s3protection = true
let kubernetesprotection = true
let publishfrequency = 'FIFTEEN_MINUTES'

    rule compliancecheck when
        resourceType == "AWS::GuardDuty::Detector" {
            configuration.DataSources.S3Logs.Enable == %s3protection
            configuration.DataSources.Kubernetes.AuditLogs.Enable ==
 %kubernetesprotection
            configuration.FindingPublishingFrequency == %publishfrequency
        }
```

Untuk kebijakan kustom Guard lengkap yang mengimplementasikan aturan kustom ini, lihat awsconfig-guard-cft-gd.yaml di repositori kode. GitHub Untuk kode HashiCorp Terraform yang menerapkan kebijakan kustom ini di Guard, lihat awsconfig-guard-tf-gd.json di repositori kode.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu.

Alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.

Repositori kode

Kode untuk pola ini tersedia di AWS CloudFormation Guard repositori GitHub AWS Config with. Repositori kode ini berisi sampel untuk kedua skenario yang dijelaskan dalam pola ini.

Epik

Membuat aturan AWS Config khusus

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Pilih pasangan kunci-nilai untuk aturan.	Selesaikan langkah-langkah ini jika Anda mendefinisikan kebijakan Penjaga kustom. Jika Anda menggunakan salah satu contoh kebijakan untuk skenario 1 atau 2, lewati langkah-langkah ini. 1. Masuk ke Konsol Manajemen AWS dan buka AWS Config konsol.	Administrator AWS, Insinyur keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di navigasi kiri, pilih Resources. 3. Dalam inventaris sumber daya, pilih jenis sumber daya yang ingin Anda buat aturan AWS Config kustom. 4. Pilih Lihat detail. 5. Pilih Lihat Item Konfigurasi (JSON). Bagian ini diperluas untuk menampilkan item konfigurasi dalam format JSON. 6. Identifikasi pasangan nilai kunci yang ingin Anda buat aturan AWS Config khusus.	
Buat aturan khusus.	Menggunakan pasangan kunci-nilai yang Anda identifikasi sebelumnya atau menggunakan salah satu contoh kebijakan Guard yang disediakan, ikuti petunjuk dalam Membuat Aturan Kebijakan AWS Config Kustom untuk membuat aturan kustom.	Administrator AWS, Insinyur keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi aturan kustom.	Lakukan salah satu hal berikut untuk memvalidasi aturan Penjaga kustom: • Masukkan perintah berikut di AWS Command Line Interface (AWS CLI). `cfn-guard validate -r guard-s3.guard -d s3bucket-prod-pass.json` • Ikuti instruksi dalam mode Detektif dalam Mengevaluasi Sumber Daya Anda dengan AWS Config Aturan untuk menerapkan aturan. AWS Config Konfirmasikan bahwa sintaks Guard cocok dengan benar dengan sumber daya yang sesuai di akun atau file target.	Administrator AWS, Insinyur keamanan

Pemecahan masalah

Isu	Solusi
Uji kebijakan Penjaga di luar AWS Config	Pengujian unit dapat dilakukan pada perangkat lokal Anda atau dalam lingkungan pengembangan terintegrasi (IDE), seperti AWS Cloud9 IDE. Untuk melakukan pengujian unit, lakukan hal berikut:

Isu	Solusi

	1. Instal AWS CloudFormation Guard CLI dan dependensinya.
	2. Simpan sampel CI berformat JSON ke workstation Anda sebagai file.json.
	3. Simpan GuardDuty kebijakan ke workstation Anda sebagai file.guard.
	4. Di CLI Penjaga, masukkan perintah berikut untuk memvalidasi file JSON sampel dengan menggunakan kebijakan Guard.
	`cfn-guard validate \`
	`-r guard-s3.guard \`
	`-d s3bucket-prod-pass.json`

Debug aturan AWS Config kustom	Dalam kebijakan Penjaga Anda, ubah `EnableDebugLogDelivery` nilainya menjadi`true`. Nilai default-nya adalah `false`. Pesan log disimpan di Amazon CloudWatch.

Sumber daya terkait

AWS dokumentasi

- Membuat Aturan Kebijakan AWS Config Kustom (AWS Config dokumentasi)
- AWS CloudFormation Guard Aturan penulisan (Dokumentasi penjaga)

AWS posting blog dan lokakarya

- Memperkenalkan AWS CloudFormation Guard 2.0 (posting AWS blog)

Sumber daya lainnya

- AWS CloudFormation Guard (GitHub)
- AWS CloudFormation Guard Dokumentasi CLI () GitHub

Buat laporan konsolidasi temuan keamanan Prowler dari beberapa Akun AWS

Mike Virgilio, Jay Durga, dan Andrea Di Fabio, Amazon Web Services

Ringkasan

Prowler (GitHub) adalah alat baris perintah sumber terbuka yang dapat membantu Anda menilai, mengaudit, dan memantau akun Amazon Web Services (AWS) Anda untuk kepatuhan terhadap praktik terbaik keamanan. Dalam pola ini, Anda menyebarkan Prowler di terpusat Akun AWS di organisasi Anda, dikelola oleh AWS Organizations, dan kemudian menggunakan Prowler untuk melakukan penilaian keamanan semua akun di organisasi.

Meskipun ada banyak metode untuk menyebarkan dan memanfaatkan Prowler untuk penilaian, solusi ini telah dirancang untuk penyebaran cepat, analisis penuh dari semua akun dalam organisasi atau akun target yang ditentukan, dan pelaporan temuan keamanan yang dapat diakses. Dalam solusi ini, ketika Prowler menyelesaikan penilaian keamanan semua akun dalam organisasi, itu mengkonsolidasikan hasilnya. Ini juga menyaring pesan kesalahan yang diharapkan, seperti kesalahan yang terkait dengan pembatasan yang mencegah Prowler memindai bucket Amazon Simple Storage Service (Amazon S3) di akun yang disediakan. AWS Control Tower Hasil yang disaring dan terkonsolidasi dilaporkan dalam template Microsoft Excel yang disertakan dengan pola ini. Anda dapat menggunakan laporan ini untuk mengidentifikasi potensi peningkatan kontrol keamanan di organisasi Anda.

Solusi ini dirancang dengan mempertimbangkan hal-hal berikut:

- AWS CloudFormation Template mengurangi upaya yang diperlukan untuk menyebarkan AWS sumber daya dalam pola ini.
- Anda dapat menyesuaikan parameter dalam CloudFormation template dan skrip prowler_scan.sh pada saat penerapan untuk menyesuaikan template untuk lingkungan Anda.
- Penilaian prowler dan kecepatan pelaporan dioptimalkan melalui pemrosesan paralel, hasil agregat Akun AWS, pelaporan terkonsolidasi dengan remediasi yang direkomendasikan, dan visualisasi yang dihasilkan secara otomatis.
- Pengguna tidak perlu memantau kemajuan pemindaian. Ketika penilaian selesai, pengguna diberi tahu melalui topik Amazon Simple Notification Service (Amazon SNS) sehingga mereka dapat mengambil laporan.

- Template laporan membantu Anda membaca dan menilai hanya hasil yang relevan untuk seluruh organisasi Anda.

Prasyarat dan batasan

Prasyarat

- Sebuah Akun AWS untuk layanan dan alat keamanan hosting, dikelola sebagai akun anggota organisasi di AWS Organizations. Dalam pola ini, akun ini disebut sebagai akun keamanan.
- Di akun keamanan, Anda harus memiliki subnet pribadi dengan akses internet keluar. Untuk petunjuk, lihat VPC dengan server di subnet pribadi dan NAT di dokumentasi Amazon Virtual Private Cloud (Amazon VPC). Anda dapat membuat akses internet dengan menggunakan gateway NAT yang disediakan di subnet publik.
- Akses ke akun AWS Organizations manajemen atau akun yang telah mendelegasikan izin administrator untuk. CloudFormation Untuk petunjuk, lihat Mendaftarkan administrator yang didelegasikan dalam CloudFormation dokumentasi.
- Aktifkan akses tepercaya antara AWS Organizations dan CloudFormation. Untuk petunjuk, lihat Mengaktifkan akses tepercaya dengan AWS Organizations dalam CloudFormation dokumentasi.

Batasan

- Target Akun AWS harus dikelola sebagai organisasi di AWS Organizations. Jika Anda tidak menggunakan AWS Organizations, Anda dapat memperbarui CloudFormation template IAM-ProwlerExecRole .yaml dan skrip prowler_scan.sh untuk lingkungan Anda. Sebagai gantinya, Anda memberikan daftar Akun AWS IDs dan Wilayah tempat Anda ingin menjalankan skrip.
- CloudFormation Template ini dirancang untuk menyebarkan instans Amazon Elastic Compute Cloud EC2 (Amazon) di subnet pribadi yang memiliki akses internet keluar. AWS Systems Manager Agen (Agen SSM) memerlukan akses keluar untuk mencapai titik akhir AWS Systems Manager layanan, dan Anda memerlukan akses keluar untuk mengkloning repositori kode dan menginstal dependensi. Jika Anda ingin menggunakan subnet publik, Anda harus memodifikasi template prowler-resources.yaml untuk mengaitkan alamat IP Elastis dengan instance. EC2

Versi produk

- Prowler versi 4.0 atau yang lebih baru

Arsitektur

Diagram menunjukkan proses berikut:

1. Menggunakan Session Manager, kemampuan AWS Systems Manager, pengguna mengautentikasi ke EC2 instance dan menjalankan skrip prowler_scan.sh. Skrip shell ini melakukan langkah 2—8.
2. EC2 Instance mengasumsikan peran `ProwlerEC2Role` IAM, yang memberikan izin untuk mengakses bucket S3 dan untuk mengambil peran `ProwlerExecRole` IAM di akun lain di organisasi.
3. EC2 Instance mengasumsikan peran `ProwlerExecRole` IAM dalam akun manajemen organisasi dan menghasilkan daftar akun dalam organisasi.
4. EC2 Instance mengasumsikan peran `ProwlerExecRole` IAM dalam akun anggota organisasi (disebut akun beban kerja dalam diagram arsitektur) dan melakukan penilaian keamanan di setiap akun. Temuan disimpan sebagai file CSV dan HTML pada EC2 instance.

   > **Note**
   >
   > File HTML adalah output dari penilaian Prowler. Karena sifat HTML, mereka tidak digabungkan, diproses, atau digunakan secara langsung dalam pola ini. Namun, ini mungkin berguna untuk tinjauan laporan akun individu.

5. EC2 Instance memproses semua file CSV untuk menghapus kesalahan yang diketahui dan diharapkan dan mengkonsolidasikan temuan yang tersisa ke dalam satu file CSV.
6. EC2 Instance mengemas hasil akun individual dan hasil agregat ke dalam file zip.
7. EC2 Instance mengunggah file zip ke bucket S3.
8. EventBridge Aturan mendeteksi unggahan file dan menggunakan topik Amazon SNS untuk mengirim email ke pengguna yang memberi tahu mereka bahwa penilaian telah selesai.
9. Pengguna mengunduh file zip dari bucket S3. Pengguna mengimpor hasil ke template Excel dan meninjau hasilnya.

Alat

Layanan AWS

- Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat meluncurkan server virtual sebanyak yang Anda butuhkan dan dengan cepat meningkatkannya ke atas atau ke bawah.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data real-time dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar. Pola ini menggunakan Session Manger, kemampuan Systems Manager.

Alat lainnya

- Prowler adalah alat baris perintah sumber terbuka yang membantu Anda menilai, mengaudit, dan memantau akun Anda untuk kepatuhan terhadap praktik terbaik AWS keamanan serta kerangka kerja dan standar keamanan lainnya.

Repositori kode

Kode untuk pola ini tersedia dalam Penilaian Keamanan GitHub Multi-Akun melalui repositori Prowler. Repositori kode berisi file-file berikut:

- prowler_scan.sh - Skrip bash ini digunakan untuk memulai penilaian keamanan Prowler dari beberapa, Akun AWS secara paralel. Seperti yang didefinisikan dalam Prowler-Resources.yaml

CloudFormationtemplate, skrip ini secara otomatis digunakan ke folder pada instance. `usr/local/prowler` EC2

- Prowler-Resources.yaml — Anda menggunakan CloudFormation template ini untuk membuat tumpukan di akun keamanan di organisasi. Template ini menyebarkan semua sumber daya yang diperlukan untuk akun ini untuk mendukung solusinya. Tumpukan ini harus digunakan sebelum template IAM- ProwlerExecRole .yaml. Kami tidak menyarankan Anda menyebarkan sumber daya ini di akun yang menampung beban kerja produksi penting.

> **Note**
>
> Jika tumpukan ini dihapus dan digunakan kembali, Anda harus membangun kembali kumpulan `ProwlerExecRole` tumpukan untuk membangun kembali dependensi lintas akun di antara peran IAM.

- IAM- ProwlerExecRole .yaml - Anda menggunakan CloudFormation templat ini untuk membuat kumpulan tumpukan yang menerapkan peran `ProwlerExecRole` IAM di semua akun di organisasi, termasuk akun manajemen.
- prowler-report-template.xlsm — Anda menggunakan template Excel ini untuk memproses temuan Prowler. Tabel pivot dalam laporan menyediakan kemampuan pencarian, bagan, dan temuan terkonsolidasi.

Epik

Mempersiapkan penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode.	1. Di antarmuka baris perintah, ubah direktori kerja Anda ke lokasi tempat Anda ingin menyimpan file sampel. 2. Masukkan perintah berikut: `git clone https://github.com/aws-samples/multi-account-`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`security-assessmen t-via-prowler.git`	
Tinjau template.	1. Di repositori kloning, buka file Prowler-Resources.yaml dan IAM- .yaml. ProwlerEx ecRole 2. Tinjau sumber daya yang dibuat oleh templat ini dan sesuaikan templat sesuai kebutuhan untuk lingkunga n Anda. Untuk informasi selengkapnya, lihat Bekerja dengan templat dalam CloudFormation dokumenta si. 3. Simpan dan tutup file Prowler-Resources.yaml dan IAM- .yaml. ProwlerEx ecRole	AWS DevOps

Buat CloudFormation tumpukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan sumber daya di akun keamanan.	Menggunakan template prowler-resources.yaml, Anda membuat CloudFormation tumpukan yang menyebark an semua sumber daya yang diperlukan di akun keamanan. Untuk petunjuk, lihat Membuat tumpukan dalam CloudForm ation dokumentasi. Perhatika	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	n hal berikut saat menerapkan template ini: 1. Pada halaman Specify template, pilih Template is ready, lalu upload file prowler-resources.yaml. 2. Pada halaman Tentukan detail tumpukan, di kotak Nama tumpukan, masukkan`Prowler-Resources` . 3. Di bagian Parameter, masukkan yang berikut ini: • `VPCId`— Pilih VPC di akun. • `SubnetId`— Pilih subnet pribadi yang memiliki akses internet. Catatan: Jika Anda memilih subnet publik, EC2 instans tidak akan diberi alamat IP publik karena CloudFormation templat, secara default, tidak menyediakan dan melampirkan alamat IP Elastis. • `InstanceType` — Pilih ukuran instans berdasarkan jumlah penilaian paralel:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Untuk 10, pilih`r6i.large` . • Untuk 12, pilih`r6i.xlarge` . • Untuk 14-18, pilih. `r6i.2xlarge` • `InstanceImageId` — Tinggalkan default untuk Amazon Linux. • `KeyPairName` — Jika Anda menggunakan SSH untuk akses, tentukan nama key pair yang ada. • `PermittedSSHInbound` — Jika Anda menggunakan SSH untuk akses, tentukan blok CIDR yang diizinkan. Jika Anda tidak menggunakan SSH, pertahankan nilai default. `127.0.0.1` • `BucketName` — Nilai defaultnya adalah`prowler-output-<accountID>-<region>` . Anda dapat memodifikasi ini sesuai kebutuhan. Jika Anda menentukan nilai kustom, ID akun dan Wilayah secara otomatis	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ditambahkan ke nilai yang ditentukan. • `EmailAddress` — Tentukan alamat email untuk notifikasi Amazon SNS saat Prowler menyelesaikan penilaian dan mengunggah file.zip ke bucket S3. Catatan: Konfigurasi langganan SNS harus dikonfirmasi sebelum Prowler menyelesaikan penilaian atau pemberitahuan tidak akan dikirim. • `IAMProwlerEC2Role` — Pertahankan default kecuali konvensi penamaan Anda memerlukan nama yang berbeda untuk peran IAM ini. • `IAMProwlerExecRole` — Pertahankan default kecuali nama lain akan digunakan saat menerapkan file IAM-ProwlerExecRole .yaml. • `Parallelism` — Tentukan jumlah penilaian paralel yang akan dilakukan. Pastikan bahwa nilai dalam	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`InstanceType` parameter mendukung jumlah penilaian paralel ini. • `FindingOutput` — Jika Anda ingin mengecualikan hasil lulus, pilih`FailOnly`. Ini secara signifikan mengurangi ukuran output dan berfokus pada pemeriksaan yang mungkin perlu diselesaikan. Jika Anda ingin memasukkan hasil lulus, pilih`FailAndPass` . 4. Pada halaman Tinjauan, pilih Sumber daya berikut memerlukan kemampuan: [AWS::IAM::Role], lalu pilih Buat Tumpukan. 5. Setelah tumpukan berhasil dibuat, di CloudFormation konsol, pada tab Output, salin Nama Sumber Daya `ProwlerEC2Role` Amazon (ARN). Anda menggunakan ARN ini nanti saat menerapkan file IAM - .yaml. ProwlerExecRole	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan peran IAM di akun anggota.	Di akun AWS Organizations manajemen atau akun dengan izin administrator yang didelegasikan CloudFormation, gunakan template IAM- ProwlerExecRole .yaml untuk membuat kumpulan tumpukan. CloudFormation Kumpulan tumpukan menyebarkan peran `ProwlerExecRole` IAM di semua akun anggota di organisasi. Untuk petunjukn ya, lihat Membuat kumpulan tumpukan dengan izin yang dikelola layanan dalam dokumentasi. CloudFormation Perhatikan hal berikut saat menerapkan template ini: 1. Di bawah Siapkan template, pilih Template siap, lalu unggah file IAM- ProwlerExecRole .yaml. 2. Pada halaman Tentukan StackSet detail, beri nama kumpulan tumpukan`IAM-ProwlerExecRole` . 3. Di bagian Parameter, masukkan yang berikut ini: • `AuthorizedARN` — Masukkan `ProwlerEC2Role` ARN, yang Anda salin saat Anda membuat	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tumpukan. `Prowler-Resources` • `ProwlerExecRoleName` — Pertahankan nilai default `ProwlerExecRole` kecuali nama lain digunakan saat menerapkan file Prowler-Resources.yaml. 4. Dalam Izin, pilih Izin yang dikelola layanan. 5. Pada halaman Setel opsi penerapan, di bawah target Deployment, pilih Deploy to organization dan terima semua default. Catatan: Jika Anda ingin tumpukan disebarkan ke semua akun anggota secara bersamaan, tetapkan akun bersamaan maksimum dan Toleransi kegagalan ke nilai tinggi, seperti. `100` 6. Di bawah wilayah Deployment, pilih Wilayah AWS tempat EC2 instance untuk Prowler digunakan. Karena sumber daya IAM bersifat global dan bukan Regional, ini menyebarkan peran IAM di semua Wilayah aktif.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	7. Pada halaman Tinjauan, pilih Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM dengan nama khusus, lalu pilih Buat. StackSet 8. Pantau tab Stack instance (untuk status akun individual) dan tab Operasi (untuk status keseluruhan) untuk menentukan kapan penerapan selesai.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyediakan peran IAM dalam akun manajemen.	Menggunakan template IAM-ProwlerExecRole .yaml, Anda membuat CloudFormation tumpukan yang menerapkan peran `ProwlerExecRole` IAM di akun manajemen organisasi. Kumpulan tumpukan yang Anda buat sebelumnya tidak menerapkan peran IAM di akun manajemen . Untuk petunjuk, lihat Membuat tumpukan dalam CloudFormation dokumenta si. Perhatikan hal berikut saat menerapkan template ini: 1. Pada halaman Tentukan template, pilih Template siap, lalu unggah file IAM-ProwlerExecRole .yaml. 2. Pada halaman Tentukan detail tumpukan, di kotak Nama tumpukan, masukkan`IAM-Prowl erExecRole` . 3. Di bagian Parameter, masukkan yang berikut ini: • `AuthorizedARN` — Masukkan `ProwlerEC 2Role` ARN, yang Anda salin saat Anda membuat tumpukan. `Prowler-R esources`	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `ProwlerExecRoleNam e` — Pertahankan nilai default `ProwlerEx ecRole` kecuali nama lain digunakan saat menerapkan file Prowler-Resources.yaml. 4. Pada halaman Tinjauan, pilih Sumber daya berikut memerlukan kemampuan: [AWS::IAM::Role], lalu pilih Buat Tumpukan.	

Lakukan penilaian keamanan Prowler

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan pemindaian.	1. Masuk ke akun keamanan di organisasi. 2. Menggunakan Session Manager, sambungkan ke EC2 instance untuk Prowler yang sebelumnya Anda sediakan. Untuk petunjuk, lihat Connect ke instance Linux Anda menggunak an Session Manager. Jika Anda tidak dapat terhubung , lihat bagian Pemecahan Masalah dari pola ini. 3. Arahkan ke`usr/local /prowler` , lalu buka file prowler_scan.sh.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	4. Tinjau dan modifikasi parameter dan variabel yang dapat disesuaikan dalam skrip ini sesuai kebutuhan untuk lingkungan Anda. Untuk informasi selengkapnya tentang opsi penyesuaian, lihat komentar di awal skrip.
	
	Misalnya, alih-alih mendapatkan daftar semua akun anggota di organisasi dari akun manajemen, Anda dapat memodifikasi skrip untuk menentukan Akun AWS IDs atau Wilayah AWS yang ingin Anda pindai, atau Anda dapat mereferensikan file eksternal yang berisi parameter ini.
	5. Simpan dan tutup file prowler_scan.sh.
	6. Masukkan perintah berikut. Ini menjalankan skrip prowler_scan.sh.
	
	`sudo -i`
	`screen`
	`cd /usr/local/prowler`
	`./prowler_scan.sh`
	
	Perhatikan hal-hal berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Perintah layar memungkinkan skrip untuk terus berjalan jika waktu koneksi habis atau Anda kehilangan akses konsol. • Setelah pemindaian dimulai, Anda dapat memaksa layar terlepas dengan menekan Ctrl +A D. Layar terlepas, dan Anda dapat menutup koneksi instance dan memungkinkan penilaian dilanjutkan. • Untuk melanjutk an sesi terpisah, sambungkan ke instance, masukkan `sudo -i` lalu masukkan`screen -r`. • Untuk memantau kemajuan penilaian akun individual, Anda dapat menavigasi ke `usr/local/prowler` direktori dan memasukka n perintah`tail -f output/stdout-<acc ount-id>` . 7. Tunggu Prowler menyelesa ikan pemindaian di semua akun. Skrip menilai beberapa akun secara bersamaan. Ketika	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	penilaian selesai di semua akun, Anda menerima pemberitahuan jika Anda menentukan alamat email saat Anda menerapkan file Prowler-Resources.yaml.	
Ambil temuan Prowler.	1. Unduh `prowler-output-<assessDate>.zip` file dari `prowler-output-<accountID>-<region>` ember. Untuk petunjuk, lihat Mengunduh objek di dokumentasi Amazon S3. 2. Hapus semua objek di bucket, termasuk file yang Anda unduh. Ini adalah praktik terbaik untuk optimasi biaya dan untuk memastikan bahwa Anda dapat menghapus `Prowler-Resources` CloudFormation tumpukan kapan saja. Untuk petunjuknya, lihat Menghapus objek dalam dokumentasi Amazon S3.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hentikan EC2 instance.	Untuk mencegah penagihan saat instance tidak aktif, hentikan EC2 instance yang menjalankan Prowler. Untuk petunjuknya, lihat Berhenti dan mulai instans Anda di EC2 dokumentasi Amazon.	AWS DevOps

Buat laporan temuan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Impor temuan.	1. Di Excel, buka prowler-report-templatefile.xlsx, lalu pilih lembar kerja Prowler CSV. 2. Hapus semua data sampel, termasuk baris header. Jika Anda ditanya apakah akan menghapus kueri yang terkait dengan data yang dihapus, pilih Tidak. Menghapus kueri dapat memengaruhi fungsionalitas tabel pivot di template Excel. 3. Ekstrak isi file zip yang Anda unduh dari bucket S3. 4. Di Excel, buka prowler-fullorgresults-accessdeniedfiltered.txt. Kami menyarankan Anda menggunakan file ini karena kesalahan	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang paling umum dan tidak dapat ditindaklanjuti telah dihapus, seperti `Access Denied` kesalahan yang terkait dengan upaya pemindaian sumber daya. AWS Control Tower Jika Anda menginginkan temuan tanpa filter, buka file prowler-fullorgresults.txt sebagai gantinya. 5. Pilih kolom A. 6. Jika Anda menggunak an Windows, masukkan Ctrl+C, atau jika Anda menggunakan macOS, masukkan Cmd+C. Ini menyalin semua data ke clipboard. 7. Dalam template laporan Excel, pada lembar kerja Prowler CSV, pilih sel A1. 8. Jika Anda menggunak an Windows, masukkan Ctrl+V, atau jika Anda menggunakan macOS, masukkan Cmd+V. Ini menempelkan temuan ke dalam laporan. 9. Konfirmasikan semua sel yang berisi data yang disisipkan dipilih. Jika tidak, pilih kolom A.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	10. Pada tab Data, pilih Teks ke Kolom. 11. Di wizard, lakukan hal berikut: • Untuk langkah 1, pilih Delimited. • Untuk langkah 2, untuk Delimiters, pilih Titik koma. Di panel Pratinjau data, konfirmasikan bahwa data sedang dipisahkan menjadi kolom. • Untuk langkah 3, pilih Selesai. 12. Konfirmasikan bahwa data teks dibatasi di beberapa kolom. 13. Simpan laporan Excel dengan nama baru. 14. Cari dan hapus `Access Denied` kesalahan apa pun dalam temuan. Untuk petunjuk tentang cara menghapus ini secara terprogram, lihat Menghapus kesalahan secara terprogram di bagian Informasi tambahan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Selesaikan laporan.	1. Pilih lembar kerja Temuan, lalu pilih sel A17. Sel ini adalah header dari tabel pivot. 2. Di pita, di bawah PivotTable Tools, pilih Analyze, dan kemudian di bawah Refresh, pilih Refresh All. Ini memperbarui tabel pivot dengan kumpulan data baru. 3. Secara default, Excel tidak menampilkan Akun AWS angka dengan benar. Untuk memperbaiki pemformatan angka, lakukan hal berikut: • Pada lembar kerja Temuan, buka menu konteks (klik kanan) untuk kolom A, lalu pilih Format Sel. • Pilih Angka, dan di tempat desimal, masukkan. `0` • Pilih OK. Catatan: Jika Akun AWS angka dimulai dengan satu atau lebih nol, Excel secara otomatis menghapus nol. Jika Anda melihat nomor akun yang kurang dari 12 digit dalam laporan, digit	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang hilang adalah nol di awal angka. 4. (Opsional) Anda dapat menciutkan bidang untuk membuat temuan lebih mudah dibaca. Lakukan hal-hal berikut: • Pada lembar kerja Temuan, jika Anda memindahkan kursor ke garis antara baris 18 dan 19 (spasi antara header kritis dan temuan pertama), ikon kursor berubah menjadi panah kecil yang mengarah ke bawah. • Klik untuk memilih semua bidang pencarian. • Buka menu konteks (klik kanan), temukan Perluas/Ciutkan, lalu pilih Ciutkan. 5. Untuk detail tentang penilaian, tinjau lembar kerja Temuan, Keparahan, dan Lulus Gagal.	

(Opsional) Perbarui Prowler atau sumber daya di repositori kode

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Perbarui Prowler.	Jika Anda ingin memperbarui Prowler ke versi terbaru, lakukan hal berikut:
	1. Connect ke EC2 instance untuk Prowler dengan menggunakan Session Manager. Untuk petunjuk, lihat Connect ke instance Linux Anda menggunakan Session Manager.
	2. Masukkan perintah berikut.
	`sudo -i`
	`pip3 install --upgrade prowler`
		AWS Umum

Perbarui skrip prowler_scan.sh.	Jika Anda ingin memperbarui skrip prowler_scan.sh ke versi terbaru di repo, lakukan hal berikut: 1. Connect ke EC2 instance untuk Prowler dengan menggunakan Session Manager. Untuk petunjuk, lihat Connect ke instance Linux Anda menggunakan Session Manager. 2. Masukkan perintah berikut. `sudo -i`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	3. Arahkan ke direktori skrip Prowler.
	`cd /usr/local/prowler`
	4. Masukkan perintah berikut untuk menyimpan skrip lokal sehingga Anda dapat menggabungkan perubahan kustom ke versi terbaru.
	`git stash`
	5. Masukkan perintah berikut untuk mendapatkan versi terbaru dari skrip.
	`git pull`
	6. Masukkan perintah berikut untuk menggabungkan skrip kustom dengan versi terbaru skrip.
	`git stash pop`
	Note
	Anda mungkin menerima peringatan terkait dengan file yang dibuat secara lokal yang tidak ada dalam GitHub repo, seperti menemukan	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	laporan. Anda dapat mengabaikan ini selama prowler_scan.sh menunjukkan bahwa perubahan yang disimpan secara lokal digabungkan kembali.	

(Opsional) Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus semua sumber daya yang digunakan.	Anda dapat meninggalkan sumber daya yang digunakan di akun. Jika Anda mematikan EC2 instance saat tidak digunakan dan membiarkan bucket S3 kosong, ini mengurangi biaya pemeliharaan sumber daya untuk pemindaian masa depan. Jika Anda ingin menghentikan semua sumber daya, lakukan hal berikut: 1. Hapus `IAM-ProwlerExecRole` tumpukan yang disediakan di akun manajemen. Untuk petunjuk, lihat Menghapus tumpukan	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dalam CloudFormation dokumentasi. 2. Hapus kumpulan `IAM-ProwlerExecRole` tumpukan yang disediakan di akun manajemen organisasi atau di akun administrator yang didelegasikan. Untuk petunjuk, lihat Menghapus kumpulan tumpukan dalam CloudFormation dokumentasi. 3. Hapus semua objek di bucket `prowler-output` S3. Untuk petunjuknya, lihat Menghapus objek dalam dokumentasi Amazon S3. 4. Hapus `Prowler-Resources` tumpukan yang disediakan di akun keamanan. Untuk petunjuk, lihat Menghapus tumpukan dalam CloudFormation dokumentasi.	

Pemecahan masalah

Isu	Solusi
Tidak dapat terhubung ke EC2 instance dengan menggunakan Session Manager.	Agen SSM harus dapat berkomunikasi dengan endpoint Systems Manager. Lakukan hal-hal berikut:

Isu	Solusi
	1. Validasi subnet tempat EC2 instance digunakan memiliki akses internet. 2. Reboot EC2 instance.
Saat menerapkan set tumpukan, CloudFormation konsol meminta Anda untuk melakukannya. `Enable trusted access with AWS Organizations to use service-managed permissions`	Ini menunjukkan bahwa akses tepercaya belum diaktifkan antara AWS Organizations dan CloudFormation. Akses tepercaya diperlukan untuk menerapkan set tumpukan yang dikelola layanan. Pilih tombol untuk mengaktifkan akses tepercaya. Untuk informasi selengkapnya, lihat Mengaktifkan akses tepercaya dalam CloudFormation dokumentasi.

Sumber daya terkait

AWS dokumentasi

- Menerapkan kontrol keamanan pada AWS (Panduan AWS Preskriptif)

Sumber daya lainnya

- Prowler () GitHub

Informasi tambahan

Menghapus kesalahan secara terprogram

Jika hasilnya mengandung `Access Denied` kesalahan, Anda harus menghapusnya dari temuan. Kesalahan ini biasanya disebabkan oleh izin pengaruh eksternal yang mencegah Prowler menilai sumber daya tertentu. Misalnya, beberapa pemeriksaan gagal saat meninjau bucket S3 yang disediakan. AWS Control Tower Anda dapat mengekstrak hasil ini secara terprogram dan menyimpan hasil yang difilter sebagai file baru.

Perintah berikut menghapus baris yang berisi string teks tunggal (pola) dan kemudian output hasilnya ke file baru.

- Untuk Linux atau macOS (Grep)

```
grep -v -i "Access Denied getting bucket" myoutput.csv > myoutput_modified.csv
```

- Untuk Windows (PowerShell)

```
Select-String -Path myoutput.csv -Pattern 'Access Denied getting bucket' -NotMatch >
 myoutput_modified.csv
```

Perintah berikut menghapus baris yang cocok dengan lebih dari satu string teks dan kemudian menampilkan hasilnya ke file baru.

- Untuk Linux atau macOS (Menggunakan pipa yang keluar di antara string)

```
grep -v -i 'Access Denied getting bucket\|Access Denied Trying to Get' myoutput.csv >
 myoutput_modified.csv
```

- Untuk Windows (Menggunakan koma di antara string)

```
Select-String -Path myoutput.csv -Pattern 'Access Denied getting bucket', 'Access
 Denied Trying to Get' -NotMatch > myoutput_modified.csv
```

Laporkan contoh

Gambar berikut adalah contoh lembar kerja Temuan dalam laporan temuan Prowler terkonsolidasi.

Gambar berikut adalah contoh lembar kerja Pass Fail dalam laporan temuan Prowler terkonsolidasi. (Secara default, hasil lulus dikecualikan dari output.)

Gambar berikut adalah contoh lembar kerja Keparahan dalam laporan temuan Prowler terkonsolidasi.

Menyebarkan dan mengelola AWS Control Tower kontrol dengan menggunakan AWS CDK dan CloudFormation

Iker Reina Fuente dan Ivan Girardi, Amazon Web Services

Ringkasan

Pola ini menjelaskan bagaimana menggunakan AWS CloudFormation dan AWS Cloud Development Kit (AWS CDK) menerapkan dan mengelola AWS Control Tower kontrol preventif, detektif, dan proaktif sebagai infrastruktur sebagai kode (IAc). Kontrol (juga dikenal sebagai pagar pembatas) adalah aturan tingkat tinggi yang menyediakan tata kelola berkelanjutan untuk lingkungan Anda secara keseluruhan. AWS Control Tower Misalnya, Anda dapat menggunakan kontrol untuk meminta pencatatan untuk Anda Akun AWS dan kemudian mengonfigurasi pemberitahuan otomatis jika peristiwa terkait keamanan tertentu terjadi.

AWS Control Tower membantu Anda menerapkan kontrol preventif, detektif, dan proaktif yang mengatur AWS sumber daya Anda dan memantau kepatuhan di beberapa. Akun AWS Setiap kontrol memberlakukan satu aturan. Dalam pola ini, Anda menggunakan template IAc yang disediakan untuk menentukan kontrol mana yang ingin Anda terapkan di lingkungan Anda.

AWS Control Tower kontrol berlaku untuk seluruh unit organisasi (OU), dan kontrol mempengaruhi setiap Akun AWS dalam OU. Oleh karena itu, ketika pengguna melakukan tindakan apa pun di akun mana pun di landing zone Anda, tindakan tersebut tunduk pada kontrol yang mengatur OU.

Menerapkan AWS Control Tower kontrol membantu membangun fondasi keamanan yang kuat untuk AWS landing zone Anda. Dengan menggunakan pola ini untuk menerapkan kontrol sebagai IAc CloudFormation dan AWS CDK, Anda dapat menstandarisasi kontrol di landing zone Anda dan menerapkan dan mengelolanya dengan lebih efisien. Solusi ini menggunakan cdk_nag untuk memindai aplikasi selama penerapan. AWS CDK Alat ini memeriksa aplikasi untuk kepatuhan terhadap praktik AWS terbaik.

Untuk menerapkan AWS Control Tower kontrol sebagai IAc, Anda juga dapat menggunakan HashiCorp Terraform sebagai gantinya. AWS CDK Untuk informasi selengkapnya, lihat Menerapkan dan mengelola AWS Control Tower kontrol menggunakan Terraform.

Audiens yang dituju

Pola ini direkomendasikan untuk pengguna yang memiliki pengalaman dengan AWS Control Tower, CloudFormation, AWS CDK, dan AWS Organizations.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dikelola sebagai organisasi di AWS Organizations dan AWS Control Tower landing zone. Untuk petunjuk, lihat Memulai dalam AWS Control Tower dokumentasi.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi.
- Manajer paket node (npm), diinstal dan dikonfigurasi untuk file. AWS CDK
- Prasyarat untuk. AWS CDK
- Izin untuk mengambil peran AWS Identity and Access Management (IAM) yang ada dalam akun penerapan.
- Izin untuk mengambil peran IAM dalam akun manajemen organisasi yang dapat digunakan untuk bootstrap. AWS CDK Peran harus memiliki izin untuk memodifikasi dan menyebarkan CloudFormation sumber daya. Untuk informasi selengkapnya, lihat Bootstrapping di dokumentasi. AWS CDK
- Izin untuk membuat peran dan kebijakan IAM di akun manajemen organisasi. Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk mengakses sumber daya IAM dalam dokumentasi IAM.

Batasan

- Pola ini memberikan instruksi untuk menerapkan solusi ini di seluruh Akun AWS, dari akun penerapan ke akun manajemen organisasi. Untuk tujuan pengujian, Anda dapat menerapkan solusi ini secara langsung di akun manajemen, tetapi instruksi untuk konfigurasi ini tidak disediakan secara eksplisit.
- Untuk AWS Control Tower kontrol, pola ini memerlukan penggunaan pengidentifikasi global yang dalam format berikut:

```
arn:<PARTITION>:controlcatalog:::control/<CONTROL_CATALOG_OPAQUE_ID>
```

  Versi sebelumnya dari pola ini menggunakan pengidentifikasi regional yang tidak lagi didukung. Kami menyarankan Anda bermigrasi dari pengenal Regional ke pengidentifikasi global. Pengidentifikasi global membantu Anda mengelola kontrol dan memperluas jumlah kontrol yang dapat Anda gunakan.

> **Note**
>
> Dalam kebanyakan kasus, nilainya `<PARTITION>` adalah`aws`.

Versi produk

- AWS Control Tower versi 3.2 atau yang lebih baru
- Python versi 3.9 atau yang lebih baru
- npm versi 8.9.0 atau yang lebih baru

Arsitektur

Bagian ini memberikan gambaran tingkat tinggi dari solusi ini dan arsitektur yang ditetapkan oleh kode sampel. Diagram berikut menunjukkan kontrol yang digunakan di berbagai akun di OU.

AWS Control Tower kontrol dikategorikan menurut perilaku dan bimbingannya.

Ada tiga jenis utama perilaku kontrol:

1. Kontrol preventif dirancang untuk mencegah tindakan terjadi. Ini diimplementasikan dengan kebijakan kontrol layanan (SCPs) atau kebijakan kontrol sumber daya (RCPs) di AWS Organizations. Status kontrol preventif ditegakkan atau tidak diaktifkan. Kontrol preventif didukung di semua Wilayah AWS.
2. Kontrol Detektif dirancang untuk mendeteksi peristiwa tertentu ketika terjadi dan mencatat tindakan. AWS CloudTrail Ini diimplementasikan dengan AWS Config aturan. Status kontrol detektif jelas, melanggar, atau tidak diaktifkan. Kontrol Detektif hanya berlaku pada kontrol yang Wilayah AWS didukung oleh. AWS Control Tower
3. Kontrol proaktif memindai sumber daya yang akan disediakan oleh AWS CloudFormation dan memeriksa apakah mereka sesuai dengan kebijakan dan tujuan perusahaan Anda. Sumber daya yang tidak sesuai tidak akan disediakan. Ini diimplementasikan dengan AWS CloudFormation kait. Status kontrol proaktif adalah LULUS, GAGAL, atau SKIP.

Panduan kontrol mengacu pada praktik yang direkomendasikan untuk bagaimana menerapkan setiap kontrol ke Anda OUs. AWS Control Tower menyediakan tiga kategori panduan: wajib, sangat

direkomendasikan, dan elektif. Bimbingan kontrol tidak tergantung pada perilakunya. Untuk informasi selengkapnya, lihat Mengontrol perilaku dan panduan.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode. AWS CDK Toolkit adalah alat utama untuk berinteraksi dengan aplikasi Anda AWS CDK .
- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu.
- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.

Alat-alat lainnya

- cdk_nag adalah alat sumber terbuka yang menggunakan kombinasi paket aturan untuk memeriksa kepatuhan AWS CDK aplikasi terhadap praktik terbaik.
- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.
- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di AWS Control Tower kontrol GitHub Deploy menggunakan AWS CDK repositori. Anda menggunakan file cdk.json untuk berinteraksi dengan AWS CDK aplikasi, dan Anda menggunakan file package.json untuk menginstal paket npm.

Praktik terbaik

- Ikuti prinsip hak istimewa paling sedikit (dokumentasi IAM). Contoh kebijakan IAM dan kebijakan kepercayaan yang disediakan dalam pola ini mencakup izin minimum yang diperlukan, dan AWS CDK tumpukan yang dibuat di akun manajemen dibatasi oleh izin ini.
- Ikuti Praktik terbaik untuk AWS Control Tower administrator (AWS Control Tower dokumentasi).
- Ikuti Praktik terbaik untuk mengembangkan dan menerapkan infrastruktur cloud dengan AWS CDK (AWS CDK dokumentasi).
- Saat mem-bootstrap AWS CDK, sesuaikan template bootstrap untuk menentukan kebijakan dan akun tepercaya yang harus memiliki kemampuan untuk membaca dan menulis ke sumber daya apa pun di akun manajemen. Untuk informasi selengkapnya, lihat Menyesuaikan bootstrap.
- Gunakan alat analisis kode, seperti cfn_nag, untuk memindai templat yang dihasilkan. CloudFormation Alat cfn-nag mencari pola dalam CloudFormation template yang mungkin menunjukkan infrastruktur tidak aman. Anda juga dapat menggunakan cdk-nag untuk memeriksa CloudFormation template Anda dengan menggunakan modul cloudformation-include.

Epik

Bersiaplah untuk mengaktifkan kontrol

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran IAM di akun manajemen.	1. Buat kebijakan IAM di akun manajemen dengan izin yang ditentukan dalam kebijakan IAM di bagian Informasi tambahan. Untuk petunjuk, lihat Membuat kebijakan IAM dalam dokumentasi IAM. Catat Nama Sumber Daya Amazon (ARN) dari kebijakan tersebut. Berikut ini adalah contoh ARN.	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`arn:aws:iam::<MANAGEMENT-ACCOUNT-ID>:policy/<POLICY-NAME>`
	2. Buat peran IAM di akun manajemen, lampirkan kebijakan izin IAM yang Anda buat pada langkah sebelumnya, dan lampirkan kebijakan kepercayaan khusus dalam kebijakan Trust di bagian Informasi tambahan. Untuk petunjukn ya, lihat Membuat peran menggunakan kebijakan kepercayaan khusus dalam dokumentasi IAM. Berikut ini adalah contoh ARN untuk peran baru.
	`arn:aws:iam::<MANAGEMENT-ACCOUNT-ID>:role/<ROLE-NAME>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap AWS CDK.	1. Di akun manajemen, asumsikan peran yang memiliki izin untuk bootstrap AWS CDK. 2. Masukkan perintah berikut, ganti yang berikut ini: • `<MANAGEMENT-ACCOUNT-ID>` adalah ID akun manajemen organisasi. • `<AWS-CONTROL-TOWER-REGION>` adalah Wilayah AWS tempat AWS Control Tower dikerahkan. Untuk daftar lengkap kode Wilayah, lihat Titik akhir Regional di Referensi AWS Umum. • `<DEPLOYMENT-ACCOUNT-ID>` adalah ID dari akun penerapan. • `<DEPLOYMENT-ROLE-NAME>` adalah nama peran IAM yang Anda gunakan akun penerapan . • `<POLICY-NAME>` adalah nama kebijakan yang Anda buat di akun manajemen. `$ npx cdk bootstrap aws://<MANAGEMENT-ACCOUNT-ID>/<AWS-C`	DevOps insinyur, AWS Umum, Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>ONTROL-TOWER-REGIO
	N> \
	--trust arn:aws:i
	am::<DEPLOYMENT-AC
	COUNT-ID>:role/<DE
	PLOYMENT-ROLE-NAME>
	 \
	--cloudformation-
	execution-policies
	 arn:aws:iam::<MANA
	GEMENT-ACCOUNT-ID>
	:policy/<POLICY-NA
	ME></pre>	

Kloning repositori.	Dalam shell bash, masukkan perintah berikut. Ini mengkloning AWS Control Tower kontrol Deploy menggunakan AWS CDK repositori dari. GitHub
	<pre>git clone https://g
	ithub.com/aws-samp
	les/aws-control-to
	wer-controls-cdk.git</pre>
		DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Edit file AWS CDK konfigurasi.	1. Di repositori kloning, buka file constants.py. 2. Dalam `ACCOUNT_ID` parameter, masukkan ID akun manajemen Anda. 3. Dalam `<AWS-CONTROL-TOWER-REGION>` parameter, masukkan Wilayah AWS tempat AWS Control Tower dikerahkan. 4. Dalam `ROLE_ARN` parameter, masukkan ARN dari peran yang Anda buat di akun manajemen. 5. Buka Semua pengidentifikasi global dalam AWS Control Tower dokumentasi. 6. Dalam daftar berformat JSON, cari kontrol yang ingin Anda terapkan, lalu salin pengenal globalnya (juga dikenal sebagai nilai {CONTROL_CATALOG_OPAQUE_ID}). Misalnya, pengidentifikasi global untuk kontrol AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED adalah. `k4izcjxhukijhajp6ks5mjxk` 7. Di `GUARDRAILS_CONFIGURATION` bagian, dalam `Enable-`	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Control` parameter, masukkan pengidentifikasi global yang Anda salin. Masukkan pengenal dalam tanda kutip ganda, dan pisahkan beberapa pengidentifikasi dengan koma. 8. Di `GUARDRAILS_CONFIGURATION` bagian, dalam `OrganizationalUnitIds` parameter, masukkan ID unit organisasi tempat Anda ingin mengaktifkan kontrol, seperti `ou-1111-11111111`. Masukkan ID dalam tanda kutip ganda, dan pisahkan beberapa IDs dengan koma. Untuk informasi selengkapnya tentang cara mengambil OU IDs, lihat Melihat detail OU. 9. Simpan dan tutup file constants.py. Untuk contoh file constants.py yang diperbarui, lihat bagian Informasi tambahan dari pola ini.	

Aktifkan kontrol di akun manajemen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Asumsikan peran IAM dalam akun penerapan.	Di akun penerapan, asumsikan peran IAM yang memiliki izin untuk menyebarkan AWS CDK tumpukan di akun manajemen. Untuk informasi selengkapnya tentang mengasumsikan peran IAM dalam AWS CLI, lihat Menggunakan peran IAM dalam. AWS CLI	DevOps insinyur, AWS Umum

Aktifkan lingkungan.	Jika Anda menggunakan Linux atau macOS:
	1. Masukkan perintah berikut untuk membuat lingkungan virtual.
	`$ python3 -m venv .venv`
	2. Setelah lingkungan virtual dibuat, masukkan perintah berikut untuk mengaktifkannya.
	`$ source .venv/bin/activate`
	Jika Anda menggunakan Windows:
		DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	1. Masukkan perintah berikut untuk mengaktifkan lingkungan virtual. `% .venv\Scripts\activate.bat`	
Instal dependensi.	Setelah lingkungan virtual diaktifkan, masukkan perintah berikut untuk menjalankan skrip install_deps.sh. Skrip ini menginstal dependensi yang diperlukan. `$ ./scripts/install_deps.sh`	DevOps insinyur, AWS Umum, Python

Menyebarkan tumpukan.	Masukkan perintah berikut untuk mensintesis dan menyebarkan tumpukan. CloudFormation
	`$ npx cdk synth`
	`$ npx cdk deploy`
		DevOps insinyur, AWS Umum, Python

Sumber daya terkait

AWS dokumentasi

- Tentang kontrol (AWS Control Tower dokumentasi)
- Pustaka kontrol (AWS Control Tower dokumentasi)
- AWS CDK Perintah Toolkit (AWS CDK dokumentasi)
- Terapkan dan kelola AWS Control Tower kontrol dengan menggunakan Terraform (Panduan Preskriptif)AWS

Sumber daya lainnya

- Python

Informasi tambahan

Contoh file constants.py

Berikut ini adalah contoh dari file constants.py diperbarui. Sampel ini mengaktifkan kontrol AWS-GR_DISALLOW_CROSS_REGION_NETWORKING (global ID:) dan kontrol AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED (ID global:). **dvuaav61i5cnfazfelmvn9m6k** 50z1ot237wl8u1lv5ufau6qqo Untuk daftar global IDs, lihat Semua pengidentifikasi global dalam AWS Control Tower dokumentasi.

```
ACCOUNT_ID = 111122223333
AWS_CONTROL_TOWER_REGION = us-east-2
ROLE_ARN = "arn:aws:iam::111122223333:role/CT-Controls-Role"
GUARDRAILS_CONFIGURATION = [
    {
        "Enable-Control": {
            "dvuaav61i5cnfazfelmvn9m6k": {  # AWS-GR_DISALLOW_CROSS_REGION_NETWORKING
                "Parameters": {
                    "ExemptedPrincipalArns": ["arn:aws:iam::111122223333:role/
RoleName"]
                },
                "Tags": [{"key": "Environment", "value": "Production"}]
            },
            ...
        },
        "OrganizationalUnitIds": ["ou-1111-11111111", "ou-2222-22222222"...],
    },
    {
        "Enable-Control": {
            "50z1ot237wl8u1lv5ufau6qqo",  # AWS-
GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED
            ...
        },
        "OrganizationalUnitIds": ["ou-2222-22222222"...],
    },
]
```

Kebijakan IAM

Kebijakan contoh berikut memungkinkan tindakan minimum yang diperlukan untuk mengaktifkan atau menonaktifkan AWS Control Tower kontrol saat menerapkan AWS CDK tumpukan dari akun penerapan ke akun manajemen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "controltower:EnableControl",
                "controltower:DisableControl",
                "controltower:GetControlOperation",
                "controltower:ListEnabledControls",
                "organizations:AttachPolicy",
                "organizations:CreatePolicy",
                "organizations:DeletePolicy",
                "organizations:DescribeOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DetachPolicy",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListChildren",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListParents",
                "organizations:ListPoliciesForTarget",
                "organizations:ListRoots",
                "organizations:UpdatePolicy",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan kepercayaan

Kebijakan kepercayaan khusus berikut memungkinkan peran IAM tertentu dalam akun penerapan untuk mengambil peran IAM dalam akun manajemen. Ganti yang berikut ini:

- `<DEPLOYMENT-ACCOUNT-ID>`adalah ID dari akun penerapan

- <DEPLOYMENT-ROLE-NAME>adalah nama peran dalam akun penyebaran yang diizinkan untuk mengambil peran dalam akun manajemen

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<DEPLOYMENT-ACCOUNT-ID>:role/<DEPLOYMENT-ROLE-
NAME>"
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

Terapkan dan kelola AWS Control Tower kontrol dengan menggunakan Terraform

Iker Reina Fuente dan Ivan Girardi, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan AWS Control Tower kontrol, HashiCorp Terraform, dan infrastruktur sebagai kode (IAc) untuk mengimplementasikan dan mengelola kontrol keamanan preventif, detektif, dan proaktif. Kontrol (juga dikenal sebagai pagar pembatas) adalah aturan tingkat tinggi yang menyediakan tata kelola berkelanjutan untuk lingkungan Anda secara keseluruhan. AWS Control Tower Misalnya, Anda dapat menggunakan kontrol untuk meminta pencatatan untuk Anda Akun AWS dan kemudian mengonfigurasi pemberitahuan otomatis jika peristiwa terkait keamanan tertentu terjadi.

AWS Control Tower membantu Anda menerapkan kontrol preventif, detektif, dan proaktif yang mengatur AWS sumber daya Anda dan memantau kepatuhan di beberapa. Akun AWS Setiap kontrol memberlakukan satu aturan. Dalam pola ini, Anda menggunakan template IAc yang disediakan untuk menentukan kontrol mana yang ingin Anda terapkan di lingkungan Anda.

AWS Control Tower kontrol berlaku untuk seluruh unit organisasi (OU), dan kontrol mempengaruhi setiap Akun AWS dalam OU. Oleh karena itu, ketika pengguna melakukan tindakan apa pun di akun mana pun di landing zone Anda, tindakan tersebut tunduk pada kontrol yang mengatur OU.

Menerapkan AWS Control Tower kontrol membantu membangun fondasi keamanan yang kuat untuk AWS landing zone Anda. Dengan menggunakan pola ini untuk menerapkan kontrol sebagai IAc melalui Terraform, Anda dapat menstandarisasi kontrol di landing zone Anda dan menerapkan dan mengelolanya dengan lebih efisien.

Untuk menerapkan AWS Control Tower kontrol sebagai IAc, Anda juga dapat menggunakan AWS Cloud Development Kit (AWS CDK) alih-alih Terraform. Untuk informasi selengkapnya, lihat Menerapkan dan mengelola AWS Control Tower kontrol dengan menggunakan AWS CDK dan AWS CloudFormation.

Audiens yang dituju

Pola ini direkomendasikan untuk pengguna yang memiliki pengalaman dengan AWS Control Tower, Terraform, dan. AWS Organizations

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dikelola sebagai organisasi di AWS Organizations dan AWS Control Tower landing zone. Untuk petunjuk, lihat Memulai dalam AWS Control Tower dokumentasi.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi.
- Peran AWS Identity and Access Management (IAM) dalam akun manajemen yang memiliki izin untuk menerapkan pola ini. Untuk informasi selengkapnya tentang izin yang diperlukan dan kebijakan sampel, lihat Izin hak istimewa terkecil untuk peran IAM di bagian Informasi tambahan pada pola ini.
- Izin untuk mengambil peran IAM dalam akun manajemen.
- Terraform CLI, diinstal (dokumentasi Terraform).
- AWS Penyedia Terraform, dikonfigurasi (dokumentasi Terraform).
- Backend Terraform, dikonfigurasi (dokumentasi Terraform).

Batasan

- Untuk AWS Control Tower kontrol, pola ini memerlukan penggunaan pengidentifikasi global yang dalam format berikut:

```
arn:<PARTITION>:controlcatalog:::control/<CONTROL_CATALOG_OPAQUE_ID>
```

  Versi sebelumnya dari pola ini menggunakan pengidentifikasi regional yang tidak lagi didukung. Kami menyarankan Anda bermigrasi dari pengenal Regional ke pengidentifikasi global. Pengidentifikasi global membantu Anda mengelola kontrol dan memperluas jumlah kontrol yang dapat Anda gunakan.

> **Note**
>
> Dalam kebanyakan kasus, nilainya `<PARTITION>` adalah`aws`.

Versi produk

- AWS Control Tower versi 3.2 atau yang lebih baru

- Terraform versi 1.5 atau yang lebih baru
- AWS Penyedia Terraform versi 4.67 atau yang lebih baru

Arsitektur

Bagian ini memberikan gambaran tingkat tinggi dari solusi ini dan arsitektur yang ditetapkan oleh kode sampel. Diagram berikut menunjukkan kontrol yang digunakan di berbagai akun di OU.

AWS Control Tower kontrol dikategorikan menurut perilaku dan bimbingannya.

Ada tiga jenis utama perilaku kontrol:

1. Kontrol preventif dirancang untuk mencegah tindakan terjadi. Ini diimplementasikan dengan kebijakan kontrol layanan (SCPs) atau kebijakan kontrol sumber daya (RCPs) di AWS Organizations. Status kontrol preventif ditegakkan atau tidak diaktifkan. Kontrol preventif didukung di semua Wilayah AWS.
2. Kontrol Detektif dirancang untuk mendeteksi peristiwa tertentu ketika terjadi dan mencatat tindakan. AWS CloudTrail Ini diimplementasikan dengan AWS Config aturan. Status kontrol detektif jelas, melanggar, atau tidak diaktifkan. Kontrol Detektif hanya berlaku pada kontrol yang Wilayah AWS didukung oleh. AWS Control Tower
3. Kontrol proaktif memindai sumber daya yang akan disediakan oleh AWS CloudFormation dan memeriksa apakah mereka sesuai dengan kebijakan dan tujuan perusahaan Anda. Sumber daya yang tidak sesuai tidak akan disediakan. Ini diimplementasikan dengan AWS CloudFormation kait. Status kontrol proaktif adalah LULUS, GAGAL, atau SKIP.

Panduan kontrol adalah praktik yang direkomendasikan untuk bagaimana menerapkan setiap kontrol ke Anda OUs. AWS Control Tower menyediakan tiga kategori panduan: wajib, sangat direkomendasikan, dan elektif. Bimbingan kontrol tidak tergantung pada perilakunya. Untuk informasi selengkapnya, lihat Mengontrol perilaku dan panduan.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.

- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu.
- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.

Alat-alat lainnya

- HashiCorp Terraform adalah alat infrastruktur sebagai kode (IAc) yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.

Repositori kode

Kode untuk pola ini tersedia di GitHub Deploy dan kelola AWS Control Tower kontrol dengan menggunakan repositori Terraform.

Praktik terbaik

- Peran IAM yang digunakan untuk menerapkan solusi ini harus mematuhi prinsip hak istimewa paling sedikit (dokumentasi IAM).
- Ikuti Praktik terbaik untuk AWS Control Tower administrator (AWS Control Tower dokumentasi).

Epik

Aktifkan kontrol di akun manajemen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Dalam shell bash, masukkan perintah berikut. Ini mengkloni ng AWS Control Tower kontrol Deploy dan mengelola dengan menggunakan repositori Terraform dari. GitHub	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/aws-samples/aws-control-tower-controls-terraform.git`	
Edit file konfigurasi backend Terraform.	1. Di repositori kloning, buka file backend.tf. 2. Edit file untuk mengatur konfigurasi backend Terraform. Konfigurasi yang Anda tentukan dalam file ini tergantung pada lingkungan Anda. Untuk informasi selengkapnya, lihat Konfigurasi backend (dokumentasi Terraform). 3. Simpan dan tutup file backend.tf.	DevOps insinyur, Terraform
Edit file konfigurasi penyedia Terraform.	1. Di repositori kloning, buka file provider.tf. 2. Edit file untuk mengatur konfigurasi penyedia Terraform. Untuk informasi selengkapnya, lihat Konfigurasi penyedia (Dokumentasi Terraform). Tetapkan Wilayah AWS sebagai Wilayah tempat AWS Control Tower API tersedia. 3. Simpan dan tutup file provider.tf.	DevOps insinyur, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Edit file konfigurasi.	1. Di repositori kloning, buka file.tfvars variabel.tfvars. 2. Buka Semua pengidentifikasi global dalam AWS Control Tower dokumentasi. 3. Dalam daftar berformat JSON, cari kontrol yang ingin Anda terapkan, lalu salin pengenal globalnya (juga dikenal sebagai nilai {CONTROL_CATALOG_OPAQUE_ID}). Misalnya, pengidentifikasi global untuk kontrol AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED adalah. `k4izcjxhukijhajp6ks5mjxk` 4. Di `controls` bagian, dalam `control_names` parameter, masukkan pengidentifikasi global yang Anda salin. 5. Di `controls` bagian, dalam `organizational_unit_ids` parameter, masukkan ID unit organisasi tempat Anda ingin mengaktifkan kontrol, sepertiou-1111-1111111 . Masukkan ID dalam tanda kutip ganda, dan pisahkan beberapa IDs dengan koma. Untuk	DevOps insinyur, AWS Umum, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi selengkapnya tentang cara mengambil OU IDs, lihat Melihat detail OU. 6. Simpan dan tutup file variabel.tfvars. Untuk contoh file variabel.tfvars yang diperbarui, lihat bagian Informasi tambahan dari pola ini.	
Asumsikan peran IAM dalam akun manajemen.	Di akun manajemen, asumsikan peran IAM yang memiliki izin untuk menyebarkan file konfigurasi Terraform. Untuk informasi selengkapnya tentang izin yang diperlukan dan kebijakan sampel, lihat Izin hak istimewa terkecil untuk peran IAM di bagian Informasi tambahan. Untuk informasi selengkapnya tentang mengasumsikan peran IAM dalam AWS CLI, lihat Menggunakan peran IAM dalam. AWS CLI	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menyebarkan file konfigurasi.	1. Masukkan perintah berikut untuk menginisialisasi Terraform.
	`$ terraform init -upgrade`
	2. Masukkan perintah berikut untuk melihat pratinjau perubahan dibandingkan keadaan saat ini.
	`$ terraform plan -var-file="variables.tfvars"`
	3. Tinjau perubahan konfigurasi dalam paket Terraform dan konfirmasikan bahwa Anda ingin menerapkan perubahan ini di organisasi.
	4. Masukkan perintah berikut untuk menyebarkan sumber daya.
	`$ terraform apply -var-file="variables.tfvars"`
		DevOps insinyur, AWS Umum, Terraform

(Opsional) Nonaktifkan kontrol di akun AWS Control Tower manajemen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan perintah `destroy`.	Masukkan perintah berikut untuk menghapus sumber	DevOps insinyur, AWS Umum, Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	daya yang digunakan oleh pola ini. `$ terraform destroy -var-file="variables.tfvars"`	

Pemecahan masalah

Isu	Solusi
`Error: creating ControlTower Control ValidationException: Guardrail <control ID> is already enabled on organizational unit <OU ID>` kesalahan	Kontrol yang Anda coba aktifkan sudah diaktifkan di OU target. Kesalahan ini dapat terjadi jika pengguna secara manual mengaktifkan kontrol melalui Konsol Manajemen AWS, melalui AWS Control Tower atau melalui AWS Organizations. Untuk menerapkan file konfigurasi Terraform, Anda dapat menggunakan salah satu dari opsi berikut. Opsi 1: Perbarui file status Terraform saat ini Anda dapat mengimpor sumber daya ke file status Terraform saat ini. Saat Anda menjalankan kembali `apply` perintah, Terraform akan melewati sumber daya ini. Lakukan hal berikut untuk mengimpor sumber daya ke status Terraform saat ini: 1. Di akun AWS Control Tower manajemen, masukkan perintah berikut untuk mengambil daftar Nama Sumber Daya Amazon (ARNs) untuk OUs, di `<root-ID>` mana root organisasi. Untuk informasi selengkapnya

Isu	Solusi

	tentang mengambil ID ini, lihat Melihat detail root.
	
	`aws organizations list-organizational-units-for-parent --parent-id <root-ID>`
	
	2. Untuk setiap OU yang dikembalikan pada langkah sebelumnya, masukkan perintah berikut, di `<OU-ARN>` mana ARN dari OU.
	
	`aws controltower list-enabled-controls --target-identifier <OU-ARN>`
	
	3. Salin ARNs dan lakukan impor Terraform dalam modul yang diperlukan sehingga disertakan dalam status Terraform. Untuk petunjuk, lihat Impor (dokumentasi Terraform).
	4. Ulangi langkah-langkah dalam Menyebarkan konfigurasi di bagian Epik.
	
	Opsi 2: Nonaktifkan kontrol
	
	Jika Anda bekerja di lingkungan non-produksi, Anda dapat menonaktifkan kontrol di konsol. Aktifkan kembali dengan mengulangi langkah-langkah di Deploy konfigurasi di bagian Epics. Pendekatan ini tidak direkomendasikan untuk lingkungan produksi karena ada periode waktu ketika kontrol akan dinonaktifkan. Jika Anda ingin menggunakan opsi ini di lingkungan produksi, Anda dapat menerapkan kontrol sementara, seperti menerapkan SCP sementara. AWS Organizations

Sumber daya terkait

AWS dokumentasi

- Tentang kontrol (AWS Control Tower dokumentasi)
- Pustaka kontrol (AWS Control Tower dokumentasi)
- Menyebarkan dan mengelola AWS Control Tower kontrol dengan menggunakan AWS CDK dan AWS CloudFormation (Panduan AWS Preskriptif)

Sumber daya lainnya

- Terraform
- Dokumentasi CLI Terraform

Informasi tambahan

Contoh variabel.file tfvars

Berikut ini adalah contoh dari file variabel.tfvars yang diperbarui. Sampel ini mengaktifkan kontrol AWS-GR_ENCRYPTED_VOLUME (global ID:) dan kontrol AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED (ID global:**`503uicglhjkokaajywfpt6ros`**). `50z1ot237wl8u1lv5ufau6qqo` Untuk daftar global IDs, lihat Semua pengidentifikasi global dalam AWS Control Tower dokumentasi.

Contoh berikut juga memungkinkan kontrol yang memiliki parameter seperti CT.S3.PV.5 (global ID:**`7mo7a2h2ebsq71l8k6uzr96ou`**) dan CT.SECRETSMANAGER.PV.1 (global ID:). `dvhe47fxg5o6lryqrq9g6sxg4` Untuk daftar kontrol dengan parameter, lihat Kontrol dengan parameter dalam AWS Control Tower dokumentasi.

```
controls = [
    {
        control_names = [
            "503uicglhjkokaajywfpt6ros", # AWS-GR_ENCRYPTED_VOLUMES
            ...
        ],
        organizational_unit_ids = ["ou-1111-11111111", "ou-2222-22222222"...],
    },
    {
        control_names = [
```

```
            "50z1ot237wl8u1lv5ufau6qqo", # AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED
            ...
        ],
        organizational_unit_ids = ["ou-1111-11111111"...],
    },
]

controls_with_params = [
  {
    control_names = [
      { "7mo7a2h2ebsq71l8k6uzr96ou" = { # CT.S3.PV.5
        parameters = {
          "ExemptedPrincipalArns" : ["arn:aws:iam::*:role/RoleName"],
          "ExemptedResourceArns" : [],
        }
      } },
      { "dvhe47fxg5o6lryqrq9g6sxg4" = { # CT.SECRETSMANAGER.PV.1
        parameters = {
          "ExemptedPrincipalArns" : ["arn:aws:iam::*:role/RoleName"],
        }
      } },
      ...
    ],
    organizational_unit_ids = ["ou-1111-11111111"...]
  },
  {
    control_names = [
      { "dvuaav61i5cnfazfelmvn9m6k" = { # AWS-GR_DISALLOW_CROSS_REGION_NETWORKING
        parameters = {
          "ExemptedPrincipalArns" : ["arn:aws:iam::*:role/RoleName"],
        }
      } },
      { "41ngl8m5c4eb1myoz0t707n7h" = { # AWS-GR_DISALLOW_VPC_INTERNET_ACCESS
        parameters = {
          "ExemptedPrincipalArns" : ["arn:aws:iam::*:role/RoleName"],
        }
      } },
      ...
    ],
    organizational_unit_ids = ["ou-2222-22222222"...]
  }
]
```

Izin hak istimewa paling sedikit untuk peran IAM

Pola ini mengharuskan Anda mengambil peran IAM dalam akun manajemen. Praktik terbaik adalah mengambil peran dengan izin sementara dan membatasi izin sesuai dengan prinsip hak istimewa paling sedikit. Kebijakan contoh berikut memungkinkan tindakan minimum yang diperlukan untuk mengaktifkan atau menonaktifkan AWS Control Tower kontrol.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "controltower:EnableControl",
                "controltower:DisableControl",
                "controltower:GetControlOperation",
                "controltower:ListEnabledControls",
                "organizations:AttachPolicy",
                "organizations:CreatePolicy",
                "organizations:DeletePolicy",
                "organizations:DescribeOrganization",
                "organizations:DetachPolicy",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListChildren",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListParents",
                "organizations:ListPoliciesForTarget",
                "organizations:ListRoots",
                "organizations:UpdatePolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Terapkan Otomatisasi Keamanan untuk AWS WAF solusi dengan menggunakan Terraform

Dr. Rahul Sharad Gaikwad dan Tamilselvan P, Amazon Web Services

Ringkasan

AWS WAF adalah firewall aplikasi web yang membantu melindungi aplikasi dari eksploitasi umum dengan menggunakan aturan yang dapat disesuaikan, yang Anda tentukan dan terapkan dalam daftar kontrol akses web (). ACLs Mengkonfigurasi AWS WAF aturan dapat menjadi tantangan, terutama bagi organisasi yang tidak memiliki tim keamanan khusus. Untuk menyederhanakan proses ini, Amazon Web Services (AWS) menawarkan Security Automations for AWS WAF solution, yang secara otomatis menyebarkan ACL web tunggal dengan seperangkat AWS WAF aturan yang memfilter serangan berbasis web. Selama penerapan Terraform, Anda dapat menentukan fitur pelindung mana yang akan disertakan. Setelah menerapkan solusi ini, AWS WAF periksa permintaan web ke CloudFront distribusi Amazon atau Application Load Balancer yang ada, dan blokir permintaan apa pun yang tidak sesuai dengan aturan.

Otomatisasi Keamanan untuk AWS WAF solusi dapat digunakan dengan menggunakan AWS CloudFormation sesuai dengan instruksi dalam Panduan Otomasi Keamanan untuk Implementasi. AWS WAF Pola ini memberikan opsi penerapan alternatif untuk organisasi yang menggunakan HashiCorp Terraform sebagai alat infrastruktur pilihan mereka sebagai kode (IAc) untuk menyediakan dan mengelola infrastruktur cloud mereka. Saat Anda menerapkan solusi ini, Terraform secara otomatis menerapkan perubahan di cloud dan menyebarkan serta mengonfigurasi pengaturan dan fitur pelindung. AWS WAF

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- AWS Command Line Interface (AWS CLI) versi 2.4.25 atau yang lebih baru, diinstal dan dikonfigurasi dengan izin yang diperlukan. Untuk informasi selengkapnya, lihat Memulai (AWS CLI dokumentasi).
- Terraform versi 1.1.9 atau yang lebih baru, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal Terraform (dokumentasi Terraform).

Arsitektur

Arsitektur target

Pola ini menyebarkan Otomatisasi Keamanan untuk AWS WAF solusi. Untuk informasi selengkapnya tentang arsitektur target, lihat Ikhtisar arsitektur di Panduan Otomasi Keamanan untuk AWS WAF Implementasi. Untuk informasi selengkapnya tentang AWS Lambda otomatisasi dalam penerapan ini, Parser log aplikasi, parser AWS WAF log, pengurai daftar IP, dan pengendali Access, lihat Detail komponen dalam Panduan Otomasi Keamanan untuk Implementasi. AWS WAF

Penerapan Terraform

Saat Anda menjalankan`terraform apply`, Terraform melakukan hal berikut:

1. Terraform membuat peran AWS Identity and Access Management (IAM) dan fungsi Lambda berdasarkan input dari file.tfvars pengujian.
2. Terraform membuat aturan AWS WAF ACL dan set IP berdasarkan input dari file testing.tfvars.
3. Terraform membuat bucket Amazon Simple Storage Service (Amazon S3), AWS Glue aturan Amazon, tabel database, dan grup kerja EventBridge Amazon Athena berdasarkan input dari file.tfvars pengujian.
4. Terraform menerapkan AWS CloudFormation tumpukan untuk menyediakan sumber daya khusus.
5. Terraform membuat resource Amazon API Gateway berdasarkan input yang diberikan dari file testing.tfvars.

Otomatisasi dan skala

Anda dapat menggunakan pola ini untuk membuat AWS WAF aturan untuk beberapa Akun AWS dan Wilayah AWS untuk menerapkan Otomasi Keamanan untuk AWS WAF solusi di seluruh lingkungan Anda AWS Cloud .

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.
- AWS WAFadalah firewall aplikasi web yang membantu Anda memantau permintaan HTTP dan HTTPS yang diteruskan ke sumber daya aplikasi web Anda yang dilindungi.

Layanan lainnya

- Git adalah sistem kontrol versi terdistribusi open-source.
- HashiCorp Terraform adalah aplikasi antarmuka baris perintah yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub AWS WAF Automation Using Terraform.

Praktik terbaik

- Letakkan file statis di bucket Amazon S3 terpisah.
- Hindari variabel hardcoding.
- Batasi penggunaan skrip kustom.
- Mengadopsi konvensi penamaan.

Epik

Siapkan workstation lokal Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Git.	Ikuti petunjuk di Memulai (situs web Git) untuk menginstal Git di workstation lokal Anda.	DevOps insinyur
Kloning repositori.	Di workstation lokal Anda, masukkan perintah berikut untuk mengkloning repositori kode: `git clone https://github.com/aws-samples/aws-waf-automation-terraform-samples.git`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui variabel.	1. Arahkan ke direktori kloning dengan memasukkan perintah berikut: `cd terraform-aws-waf-automation` 2. Di editor teks apa pun, buka file.tfvars pengujian.tfvars. 3. Perbarui nilai variabel dalam file testing.tfvars. 4. Simpan dan tutup file .	DevOps insinyur

Menyediakan arsitektur target menggunakan Terraform

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Inisialisasi konfigurasi Terraform.	Masukkan perintah berikut untuk menginisialisasi direktori kerja Anda yang berisi file konfigurasi Terraform: `terraform init`	DevOps insinyur
Pratinjau paket Terraform.	Masukkan perintah berikut. Terraform mengevaluasi file konfigurasi untuk menentuka n status target untuk sumber daya yang dideklarasikan. Kemudian membandin gkan status target dengan keadaan saat ini dan membuat rencana:	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`terraform plan -var-file ="testing.tfvars"`	
Verifikasi rencananya.	Tinjau rencana dan konfirmas ikan bahwa itu mengonfigurasi arsitektur yang diperlukan dalam target Akun AWS Anda.	DevOps insinyur
Menyebarkan solusinya.	1. Masukkan perintah berikut untuk menerapkan rencana: `terraform apply -var-file="testing .tfvars"` 2. Masukkan `yes` untuk mengonfirmasi. Terraform membuat, memperbar ui, atau menghancu rkan infrastruktur untuk mencapai status target yang dideklarasikan dalam file konfigurasi. Untuk informasi lebih lanjut tentang urutan, lihat Penerapan Terraform di bagian Arsitektur dari pola ini.	DevOps insinyur

Validasi dan bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Verifikasi perubahannya.	1. Di konsol Terraform, verifikasi bahwa output	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	cocok dengan hasil yang diharapkan. 2. Masuk ke Konsol Manajemen AWS. 3. Verifikasi bahwa output di konsol Terraform telah berhasil diterapkan di konsol Anda. Akun AWS	
(Opsional) Bersihkan infrastruktur.	Jika Anda ingin menghapus semua sumber daya dan perubahan konfigurasi yang dibuat oleh solusi ini, lakukan hal berikut: 1. Di konsol Terraform, masukkan perintah berikut: `terraform destroy -var-file="testing.tfvars"` 2. Masukkan `yes` untuk mengonfirmasi.	DevOps insinyur

Pemecahan masalah

Isu	Solusi
`WAFV2 IPSet: WAFOptimisticLockException` kesalahan	Jika Anda menerima kesalahan ini saat menjalankan `terraform destroy` perintah, Anda harus menghapus set IP secara manual. Untuk petunjuk, lihat Menghapus set IP (AWS WAF dokumentasi).

Sumber daya terkait

AWS referensi

- Otomatisasi Keamanan untuk Panduan AWS WAF Implementasi
- Otomatisasi Keamanan untuk AWS WAF (Perpustakaan AWS Solusi)
- Otomasi Keamanan untuk FAQ AWS WAF

Referensi Terraform

- Konfigurasi backend Terraform
- AWS Penyedia Terraform - Dokumentasi dan Penggunaan
- AWS Penyedia Terraform (repositori) GitHub

Menerapkan pipeline yang secara bersamaan mendeteksi masalah keamanan di beberapa kiriman kode

Benjamin Morris, Tim Hahn, Sapeksh Madan, Dina Odum, dan Yesaya Schisler, Amazon Web Services

Ringkasan

Simple Code Scanning Pipeline (SCSP) menyediakan pembuatan dua klik dari pipeline analisis kode yang menjalankan alat keamanan open-source standar industri secara paralel. Ini memungkinkan pengembang untuk memeriksa kualitas dan keamanan kode mereka tanpa harus menginstal alat atau bahkan memahami cara menjalankannya. Ini membantu Anda mengurangi kerentanan dan kesalahan konfigurasi dalam kiriman kode. Ini juga mengurangi jumlah waktu yang dihabiskan organisasi Anda untuk menginstal, meneliti, dan mengonfigurasi alat keamanan.

Sebelum SCSP, memindai kode menggunakan rangkaian alat khusus ini mengharuskan pengembang untuk menemukan, menginstal secara manual, dan mengkonfigurasi alat analisis perangkat lunak. Bahkan all-in-one alat yang diinstal secara lokal, seperti Automated Security Helper (ASH), memerlukan konfigurasi kontainer Docker agar dapat dijalankan. Namun, dengan SCSP, serangkaian alat analisis kode standar industri berjalan secara otomatis di. AWS Cloud Dengan solusi ini, Anda menggunakan Git untuk mendorong kiriman kode Anda, dan kemudian Anda menerima output visual dengan at-a-glance wawasan tentang mana pemeriksaan keamanan gagal.

Prasyarat dan batasan

- Aktif Akun AWS
- Satu atau beberapa kiriman kode yang ingin Anda pindai untuk masalah keamanan
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Python versi 3.0 atau yang lebih baru dan pip versi 9.0.3 atau yang lebih baru, diinstal
- Git, diinstal
- Instal git-remote-codecommitdi workstation lokal Anda

Arsitektur

Tumpukan teknologi target

- AWS CodeCommit repositori
- AWS CodeBuild proyek
- AWS CodePipeline pipa
- bucket Amazon Simple Storage Service (Amazon S3)
- AWS CloudFormation Template

Arsitektur target

SCSP untuk analisis kode statis adalah DevOps proyek yang dirancang untuk memberikan umpan balik keamanan pada kode yang dapat dikirim.

1. Di Konsol Manajemen AWS, masuk ke target Akun AWS. Konfirmasikan bahwa Anda berada di Wilayah AWS tempat Anda ingin menyebarkan pipa.
2. Gunakan CloudFormation template dalam repositori kode untuk menyebarkan tumpukan SCSP. Ini menciptakan CodeCommit repositori dan CodeBuild proyek baru.

> **Note**
>
> Sebagai opsi penerapan alternatif, Anda dapat menggunakan CodeCommit repo yang ada dengan menyediakan Amazon Resource Name (ARN) repositori sebagai parameter selama penerapan tumpukan.

3. Kloning repositori ke workstation lokal Anda, lalu tambahkan file apa pun ke folder masing-masing di repositori kloning.
4. Gunakan Git untuk menambahkan, mengkomit, dan mendorong file ke CodeCommit repositori.
5. Mendorong ke CodeCommit repositori memulai pekerjaan. CodeBuild CodeBuild Proyek ini menggunakan alat keamanan untuk memindai kiriman kode.
6. Tinjau output dari pipa. Alat keamanan yang menemukan masalah tingkat kesalahan akan mengakibatkan tindakan gagal dalam pipeline. Perbaiki kesalahan ini atau tekan sebagai positif palsu. Tinjau detail keluaran alat di detail Tindakan di dalam CodePipeline atau di bucket S3 pipeline.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.

Alat-alat lainnya

Untuk daftar lengkap alat yang digunakan SCSP untuk memindai kiriman kode, lihat SCSP readme di. GitHub

Repositori kode

Kode untuk pola ini tersedia di repositori Simple Code Scanning Pipeline (SCSP) di. GitHub

Epik

Menyebarkan SCSP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan.	1. Masuk ke Konsol Manajemen AWS. 2. Di konsol, konfirmasikan bahwa Anda berada di Wilayah target tempat Anda ingin menerapkan solusi. Untuk informasi selengkap nya, lihat Memilih Wilayah. 3. Pilih tautan berikut. Ini membuka wizard Quick create stack di CloudForm ation. https://console.aw s.amazon.com/cloud	AWS DevOps, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	formation/rumah? #/ stacks/create/review?templateURL=https://proservetools.s3.amazonaws.com/cft/scsp -pipeline-stack.template.json&stackname=SimpleCodeScanPipeline 4. Pada wizard Quick create stack, tinjau pengaturan parameter untuk tumpukan Anda dan buat modifikasi apa pun sesuai kebutuhan untuk kasus penggunaan Anda. 5. Pilih Saya mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM, lalu pilih Buat tumpukan. Ini membuat CodeCommit repositori, CodePipeline pipeline, beberapa definisi CodeBuild pekerjaan, dan bucket S3. Build run dan hasil pemindaian disalin ke bucket ini. Setelah CloudFormation tumpukan telah sepenuhnya digunakan, SCSP siap digunakan.	

Gunakan pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Periksa hasil pemindaian.	1. Di konsol Amazon S3, di Bucket, pilih bucket simplecodescanpipeline-deleteresourcespipelinereso. 2. Pilih direktori scan_results, lalu pilih folder dengan stempel tanggal pemindaian terbaru. 3. Tinjau file log di folder ini untuk meninjau masalah yang terdeteksi oleh alat keamanan yang digunakan dalam pipeline. Alat keamanan yang menemukan masalah tingkat kesalahan akan menghasilkan `failed` tindakan dalam pipeline. Ini perlu diperbaiki atau ditekan jika positif palsu. **Note** Anda juga dapat melihat detail output alat (untuk pemindaian yang lewat dan gagal) di CodePipeline konsol, di bagian Detail tindakan.	Pengembang aplikasi, AWS DevOps

Pemecahan masalah

Isu	Solusi
HashiCorp Terraform atau AWS CloudForm ation file tidak dipindai.	Pastikan file Terraform (.tf) dan CloudFormation (.yl, .yaml, atau .json) ditempatkan di folder yang sesuai di repositori kloning. CodeCommit
`git clone`Perintah gagal.	Pastikan Anda telah menginstal `git-remot e-codecommit` dan CLI Anda memiliki akses ke AWS kredensil yang memiliki izin untuk membaca repositori. CodeCommit
Kesalahan konkurensi, seperti`Project-level concurrent build limit cannot exceed the account-level concurrent build limit of 1 .`	Jalankan kembali pipeline dengan memilih tombol Release Change di CodePipeline konsol. Ini adalah masalah yang diketahui yang tampaknya paling umum selama beberapa kali pertama pipa berjalan.

Sumber daya terkait

Memberikan umpan balik pada proyek SCSP.

Informasi tambahan

FAQ

Apakah proyek SCSP sama dengan Automated Security Helper (ASH)?

Tidak. Gunakan ASH saat Anda menginginkan alat CLI yang menjalankan alat pemindaian kode dengan menggunakan wadah. Automated Security Helper (ASH) adalah alat yang dirancang untuk mengurangi kemungkinan pelanggaran keamanan dalam kode baru, infrastruktur, atau konfigurasi sumber daya IAM. ASH adalah utilitas baris perintah yang dapat dijalankan secara lokal. Penggunaan lokal membutuhkan lingkungan kontainer diinstal dan dioperasikan pada sistem.

Gunakan SCSP saat Anda menginginkan pipeline pengaturan yang lebih mudah daripada ASH. SCSP tidak memerlukan instalasi lokal. SCSP dirancang untuk menjalankan pemeriksaan secara

individual dalam pipeline dan menampilkan hasil dengan alat. SCSP juga menghindari banyak overhead dengan menyiapkan Docker, dan itu adalah sistem operasi (OS) agnostik.

Apakah SCSP hanya untuk tim keamanan?

Tidak, siapa pun dapat menerapkan pipeline untuk menentukan bagian mana dari kode mereka yang gagal dalam pemeriksaan keamanan. Misalnya, pengguna non-keamanan dapat menggunakan SCSP untuk memeriksa kode mereka sebelum meninjau dengan tim keamanan mereka.

Dapatkah saya menggunakan SCSP jika saya bekerja dengan jenis repositori lain, seperti,, atau GitLab Bitbucket GitHub?

Anda dapat mengonfigurasi repositori git lokal untuk menunjuk ke dua repositori jarak jauh yang berbeda. Misalnya, Anda dapat mengkloning GitLab repositori yang ada, membuat instance SCSP (menentukan, Terraform CloudFormation, dan folder AWS Config Rules Development Kit (AWS RDK), jika diperlukan), dan kemudian gunakan `git remote add upstream <SCSPGitLink>` untuk mengarahkan repositori lokal ke repositori SCSP juga. CodeCommit Ini memungkinkan perubahan kode dikirim ke SCSP terlebih dahulu, divalidasi, kemudian, setelah pembaruan tambahan dilakukan untuk mengatasi temuan, didorong ke, GitLab GitHub, atau repositori Bitbucket. Untuk informasi selengkapnya tentang beberapa remote, lihat Push commit ke repositori Git tambahan (AWS posting blog).

> **Note**
>
> Hati-hati terhadap drift, seperti menghindari membuat perubahan melalui antarmuka web.

Berkontribusi dan menambahkan tindakan Anda sendiri

Pengaturan SCSP dipertahankan sebagai GitHub proyek, yang berisi kode sumber untuk aplikasi SCSP AWS Cloud Development Kit (AWS CDK) . Untuk menambahkan pemeriksaan tambahan ke pipeline, AWS CDK aplikasi perlu diperbarui dan kemudian disintesis atau disebarkan ke target Akun AWS tempat pipeline akan berjalan. Untuk melakukan ini, mulailah dengan mengkloning GitHub proyek SCSP, dan kemudian temukan file definisi tumpukan di `lib` folder.

Jika ada pemeriksaan tambahan yang ingin Anda tambahkan, `StandardizedCodeBuildProject` kelas dalam AWS CDK kode membuatnya sangat mudah untuk menambahkan tindakan. Berikan nama, deskripsi, dan `install` atau `build` perintah. AWS CDK membuat CodeBuild proyek dengan menggunakan nilai default yang masuk akal. Selain membuat proyek build, Anda perlu

menambahkannya ke CodePipeline tindakan di tahap build. Saat merancang pemeriksaan baru, tindakan harus dilakukan `FAIL` jika alat pemindaian mendeteksi masalah atau gagal dijalankan. Tindakan harus dilakukan `PASS` jika alat pemindaian tidak mendeteksi masalah apa pun. Untuk contoh mengonfigurasi alat, tinjau kode untuk `Bandit` tindakan tersebut.

Untuk informasi selengkapnya tentang input dan output yang diharapkan, lihat dokumentasi repositori.

Jika Anda menambahkan tindakan kustom, Anda perlu menerapkan SCSP dengan menggunakan `cdk deploy` atau. `cdk synth + CloudFormation deploy` Ini karena CloudFormation template tumpukan Quick create dikelola oleh pemilik repo.

Menyebarkan kontrol akses berbasis atribut detektif untuk subnet publik dengan menggunakan AWS Config

Alberto Menendez, Amazon Web Services

Ringkasan

Arsitektur jaringan edge terdistribusi bergantung pada keamanan tepi jaringan yang berjalan di samping beban kerja di cloud pribadi virtual mereka ()VPCs. Ini memberikan skalabilitas yang belum pernah terjadi sebelumnya dibandingkan dengan pendekatan yang lebih umum dan terpusat. Meskipun menyebarkan subnet publik di akun beban kerja dapat memberikan manfaat, itu juga memperkenalkan risiko keamanan baru karena meningkatkan permukaan serangan. Kami menyarankan agar Anda hanya menggunakan sumber daya Elastic Load Balancing, seperti Application Load Balancers, atau gateway NAT di subnet publik ini. VPCs Menggunakan penyeimbang beban dan gateway NAT di subnet publik khusus membantu Anda menerapkan kontrol halus untuk lalu lintas masuk dan keluar.

Kami menyarankan Anda menerapkan kontrol preventif dan detektif untuk membatasi jenis sumber daya yang dapat digunakan di subnet publik. Untuk informasi selengkapnya tentang penggunaan kontrol akses berbasis atribut (ABAC) untuk menerapkan kontrol pencegahan untuk subnet publik, lihat Menerapkan kontrol akses berbasis atribut preventif untuk subnet publik. Meskipun efektif untuk sebagian besar situasi, kontrol pencegahan ini mungkin tidak mengatasi semua kemungkinan kasus penggunaan. Oleh karena itu, pola ini dibangun berdasarkan pendekatan ABAC dan membantu Anda mengonfigurasi peringatan tentang sumber daya yang tidak sesuai yang digunakan di subnet publik. Solusinya memeriksa apakah antarmuka jaringan elastis milik sumber daya yang tidak diizinkan di subnet publik.

Untuk mencapai ini, pola ini menggunakan aturan AWS Config khusus dan ABAC. Aturan kustom memproses konfigurasi elastic network interface setiap kali dibuat atau dimodifikasi. Pada tingkat tinggi, aturan ini melakukan dua tindakan untuk menentukan apakah antarmuka jaringan sesuai:

1. Untuk menentukan apakah antarmuka jaringan berada dalam lingkup aturan, aturan memeriksa apakah subnet memiliki AWS tag tertentu yang menunjukkan itu adalah subnet publik. Misalnya, tag ini mungkin`IsPublicFacing=True`.
2. Jika antarmuka jaringan dikerahkan di subnet publik, aturan akan memeriksa sumber daya yang Layanan AWS dibuat. Jika sumber daya bukan sumber daya Elastic Load Balancing atau gateway NAT, sumber daya tersebut menandai sumber daya sebagai tidak sesuai.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- AWS Config, atur di akun beban kerja
- Izin untuk menyebarkan sumber daya yang diperlukan di akun beban kerja
- VPC dengan subnet publik
- Tag diterapkan dengan benar untuk mengidentifikasi subnet publik target
- (Opsional) Sebuah organisasi di AWS Organizations
- (Opsional) Akun keamanan pusat yang merupakan administrator yang didelegasikan untuk AWS Config dan AWS Security Hub CSPM

Arsitektur

Arsitektur target

Diagram ini menggambarkan hal sebagai berikut:

1. Ketika sebuah elastic network interface resource (`AWS::EC2::NetworkInterface`) dikerahkan atau dimodifikasi, AWS Config menangkap peristiwa dan konfigurasi.
2. AWS Config mencocokkan acara ini dengan aturan khusus yang digunakan untuk mengevaluasi konfigurasi.
3. AWS Lambda Fungsi yang terkait dengan aturan kustom ini dipanggil. Fungsi mengevaluasi sumber daya dan menerapkan logika yang ditentukan untuk menentukan apakah konfigurasi sumber daya`COMPLIANT`, `NON_COMPLIANT` atau`NOT_APPLICABLE`.
4. Jika sumber daya ditentukan`NON_COMPLIANT`, AWS Config kirimkan peringatan melalui Amazon Simple Notification Service (Amazon SNS).

> Catatan: Jika akun ini adalah akun anggota AWS Organizations, Anda dapat mengirim data kepatuhan ke akun keamanan pusat melalui AWS Config atau AWS Security Hub CSPM.

Logika evaluasi fungsi Lambda

Diagram berikut menunjukkan logika yang diterapkan oleh fungsi Lambda untuk mengevaluasi kepatuhan antarmuka elastic network.

Otomatisasi dan skala

Pola ini adalah solusi detektif. Anda juga dapat melengkapinya dengan aturan remediasi untuk secara otomatis menyelesaikan sumber daya yang tidak sesuai. Untuk informasi selengkapnya, lihat Memulihkan Sumber Daya yang Tidak Sesuai dengan Aturan. AWS Config

Anda dapat menskalakan solusi ini dengan:

- Menegakkan penerapan AWS tag terkait yang Anda buat untuk mengidentifikasi subnet yang menghadap publik. Untuk informasi selengkapnya, lihat Kebijakan tag dalam AWS Organizations dokumentasi.
- Mengonfigurasi akun keamanan pusat yang menerapkan aturan AWS Config khusus untuk setiap akun beban kerja di organisasi. Untuk informasi selengkapnya, lihat Mengotomatiskan kepatuhan konfigurasi sesuai skala AWS (posting AWS blog).
- Mengintegrasikan AWS Config dengan AWS Security Hub CSPM untuk menangkap, memusatkan, dan memberi tahu dalam skala besar. Untuk informasi selengkapnya, lihat Mengkonfigurasi AWS Config dalam AWS Security Hub CSPM dokumentasi.

Alat

- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu.
- Elastic Load Balancing mendistribusikan lalu lintas aplikasi atau jaringan yang masuk ke beberapa target. Misalnya, Anda dapat mendistribusikan lalu lintas di seluruh instans, container, dan alamat IP Amazon Elastic Compute Cloud (Amazon EC2) di satu atau beberapa Availability Zone.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang

akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Praktik terbaik

Untuk lebih banyak contoh dan praktik terbaik untuk mengembangkan AWS Config aturan kustom, lihat Repositori AWS Config Aturan resmi di. GitHub

Epik

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat fungsi Lambda.	1. Masuk ke Konsol Manajemen AWS, lalu buka AWS Lambda konsol. 2. Pada halaman Fungsi, pilih Buat fungsi. 3. Pilih Penulis dari awal. 4. Di panel Informasi dasar, untuk nama Fungsi, masukkan nama. 5. Untuk Runtime, pilih Python 3.12. 6. Biarkan arsitektur disetel ke x86_64. 7. Pilih Buat fungsi. 8. Pilih tab Kode. 9. Di file explorer, pilih lambda_function.py. 10. Tempelkan kode sampel yang disediakan di bagian Informasi tambahan dari pola ini ke tab lambda_fu	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	nction.py. Sesuaikan kode sampel untuk mengident ifikasi logika evaluasi kustom dalam `evaluate_change_notificatio n_compliance` fungsi. 11. Pilih Deploy.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan izin ke peran eksekusi fungsi Lambda.	1. Di panel navigasi, pilih Fungsi. 2. Pilih fungsi yang baru saja Anda buat. 3. Pilih Konfigurasi, lalu pilih Izin. 4. Pilih nama peran untuk membuka peran di konsol AWS Identity and Access Management (IAM). 5. Di bawah Kebijakan izin, pilih Tambahkan izin, lalu pilih Buat kebijakan sebaris. 6. PilihJSON. 7. Tempelkan kebijakan berikut ke editor kebijakan . Ini memungkinkan fungsi Lambda untuk: • Dapatkan detail tag subnet. • Kirim hasil kepatuhan kembali ke AWS Config. `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "config:PutEvaluations",`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"ec2:DescribeSubnets"
	            ],
	            "Resource": "*",
	            "Effect": "Allow"
	        }
	    ]
	}</pre>
	8. Pilih Berikutnya.
	9. Masukkan nama untuk kebijakan tersebut, lalu pilih Buat kebijakan.	

Ambil fungsi Lambda Amazon Resource Name (ARN).	1. Buka Konsol Lambda. 2. Di panel navigasi, pilih Fungsi. 3. Pilih fungsi yang baru saja Anda buat. 4. Di bagian Ikhtisar fungsi, di bawah Fungsi ARN, salin nilainya.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan AWS Config kustom.	1. Buka konsol AWS Config. 2. Pada halaman Aturan, pilih Tambahkan aturan. 3. Pada halaman Tentukan tipe aturan, pilih Buat aturan Lambda kustom, lalu pilih Berikutnya. 4. Pada halaman Konfigurasi aturan, lakukan hal berikut: a. Masukkan nama dan deskripsi. b. Untuk AWS Lambda fungsi ARN, tempel ARN yang sebelumnya Anda salin. c. Untuk jenis Pemicu, pilih Saat konfigurasi berubah. d. Untuk Lingkup perubahan, pilih Sumber Daya. e. Untuk jenis Sumber Daya, pilih AWS EC2 NetworkInterface. f. Pilih Berikutnya. 5. Pada halaman Tinjau dan buat, verifikasi aturan Anda, lalu pilih Simpan.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan pemberitahuan.	1. Ikuti petunjuk di Membuat topik Amazon SNS untuk membuat topik Amazon SNS. 2. Ikuti petunjuk di Berlangganan topik Amazon SNS untuk mengonfigurasi titik akhir yang menerima pemberitahuan untuk topik Amazon SNS. 3. Ikuti petunjuk di Bagaimana saya bisa diberi tahu ketika AWS sumber daya tidak sesuai digunakan AWS Config untuk mengonfigurasi EventBridge aturan Amazon khusus untuk sumber daya Anda yang tidak patuh.	AWS Umum

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sumber daya yang sesuai.	1. Gunakan petunjuk berikut untuk membuat salah satu sumber daya yang didukung di subnet publik: • Buat gateway NAT • Memulai Penyeimbang Beban Jaringan • Buat Application Load Balancer	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Setelah sumber daya dibuat, aturan AWS Config kustom mengevaluasi antarmuka jaringan elastis yang terkait dengan sumber daya. Ini menandai antarmuka jaringan ini sebagai`COMPLIANT` . Anda dapat melihat sumber daya AWS Config dengan mengikuti langkah-langkah berikut: a. Buka konsol AWS Config. b. Pada halaman Aturan, pilih aturan Anda. c. Pada halaman Rule Detail, pergi ke bagian bawah halaman. d. Di bawah Resources in scope, pilih Compliant . Konfirmasikan bahwa Anda melihat antarmuka jaringan yang dibuat. IDs e. Untuk detail lebih lanjut tentang konfigurasi antarmuka jaringan, pilih ID sumber daya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat sumber daya yang tidak sesuai.	1. Gunakan petunjuk berikut untuk membuat sumber daya yang tidak sesuai di subnet publik: • Luncurkan EC2 instans Amazon • Membuat instance database Amazon Relational Database Service (Amazon RDS) • Buat titik akhir VPC 2. Setelah sumber daya dibuat, aturan AWS Config kustom mengevaluasi antarmuka jaringan elastis yang terkait dengan sumber daya. Ini menandai antarmuka jaringan ini sebagai`NON_COMPLIANT` . Anda dapat melihat sumber daya AWS Config dengan mengikuti langkah-langkah berikut: a. Buka konsol AWS Config. b. Pada halaman Aturan, pilih aturan Anda. c. Pada halaman Rule Detail, pergi ke bagian bawah halaman. d. Di bawah Resources in scope, pilih NonCompli ant. Konfirmasikan	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	bahwa Anda melihat antarmuka jaringan yang dibuat. IDs e. Untuk detail lebih lanjut tentang konfigurasi antarmuka jaringan, pilih ID sumber daya. 3. Konfirmasikan bahwa Anda menerima pemberitahuan di titik akhir yang Anda konfigurasikan di Amazon SNS.	
Buat sumber daya yang tidak berlaku.	1. Dalam subnet pribadi, buat sumber daya apa pun yang membutuhkan elastic network interface. 2. Setelah sumber daya dibuat, aturan AWS Config kustom mengevaluasi antarmuka jaringan elastis yang terkait dengan sumber daya. Ini menandai antarmuka jaringan ini sebagai`NOT_APPLICABLE`. Sumber daya ini tidak ditampilkan di AWS Config konsol.	AWS Umum

Sumber daya terkait

AWS dokumentasi

- Menyiapkan AWS Config

- AWS Config aturan kustom
- ABAC untuk AWS
- Menyebarkan kontrol akses berbasis atribut preventif untuk subnet publik

AWS Sumber daya lainnya

- Otomatiskan kepatuhan konfigurasi pada skala besar AWS
- Arsitektur Inspeksi Terdistribusi dengan Load Balancer Gateway

Informasi tambahan

Berikut ini adalah contoh fungsi Lambda yang disediakan untuk tujuan demonstrasi.

```
import boto3
import json
import os

# Init clients
config_client = boto3.client('config')
ec2_client = boto3.client('ec2')

def lambda_handler(event, context):

    # Init values
    compliance_value = 'NOT_APPLICABLE'
    invoking_event = json.loads(event['invokingEvent'])
    configuration_item = invoking_event['configurationItem']

    status = configuration_item['configurationItemStatus']
    eventLeftScope = event['eventLeftScope']

    # First check if the event configuration applies. Ex. resource event is not delete
    if (status == 'OK' or status == 'ResourceDiscovered') and not eventLeftScope:
        compliance_value = evaluate_change_notification_compliance(configuration_item)


    config_client.put_evaluations(
       Evaluations=[
           {
```

```
                'ComplianceResourceType': invoking_event['configurationItem']
['resourceType'],
                'ComplianceResourceId': invoking_event['configurationItem']
['resourceId'],
                'ComplianceType': compliance_value,
                'OrderingTimestamp': invoking_event['configurationItem']
['configurationItemCaptureTime']
            },
        ],
        ResultToken=event['resultToken'])

# Function with the logs to evaluate the resource
def evaluate_change_notification_compliance(configuration_item):
    is_in_scope = is_in_scope_subnet(configuration_item['configuration']['subnetId'])

    if (configuration_item['resourceType'] != 'AWS::EC2::NetworkInterface') or not
 is_in_scope:
        return 'NOT_APPLICABLE'

    else:
        alb_condition = configuration_item['configuration']['requesterId'] in ['amazon-
elb']
        nlb_condition = configuration_item['configuration']['interfaceType'] in
 ['network_load_balancer']
        nat_gateway_condition = configuration_item['configuration']['interfaceType'] in
 ['nat_gateway']

        if alb_condition or nlb_condition or nat_gateway_condition:
            return 'COMPLIANT'
    return 'NON_COMPLIANT'

# Function to check if elastic network interface is in public subnet
def is_in_scope_subnet(eni_subnet):

    subnet_description = ec2_client.describe_subnets(
        SubnetIds=[eni_subnet]
    )

    for subnet in subnet_description['Subnets']:
        for tag in subnet['Tags']:
            if tag['Key'] == os.environ.get('TAG_KEY') and tag['Value'] ==
 os.environ.get('TAG_VALUE'):
                return True
```

```
    return False
```

Menyebarkan kontrol akses berbasis atribut preventif untuk subnet publik

Joel Alfredo Nunez Gonzalez dan Samuel Ortega Sancho, Amazon Web Services

Ringkasan

Dalam arsitektur jaringan terpusat, inspeksi dan edge virtual private cloud (VPCs) memusatkan semua lalu lintas masuk dan keluar, seperti lalu lintas ke dan dari internet. Namun, hal ini dapat menimbulkan kemacetan atau mengakibatkan tercapainya batas kuota layanan AWS. Menerapkan keamanan tepi jaringan di samping beban kerja di dalamnya VPCs memberikan skalabilitas yang belum pernah terjadi sebelumnya dibandingkan dengan pendekatan terpusat yang lebih umum. Ini disebut arsitektur tepi terdistribusi.

Meskipun menyebarkan subnet publik di akun beban kerja dapat memberikan manfaat, itu juga memperkenalkan risiko keamanan baru karena meningkatkan permukaan serangan. Kami menyarankan agar Anda hanya menggunakan sumber daya Elastic Load Balancing (ELB), seperti Application Load Balancers, atau gateway NAT di subnet publik ini. VPCs Menggunakan penyeimbang beban dan gateway NAT dalam subnet publik khusus membantu Anda menerapkan kontrol halus untuk lalu lintas masuk dan keluar.

Attribute-based access control (ABAC) adalah praktik membuat izin berbutir halus berdasarkan atribut pengguna, seperti departemen, peran pekerjaan, dan nama tim. Untuk informasi selengkapnya, lihat ABAC for AWS. ABAC dapat menyediakan pagar pembatas untuk subnet publik di akun beban kerja. Ini membantu tim aplikasi menjadi gesit, tanpa mengorbankan keamanan infrastruktur.

Pola ini menjelaskan cara membantu mengamankan subnet publik dengan menerapkan ABAC melalui kebijakan kontrol layanan (SCP) di AWS Organizations dan kebijakan di AWS Identity and Access Management (IAM). Anda menerapkan SCP ke akun anggota organisasi atau unit organisasi (OU). Kebijakan ABAC ini memungkinkan pengguna untuk menerapkan gateway NAT di subnet target dan mencegahnya menerapkan sumber daya Amazon Elastic Compute Cloud (Amazon EC2) lainnya, seperti instance dan antarmuka jaringan elastis. EC2

Prasyarat dan batasan

Prasyarat

- Organisasi di AWS Organizations
- Akses administratif ke akun root AWS Organizations
- Di organisasi, akun anggota aktif atau OU untuk menguji SCP

Batasan

- SCP dalam solusi ini tidak mencegah layanan AWS yang menggunakan peran terkait layanan menerapkan sumber daya di subnet target. Contoh layanan ini adalah Elastic Load Balancing (ELB), Amazon Elastic Container Service (Amazon ECS) Container Service (Amazon ECS), dan Amazon Relational Database Service (Amazon RDS). Untuk informasi selengkapnya, lihat efek SCP pada izin dalam dokumentasi AWS Organizations. Menerapkan kontrol keamanan untuk mendeteksi pengecualian ini.

Arsitektur

Tumpukan teknologi target

- SCP diterapkan ke akun AWS atau OU di AWS Organizations
- Peran IAM berikut:
  - `AutomationAdminRole`— Digunakan untuk memodifikasi tag subnet dan membuat sumber daya VPC setelah menerapkan SCP
  - `TestAdminRole`— Digunakan untuk menguji apakah SCP mencegah prinsipal IAM lainnya, termasuk mereka yang memiliki akses administratif, dari melakukan tindakan yang disediakan untuk `AutomationAdminRole`

Arsitektur target

1. Anda membuat peran `AutomationAdminRole` IAM di akun target. Peran ini memiliki izin untuk mengelola sumber daya jaringan. Perhatikan izin berikut yang eksklusif untuk peran ini:
   - Peran ini dapat membuat VPCs dan subnet publik.
   - Peran ini dapat memodifikasi penetapan tag untuk subnet target.
   - Peran ini dapat mengelola izinnya sendiri.
2. Di AWS Organizations, Anda menerapkan SCP ke akun AWS target atau OU. Untuk kebijakan sampel, lihat Informasi tambahan dalam pola ini.

3. Seorang pengguna atau alat dalam CI/CD pipeline dapat mengambil `AutomationAdminRole` peran untuk menerapkan `SubnetType` tag ke subnet target.
4. Dengan mengasumsikan peran IAM lainnya, prinsipal IAM resmi di organisasi Anda dapat mengelola gateway NAT di subnet target dan sumber daya jaringan lain yang diizinkan di akun AWS, seperti tabel rute. Gunakan kebijakan IAM untuk memberikan izin ini. Untuk informasi selengkapnya, lihat Identitas dan manajemen akses untuk Amazon VPC.

Otomatisasi dan skala

Untuk membantu melindungi subnet publik, tag AWS yang sesuai harus diterapkan. Setelah menerapkan SCP, gateway NAT adalah satu-satunya jenis EC2 sumber daya Amazon yang dapat dibuat oleh pengguna resmi di subnet yang memiliki tag. `SubnetType:IFA` (`IFA`berarti aset yang menghadap internet.) SCP mencegah pembuatan EC2 sumber daya Amazon lainnya, seperti instance dan antarmuka jaringan elastis. Kami menyarankan Anda menggunakan CI/CD pipeline yang mengasumsikan `AutomationAdminRole` peran untuk membuat sumber daya VPC sehingga tag ini diterapkan dengan benar ke subnet publik.

Alat

Layanan AWS

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Organizations adalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat. Di AWS Organizations, Anda dapat menerapkan kebijakan kontrol layanan (SCPs), yang merupakan jenis kebijakan yang dapat Anda gunakan untuk mengelola izin di organisasi Anda.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan sumber daya AWS ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur AWS yang dapat diskalakan.

Epik

Terapkan SCP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran admin pengujian.	Buat peran IAM yang dinamai `TestAdminRole` di akun AWS target. Lampirkan kebijakan IAM terkelola AdministratorAccessAWS ke peran baru. Untuk petunjuk, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM dalam dokumentasi IAM.	Administrator AWS
Buat peran admin otomatisasi.	1. Buat peran IAM yang dinamai `AutomationAdminRole` di akun AWS target. 2. Lampirkan kebijakan IAM terkelola AdministratorAccessAWS ke peran baru. Berikut ini adalah contoh kebijakan kepercayaan yang dapat Anda gunakan untuk menguji peran dari `111122223333` akun. `{ "Version": "2012-10-17", "Statement": [ {`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::111122223333:root"] }, "Action": "sts:AssumeRole", "Condition": {} }] }</pre>	
Buat dan lampirkan SCP.	1. Menggunakan kode sampel yang disediakan di bagian Informasi tambahan, buat kebijakan kontrol keamanan. Untuk petunjukn ya, lihat Membuat SCP di dokumentasi AWS Organizations. 2. Lampirkan SCP ke akun AWS target atau OU. Untuk petunjuk, lihat Melampirkan dan melepaskan kebijakan kontrol layanan dalam dokumentasi AWS Organizations.	Administrator AWS

Uji SCP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat VPC atau subnet.	1. Asumsikan `TestAdmin Role` peran dalam akun AWS target. 2. Cobalah untuk membuat VPC atau subnet publik baru di VPC yang ada. Untuk petunjuknya, lihat Membuat VPC, subnet, dan sumber daya VPC lainnya di dokumentasi Amazon VPC. Anda seharusny a tidak dapat membuat sumber daya ini. 3. Asumsikan `Automatio nAdminRole` peran, dan coba lagi langkah sebelumnya. Sekarang, Anda harus dapat membuat sumber daya jaringan.	Administrator AWS
Kelola tag.	1. Asumsikan `TestAdmin Role` peran dalam akun AWS target. 2. Tambahkan `SubnetTyp e:IFA` tag ke subnet publik yang tersedia. Anda harus dapat menambahk an tag ini. Untuk petunjuk tentang cara menambahkan tag melalui AWS Command Line Interface (AWS CLI) Interface, lihat membuat	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tag di AWS CLI Command Reference. 3. Tanpa mengubah kredensil Anda, cobalah untuk memodifikasi `SubnetType:IFA` tag yang ditetapkan ke subnet ini. Anda seharusnya tidak dapat memodifikasi tag ini. 4. Asumsikan `AutomationAdminRole` peran, dan coba lagi langkah-langkah sebelumnya. Peran ini harus dapat menambahkan dan memodifikasi tag ini.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan sumber daya di subnet target.	1. Asumsikan `TestAdmin Role` peran. 2. Untuk subnet publik yang memiliki `SubnetTyp e:IFA` tag, coba buat EC2 instance. Untuk petunjukn ya, lihat Meluncurkan instance di EC2 dokumenta si Amazon. Di subnet ini, Anda seharusnya tidak dapat membuat, memodifik asi, atau menghapus EC2 sumber daya Amazon apa pun kecuali gateway NAT. 3. Di subnet yang sama, buat gateway NAT. Untuk petunjuknya, lihat Membuat gateway NAT di dokumentasi Amazon VPC. Anda harus dapat membuat, memodifikasi, atau menghapus gateway NAT di subnet ini.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kelola AutomationAdminRole peran.	1. Asumsikan `TestAdmin Role` peran. 2. Cobalah untuk memodifik asi `Automatio nAdminRole` peran. Untuk petunjuk, lihat Memodifikasi peran dalam dokumentasi IAM. Anda seharusnya tidak dapat mengubah peran ini. 3. Asumsikan `Automatio nAdminRole` peran, dan coba lagi langkah sebelumnya. Sekarang, Anda harus dapat memodifikasi peran.	Administrator AWS

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bersihkan sumber daya yang digunakan.	1. Lepaskan SCP dari akun AWS atau OU. Untuk petunjuknya, lihat Melepaskan SCP dalam dokumentasi AWS Organizations. 2. Hapus SCP. Untuk petunjuk, lihat Menghapus SCP (dokumentasi AWS Organizations). 3. Hapus `Automatio nAdminRole` peran	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan `TestAdminRole` peran. Untuk petunjuk, lihat Menghapus peran dalam dokumentasi IAM. 4. Hapus semua sumber daya jaringan, seperti VPCs dan subnet, yang Anda buat untuk solusi ini.	

Sumber daya terkait

Dokumentasi AWS

- Melampirkan dan melepaskan SCPs
- Membuat, memperbarui, dan menghapus SCPs
- Menerapkan kontrol akses berbasis atribut detektif untuk subnet publik dengan menggunakan AWS Config
- Kontrol detektif
- Referensi otorisasi layanan
- Menandai sumber daya AWS
- Apa itu ABAC untuk AWS?

Referensi AWS tambahan

- Mengamankan tag sumber daya yang digunakan untuk otorisasi menggunakan Kebijakan Kontrol Layanan di AWS Organizations (postingan blog AWS)

Informasi tambahan

Kebijakan kontrol layanan berikut adalah contoh yang dapat Anda gunakan untuk menguji pendekatan ini di organisasi Anda.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyVPCActions",
      "Effect": "Deny",
      "Action": [
        "ec2:CreateVPC",
        "ec2:CreateRoute",
        "ec2:CreateSubnet",
        "ec2:CreateInternetGateway",
        "ec2:DeleteVPC",
        "ec2:DeleteRoute",
        "ec2:DeleteSubnet",
        "ec2:DeleteInternetGateway"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:*"
      ],
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
        }
      }
    },
    {
      "Sid": "AllowNATGWOnIFASubnet",
      "Effect": "Deny",
      "NotAction": [
        "ec2:CreateNatGateway",
        "ec2:DeleteNatGateway"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*"
      ],
      "Condition": {
        "ForAnyValue:StringEqualsIfExists": {
          "aws:ResourceTag/SubnetType": "IFA"
        },
        "StringNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
        }
      }
    },
    {
```

```
      "Sid": "DenyChangesToAdminRole",
      "Effect": "Deny",
      "NotAction": [
        "iam:GetContextKeysForPrincipalPolicy",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListAttachedRolePolicies",
        "iam:ListInstanceProfilesForRole",
        "iam:ListRolePolicies",
        "iam:ListRoleTags"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AutomationAdminRole"
      ],
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
        }
      }
    }
  ]
}
```

Mendeteksi instans database Amazon RDS dan Aurora yang memiliki sertifikat CA kedaluwarsa

Stephen DiCato dan Eugene Shifer, Amazon Web Services

Ringkasan

Sebagai praktik terbaik keamanan, disarankan agar Anda mengenkripsi data dalam perjalanan antara server aplikasi dan database relasional. Anda dapat menggunakan SSL atau TLS untuk mengenkripsi koneksi ke instance database (DB) atau cluster. Protokol ini membantu memberikan kerahasiaan, integritas, dan keaslian antara aplikasi dan database. Database menggunakan sertifikat server, yang dikeluarkan oleh otoritas sertifikat (CA) dan digunakan untuk melakukan verifikasi identitas server. SSL atau TLS memverifikasi keaslian sertifikat dengan memvalidasi tanda tangan digitalnya dan memastikannya tidak kedaluwarsa.

Dalam Konsol Manajemen AWS, Amazon Relational Database Service (Amazon RDS) dan Amazon Aurora memberikan pemberitahuan tentang instans DB yang memerlukan pembaruan sertifikat. Namun, untuk memeriksa pemberitahuan ini, Anda harus masuk ke masing-masing Akun AWS dan menavigasi ke konsol layanan di masing-masing Wilayah AWS. Tugas ini menjadi lebih kompleks jika Anda perlu menilai validitas sertifikat di banyak Akun AWS yang dikelola sebagai organisasi di AWS Organizations.

Dengan menyediakan infrastruktur sebagai kode (IAc) yang disediakan dalam pola ini, Anda dapat mendeteksi sertifikat CA yang kedaluwarsa untuk semua instans Amazon RDS dan Aurora DB di organisasi atau Anda. Akun AWS AWS AWS CloudFormationTemplate menyediakan AWS Config aturan, AWS Lambda fungsi, dan izin yang diperlukan. Anda dapat menerapkannya ke dalam satu akun sebagai tumpukan, atau Anda dapat menerapkannya di seluruh AWS organisasi sebagai kumpulan tumpukan.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Jika Anda menerapkan ke dalam satu Akun AWS:
  - Pastikan Anda memiliki izin untuk membuat CloudFormation tumpukan.
  - Aktifkan AWS Config di akun target.

  - (Opsional) Aktifkan AWS Security Hub CSPM di akun target.
- Jika Anda menerapkan ke dalam AWS organisasi:
  - Pastikan Anda memiliki izin untuk membuat kumpulan CloudFormation tumpukan.
  - Aktifkan Security Hub CSPM dengan AWS Organizations integrasi.
  - Aktifkan AWS Config di akun tempat Anda menerapkan solusi ini.
  - Tentukan Akun AWS untuk menjadi administrator yang didelegasikan untuk AWS Config dan Security Hub CSPM.

Batasan

- Jika Anda menerapkan ke akun individual yang tidak mengaktifkan CSPM Security Hub, Anda dapat menggunakannya AWS Config untuk mengevaluasi temuan.
- Jika Anda menerapkan ke organisasi yang tidak memiliki administrator yang didelegasikan untuk AWS Config dan CSPM Security Hub, Anda harus masuk ke akun anggota individu untuk melihat temuan.
- Jika Anda menggunakannya AWS Control Tower untuk mengelola dan mengatur akun di organisasi Anda, gunakan IAc dalam pola ini dengan menggunakan Kustomisasi untuk (CFCT). AWS Control Tower Menggunakan CloudFormation konsol akan membuat penyimpangan konfigurasi dari AWS Control Tower pagar pembatas dan mengharuskan Anda mendaftarkan ulang unit organisasi () OUs atau akun terkelola.
- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat halaman titik akhir dan kuota Layanan, dan pilih tautan untuk layanan.

Arsitektur

Menyebarkan menjadi individu Akun AWS

Diagram arsitektur berikut menunjukkan penyebaran sumber AWS daya dalam satu Akun AWS. Ini diimplementasikan dengan menggunakan CloudFormation template langsung melalui CloudFormation konsol. Jika Security Hub CSPM diaktifkan, Anda dapat melihat hasilnya di salah satu AWS Config atau Security Hub CSPM. Jika Security Hub CSPM tidak diaktifkan, Anda dapat melihat hasilnya hanya di konsol. AWS Config

Diagram menunjukkan langkah-langkah berikut:

1. Anda membuat CloudFormation tumpukan. Ini menyebarkan fungsi Lambda dan AWS Config aturan. Aturan dan fungsi diatur dengan izin AWS Identity and Access Management (IAM) yang diperlukan untuk mempublikasikan evaluasi sumber daya dan log AWS Config .
2. AWS Config Aturan ini beroperasi dalam mode evaluasi detektif dan berjalan setiap 24 jam.
3. Security Hub CSPM menerima semua AWS Config temuan.
4. Anda dapat melihat temuan di Security Hub CSPM atau di AWS Config, tergantung pada konfigurasi akun.

Menyebarkan ke dalam organisasi AWS

Diagram berikut menunjukkan penilaian kedaluwarsa sertifikat di beberapa akun yang dikelola melalui AWS Organizations dan. AWS Control Tower Anda menyebarkan CloudFormation template melalui CFCT. Hasil penilaian terpusat di Security Hub CSPM di akun administrator yang didelegasikan. AWS CodePipeline Alur kerja yang digambarkan dalam diagram menunjukkan langkah-langkah latar belakang yang terjadi selama penyebaran CFCT.

Diagram menunjukkan langkah-langkah berikut:

1. Bergantung pada konfigurasi untuk CFCT, di akun manajemen, Anda mendorong IAc ke AWS CodeCommit repositori atau Anda mengunggah file terkompresi (ZIP) dari IAc ke bucket Amazon Simple Storage Service (Amazon S3).
2. Pipeline CFCT membuka ritsleting file, menjalankan pemeriksaan cfn-nag (GitHub), dan menerapkannya sebagai kumpulan tumpukan. CloudFormation
3. Bergantung pada konfigurasi yang ditentukan dalam file manifes CFCT, CloudFormation StackSets menyebarkan tumpukan ke akun individual atau ditentukan. OUs Ini menyebarkan fungsi Lambda dan aturan di AWS Config akun target. Aturan dan fungsi diatur dengan izin IAM yang diperlukan untuk mempublikasikan evaluasi sumber daya dan log AWS Config .
4. AWS Config Aturan ini beroperasi dalam mode evaluasi detektif dan berjalan setiap 24 jam.
5. AWS Config meneruskan semua temuan ke Security Hub CSPM.
6. Temuan CSPM Security Hub dikumpulkan dalam akun administrator yang didelegasikan.
7. Anda dapat melihat temuan di Security Hub CSPM di akun administrator yang didelegasikan.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu. AWS Config Aturan menentukan pengaturan konfigurasi ideal Anda untuk sumber daya, dan AWS Config dapat mengevaluasi apakah AWS sumber daya Anda mematuhi ketentuan dalam aturan Anda.
- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik preskriptif. Kustomisasi untuk AWS Control Tower (CFCT) membantu Anda menyesuaikan AWS Control Tower landing zone dan tetap selaras dengan praktik terbaik. AWS Kustomisasi diimplementasikan dengan CloudFormation template dan kebijakan kontrol layanan ()SCPs.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS Security Hub CSPMmemberikan pandangan komprehensif tentang keadaan keamanan Anda di AWS. Ini juga membantu Anda memeriksa AWS lingkungan Anda terhadap standar industri keamanan dan praktik terbaik.

Alat-alat lainnya

- Python adalah bahasa pemrograman komputer tujuan umum.

Repositori kode

Kode untuk pola ini tersedia di instans GitHub Deteksi Amazon RDS dengan repositori sertifikat CA yang kedaluwarsa.

Praktik terbaik

Kami menyarankan Anda mematuhi praktik terbaik dalam sumber daya berikut:

- Praktik Terbaik untuk Unit Organisasi dengan AWS Organizations (AWS Cloud Operations & Migrations Blog)
- Panduan untuk Mendirikan Yayasan Awal menggunakan AWS Control Tower on AWS (Perpustakaan AWS Solusi)
- Panduan untuk membuat dan memodifikasi AWS Control Tower sumber daya (AWS Control Tower dokumentasi)
- Pertimbangan penyebaran CFCT (dokumentasi)AWS Control Tower

Epik

Tinjau solusi dan kodenya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan strategi penyebaran Anda.	Tinjau solusi dan kode untuk menentukan bagaimana Anda akan menerapkannya ke AWS lingkungan Anda. Tentukan apakah Anda akan menyebarkan ke dalam satu akun atau AWS organisasi.	Pemilik aplikasi, AWS Umum
Kloning repositori.	Masukkan perintah berikut untuk mengkloning instans Deteksi Amazon RDS dengan repositori sertifikat CA yang kedaluwarsa. `git clone https://github.com/aws-samples/config-rds-ca-expiry.git`	Pengembang aplikasi, Pemilik aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi versi Python.	1. Arahkan ke direktori tingkat atas di repositori kloning. `cd config-rds-ca-expiry` 2. Buka config-rds-ca-expiry.yaml. 3. Di `CertExpirationCheckLambdaFunction` sumber daya, konfirmasikan bahwa versi Python kompatibel dengan target Anda. Wilayah AWS Secara default, fungsi ini menggunakan Python 3.12. Untuk informasi selengkapnya, lihat AWS Lambda menambahkan dukungan untuk Python 3.12. Jika perlu, perbarui versi Python. 4. Simpan dan tutup config-rds-ca-expiry.yaml.	Pengembang aplikasi, Pemilik aplikasi

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	Menyebarkan CloudFormation template ke AWS lingkungan Anda. Lakukan salah satu tindakan berikut:	Pengembang aplikasi, administrator AWS, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Jika Anda menerapkan ke satu Akun AWS, ikuti petunjuk dalam Membuat tumpukan. • Jika Anda menerapkan ke organisasi yang tidak dikelola oleh AWS Control Tower, ikuti petunjuk di Buat kumpulan tumpukan. • Jika Anda menerapkan ke organisasi yang dikelola oleh AWS Control Tower, lihat petunjuk di Membangun kustomisasi Anda sendiri.	
Verifikasi penyebaran.	Di CloudFormation konsol, verifikasi bahwa kumpulan tumpukan atau tumpukan telah berhasil diterapkan.	Administrator AWS, Pemilik aplikasi

Tinjau temuannya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lihat temuan AWS Config aturan.	Di Security Hub CSPM, lakukan hal berikut untuk melihat daftar temuan individu: 1. Buka konsol CSPM Security Hub. 2. Di panel navigasi, pilih Temuan.	Administrator AWS, administrator sistem AWS, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di kotak Tambahkan filter, tambahkan filter berikut: • Status Kepatuhan adalah `FAILED` • Judul adalah `rds-has-expiring-ca` 4. Pilih Terapkan. Di Security Hub CSPM, lakukan hal berikut untuk melihat daftar total temuan yang dikelompokkan berdasarkan: Akun AWS 1. Buka konsol CSPM Security Hub. 2. Pada panel navigasi, silakan pilih Wawasan. 3. Pilih Buat wawasan. 4. Untuk memilih atribut pengelompokan untuk wawasan: a. Pilih kotak pencarian untuk menampilkan opsi filter. b. Pilih Grup berdasarkan. c. Pilih AwsAccountId. d. Pilih Terapkan. 5. Di kotak Tambahkan filter, tambahkan filter berikut: • Judul adalah `rds-has-expiring-ca`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Status Kepatuhan adalah `FAILED` 6. Pilih Buat wawasan. 7. Masukkan nama Insight, lalu pilih Buat wawasan. Di AWS Config, untuk melihat daftar temuan, ikuti petunjuk dalam Melihat Informasi Kepatuhan dan Hasil Evaluasi dalam AWS Config dokumentasi.	

Pemecahan masalah

Isu	Solusi
CloudFormation pembuatan atau penghapusan set tumpukan gagal	Ketika AWS Control Tower dikerahkan, itu memberlakukan pagar pembatas yang diperlukan dan mengasumsikan kontrol atas agregator dan aturan. AWS Config Ini termasuk mencegah perubahan langsung melalui CloudFormation. Untuk menerapkan atau menghapus CloudFormation template ini dengan benar, termasuk semua sumber daya terkait, Anda harus menggunakan CFCT.
CFCT gagal menghapus template CloudFormation	Jika CloudFormation template tetap ada bahkan setelah membuat perubahan yang diperlukan dalam file manifes dan menghapus file template, konfirmasikan bahwa file manifes berisi `enable_stack_set_deletion` parameter dan nilainya disetel `kefalse`. Untuk

Isu	Solusi
	informasi selengkapnya, lihat Menghapus kumpulan tumpukan dalam dokumentasi CFCT.

Sumber daya terkait

- Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans atau cluster DB (dokumentasi Amazon RDS)
- AWS Config Aturan Kustom (AWS Config dokumentasi)

Buat kebijakan IAM secara dinamis dengan IAM Access Analyzer dengan menggunakan Step Functions

Thomas Scott, Koen Blijderveen, Adil El Kanabi, dan Rafal Pawlaszek, Amazon Web Services

Ringkasan

Least-privilege adalah praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Menerapkan akses hak istimewa paling sedikit di akun Amazon Web Services (AWS) yang sudah aktif dapat menjadi tantangan karena Anda tidak ingin memblokir pengguna secara tidak sengaja untuk melakukan tugas pekerjaan mereka dengan mengubah izin mereka. Sebelum Anda dapat menerapkan perubahan kebijakan AWS Identity and Access Management (IAM), Anda perlu memahami tindakan dan sumber daya yang dilakukan pengguna akun.

Pola ini dirancang untuk membantu Anda menerapkan prinsip akses hak istimewa paling sedikit, tanpa memblokir atau memperlambat produktivitas tim. Ini menjelaskan cara menggunakan IAM Access Analyzer dan AWS Step Functions secara dinamis menghasilkan kebijakan up-to-date IAM untuk peran Anda, berdasarkan tindakan yang saat ini sedang dilakukan di akun. Kebijakan baru dirancang untuk mengizinkan aktivitas saat ini tetapi menghapus hak istimewa yang tidak perlu dan ditinggikan. Anda dapat menyesuaikan kebijakan yang dihasilkan dengan menentukan aturan izinkan dan tolak, dan solusinya mengintegrasikan aturan kustom Anda.

Pola ini mencakup opsi untuk mengimplementasikan solusi dengan AWS Cloud Development Kit (AWS CDK) atau HashiCorp CDK untuk Terraform (CDKTF). Anda kemudian dapat mengaitkan kebijakan baru dengan peran dengan menggunakan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD). Jika Anda memiliki arsitektur multi-akun, Anda dapat menerapkan solusi ini di akun mana pun yang ingin Anda buat kebijakan IAM yang diperbarui untuk peran tersebut, meningkatkan keamanan seluruh lingkungan Anda. AWS Cloud

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS dengan AWS CloudTrail jejak diaktifkan.
- Izin IAM untuk hal berikut:
  - Membuat dan menerapkan alur kerja Step Functions. Untuk informasi selengkapnya, lihat Kunci tindakan, sumber daya, dan kondisi untuk AWS Step Functions (dokumentasi Step Functions).

  - Buat AWS Lambda fungsi. Untuk informasi selengkapnya, lihat Peran eksekusi dan izin pengguna (dokumentasi Lambda).
  - Buat IAM role. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM (dokumentasi IAM).
- npm diinstal. Untuk informasi selengkapnya, lihat Mengunduh dan menginstal Node.js dan npm (dokumentasi npm).
- Jika Anda menerapkan solusi ini dengan AWS CDK (Opsi 1):
  - AWS CDK Toolkit, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal AWS CDK (AWS CDK dokumentasi).
- Jika Anda menerapkan solusi ini dengan CDKTF (Opsi 2):
  - CDKTF, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal CDK untuk Terraform (dokumentasi CDKTF).
  - Terraform, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Memulai (dokumentasi Terraform).
- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi secara lokal untuk Anda Akun AWS. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui versi terbaru AWS CLI (AWS CLI dokumentasi).

Batasan

- Pola ini tidak menerapkan kebijakan IAM baru untuk peran tersebut. Pada akhir solusi ini, kebijakan IAM baru disimpan dalam AWS CodeCommit repositori. Anda dapat menggunakan CI/CD pipeline untuk menerapkan kebijakan ke peran di akun Anda.

Arsitektur

Arsitektur target

1. Aturan EventBridge acara Amazon yang dijadwalkan secara teratur memulai alur kerja Step Functions. Anda mendefinisikan jadwal regenerasi ini sebagai bagian dari pengaturan solusi ini.
2. Dalam alur kerja Step Functions, fungsi Lambda menghasilkan rentang tanggal yang akan digunakan saat menganalisis aktivitas akun di CloudTrail log.

3. Langkah alur kerja berikutnya memanggil IAM Access Analyzer API untuk mulai membuat kebijakan.
4. Menggunakan Nama Sumber Daya Amazon (ARN) peran yang Anda tentukan selama penyiapan, IAM Access Analyzer menganalisis CloudTrail log untuk aktivitas dalam tingkat tanggal yang ditentukan. Berdasarkan aktivitas, IAM Access Analyzer menghasilkan kebijakan IAM yang hanya mengizinkan tindakan dan layanan yang digunakan oleh peran selama rentang tanggal yang ditentukan. Ketika langkah ini selesai, langkah ini menghasilkan ID pekerjaan.
5. Langkah alur kerja berikutnya memeriksa ID pekerjaan setiap 30 detik. Ketika ID pekerjaan terdeteksi, langkah ini menggunakan ID pekerjaan untuk memanggil IAM Access Analyzer API dan mengambil kebijakan IAM baru. IAM Access Analyzer mengembalikan kebijakan sebagai file JSON.
6. Langkah alur kerja berikutnya menempatkan file /policy.json <IAM role name>di bucket Amazon Simple Storage Service (Amazon S3). Anda mendefinisikan bucket S3 ini sebagai bagian dari pengaturan solusi ini.
7. Pemberitahuan acara Amazon S3 memulai fungsi Lambda.
8. Fungsi Lambda mengambil kebijakan dari bucket S3, mengintegrasikan aturan kustom yang Anda tentukan dalam file allow.json dan deny.json, lalu mendorong kebijakan yang diperbarui. CodeCommit Anda mendefinisikan jalur CodeCommit repositori, cabang, dan folder sebagai bagian dari pengaturan solusi ini.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS CDK Toolkit adalah kit pengembangan cloud baris perintah yang membantu Anda berinteraksi dengan AWS Cloud Development Kit (AWS CDK) aplikasi Anda.
- AWS CloudTrailmembantu Anda mengaudit tata kelola, kepatuhan, dan risiko operasional Anda Akun AWS.
- AWS CodeCommitadalah layanan kontrol versi yang membantu Anda menyimpan dan mengelola repositori Git secara pribadi, tanpa perlu mengelola sistem kontrol sumber Anda sendiri.
- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya. Pola ini menggunakan IAM Access Analyzer, fitur IAM, untuk menganalisis CloudTrail log Anda guna mengidentifikasi tindakan dan layanan yang telah digunakan oleh entitas IAM (pengguna atau peran) dan kemudian menghasilkan kebijakan IAM yang didasarkan pada aktivitas tersebut.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Step Functionsadalah layanan orkestrasi tanpa server yang membantu Anda menggabungkan AWS Lambda fungsi dan lainnya Layanan AWS untuk membangun aplikasi bisnis yang penting. Dalam pola ini, Anda menggunakan integrasi layanan AWS SDK di Step Functions untuk memanggil tindakan API layanan dari alur kerja Anda.

Alat-alat lainnya

- CDK for Terraform (CDKTF) membantu Anda mendefinisikan infrastruktur sebagai kode (IAc) dengan menggunakan bahasa pemrograman umum, seperti Python dan TypeScript.
- Lerna adalah sistem build untuk mengelola dan menerbitkan beberapa JavaScript atau TypeScript paket dari repositori yang sama.
- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.
- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub Automated IAM Access Analyzer Role Policy Generator.

Epik

Bersiaplah untuk penyebaran

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repo.	Perintah berikut mengkloning repositori Automated IAM Access Analyzer Role Policy Generator ()GitHub. `git clone https://github.com/aws-samples/automated-iam-access-analyzer.git`	Pengembang aplikasi
Instal Lerna.	Perintah berikut menginstal Lerna. `npm i -g lerna`	Pengembang aplikasi

Siapkan dependensi.	Perintah berikut menginstal dependensi untuk repositori.
	`cd automated-iam-access-analyzer/`
	`npm install && npm run bootstrap`
		Pengembang aplikasi

Bangun kodenya.	Perintah berikut menguji, membangun, dan menyiapkan paket zip dari fungsi Lambda.
	`npm run test:code`
	`npm run build:code`
	`npm run pack:code`
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membangun konstruksi.	Perintah berikut membangun infrastruktur mensintesis aplikasi, untuk keduanya AWS CDK dan CDKTF. `npm run build:infra`	
Konfigurasikan izin khusus apa pun.	Di folder repo repositori kloning, edit file allow.json dan deny.json untuk menentuka n izin khusus apa pun untuk peran tersebut. Jika file allow.json dan deny.json berisi izin yang sama, izin penolakan diterapkan.	Administrator AWS, Pengembang aplikasi

Opsi 1 - Menyebarkan solusi menggunakan AWS CDK

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan AWS CDK tumpukan.	Perintah berikut menyebark an infrastruktur melalui AWS CloudFormation. Tentukan parameter berikut: • `<NAME_OF_ROLE>` — ARN dari peran IAM di mana Anda membuat kebijakan baru. • `<TRAIL_ARN>` — ARN dari CloudTrail jejak tempat aktivitas peran disimpan. • `<CRON_EXPRESSION_T O_RUN_SOLUTION>` —	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Ekspresi Cron yang mendefinisikan jadwal regenerasi untuk kebijakan tersebut. Alur kerja Step Functions berjalan pada jadwal ini.
	• <TRAIL_LOOKBACK> — Periode, dalam beberapa hari, untuk melihat kembali jejak saat mengevaluasi izin peran.
	
	`cd infra/cdk`
	`cdk deploy –-parameters roleArn=<NAME_OF_ROLE> \`
	`–-parameters trailArn=<TRAIL_ARN> \`
	`--parameters schedule=<CRON_EXPRESSION_TO_RUN_SOLUTION> \`
	`[ --parameters trailLookBack=<TRAIL_LOOKBACK> ]`
	
	Note
	Tanda kurung siku menunjukkan parameter opsional.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Tunggu kebijakan baru.	Jika jejak tidak berisi jumlah aktivitas historis yang wajar untuk peran tersebut, tunggu hingga Anda yakin bahwa ada cukup aktivitas yang dicatat untuk IAM Access Analyzer untuk menghasilkan kebijakan yang akurat. Jika peran telah aktif di akun untuk jangka waktu yang cukup, masa tunggu ini mungkin tidak diperlukan.	Administrator AWS
Tinjau kebijakan yang dihasilkan secara manual.	Di CodeCommit repositor i Anda, tinjau file.json <ROLE_ARN>yang dihasilka n untuk mengonfirmasi bahwa izin izinkan dan tolak sesuai untuk peran tersebut.	Administrator AWS

Opsi 2 - Menyebarkan solusi menggunakan CDKTF

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sintesis template Terraform.	Perintah berikut mensintesis template Terraform. `lerna exec cdktf synth --scope @aiaa/tfm`	Pengembang aplikasi
Terapkan template Terraform.	Perintah berikut menavigasi ke direktori yang berisi infrastru ktur yang ditentukan CDKTF.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`cd infra/cdktf`
	
	Perintah berikut menyebarkan infrastruktur di target Akun AWS. Tentukan parameter berikut:
	
	• `<account_ID>` — ID dari akun target.
	• `<region>`- Targetnya Wilayah AWS.
	• `<selected_role_ARN>` — ARN dari peran IAM di mana Anda membuat kebijakan baru.
	• `<trail_ARN>` — ARN dari CloudTrail jejak tempat aktivitas peran disimpan.
	• `<schedule_expression>` — Ekspresi Cron yang mendefinisikan jadwal regenerasi untuk kebijakan tersebut. Alur kerja Step Functions berjalan pada jadwal ini.
	• `<trail_look_back>` — Periode, dalam beberapa hari, untuk melihat kembali jejak saat mengevaluasi izin peran.
	
	`TF_VAR_accountId=< account_ID> \`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>TF_VAR_region=<reg
	ion> \
	TF_VAR_roleArns=<s
	elected_role_ARN> \
	TF_VAR_trailArn=<t
	rail_ARN> \
	TF_VAR_schedule=<s
	chedule_expression> \
	[TF_VAR_trailLookBa
	ck=<trail_look_bac
	k>] \ cdktf deploy</pre>
	Note
	Tanda kurung siku menunjukkan parameter opsional.	

(Opsional) Tunggu kebijakan baru.	Jika jejak tidak berisi jumlah aktivitas historis yang wajar untuk peran tersebut, tunggu hingga Anda yakin bahwa ada cukup aktivitas yang dicatat untuk IAM Access Analyzer untuk menghasilkan kebijakan yang akurat. Jika peran telah aktif di akun untuk jangka waktu yang cukup, masa tunggu ini mungkin tidak diperlukan.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tinjau kebijakan yang dihasilkan secara manual.	Di CodeCommit repositor i Anda, tinjau file.json <ROLE_ARN>yang dihasilka n untuk mengonfirmasi bahwa izin izinkan dan tolak sesuai untuk peran tersebut.	Administrator AWS

Sumber daya terkait

AWS sumber daya

- Titik akhir dan kuota IAM Access Analyzer
- Mengkonfigurasi AWS CLI
- Memulai dengan AWS CDK
- Izin hak istimewa paling sedikit

Sumber daya lainnya

- CDK untuk Terraform (situs web Terraform)

Aktifkan Amazon GuardDuty secara kondisional dengan menggunakan templat AWS CloudFormation

Ram Kandaswamy, Amazon Web Services

Ringkasan

AWS CloudFormation, alat infrastruktur sebagai kode (IAc), membantu Anda mengelola AWS sumber daya melalui penerapan berbasis template. CloudFormation Biasanya digunakan untuk mengelola AWS sumber daya. Menggunakannya untuk mengaktifkan Layanan AWS, seperti Amazon GuardDuty, dapat menghadirkan tantangan unik. GuardDuty adalah layanan deteksi ancaman yang terus memantau aktivitas berbahaya dan perilaku tidak sah Anda Akun AWS . Tidak seperti sumber daya tipikal yang dapat dibuat beberapa kali, GuardDuty adalah layanan yang perlu diaktifkan sekali per akun dan Wilayah AWS. CloudFormation Kondisi tradisional hanya mendukung perbandingan nilai statis, yang membuatnya sulit untuk memeriksa keadaan layanan saat ini seperti. GuardDuty Jika Anda mencoba mengaktifkan GuardDuty melalui CloudFormation akun yang sudah aktif, penerapan tumpukan gagal. Ini dapat menciptakan tantangan operasional bagi DevOps tim yang mengelola lingkungan multi-akun.

Pola ini memperkenalkan solusi untuk tantangan ini. Ini menggunakan sumber daya CloudFormation khusus yang didukung oleh AWS Lambdafungsi untuk melakukan pemeriksaan status dinamis. Logika kondisional GuardDuty hanya memungkinkan jika belum diaktifkan. Ini menggunakan output tumpukan untuk merekam GuardDuty status untuk referensi future.

Dengan mengikuti pola ini, Anda dapat mengotomatiskan GuardDuty penerapan di seluruh AWS infrastruktur sambil mempertahankan operasi tumpukan yang bersih dan dapat diprediksi CloudFormation . Pendekatan ini sangat berharga bagi organisasi yang:

- Mengelola beberapa Akun AWS melalui IAc
- Menerapkan layanan keamanan dalam skala besar
- Membutuhkan penyebaran infrastruktur idempoten
- Mengotomatiskan penyebaran layanan keamanan

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Peran AWS Identity and Access Management (IAM) yang memiliki izin untuk membuat, memperbarui, dan menghapus tumpukan CloudFormation
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi

Batasan

Jika GuardDuty telah dinonaktifkan secara manual untuk Akun AWS atau Wilayah AWS, pola ini tidak diaktifkan GuardDuty untuk akun target atau Wilayah tersebut.

Arsitektur

Tumpukan teknologi target

Pola ini digunakan CloudFormation untuk infrastruktur sebagai kode (IAc). Anda menggunakan sumber daya CloudFormation khusus yang didukung oleh fungsi Lambda untuk mencapai kemampuan pemberdayaan layanan dinamis.

Arsitektur target

Diagram arsitektur tingkat tinggi berikut menunjukkan proses pengaktifan GuardDuty dengan menggunakan templat: CloudFormation

1. Anda menyebarkan CloudFormation template untuk membuat CloudFormation tumpukan.
2. Tumpukan menciptakan peran IAM dan fungsi Lambda.
3. Fungsi Lambda mengasumsikan peran IAM.
4. Jika belum GuardDuty diaktifkan pada target Akun AWS, fungsi Lambda mengaktifkannya.

Otomatisasi dan skala

Anda dapat menggunakan AWS CloudFormation StackSet fitur ini untuk memperluas solusi ini ke beberapa Akun AWS dan Wilayah AWS. Untuk informasi selengkapnya, lihat Bekerja dengan AWS CloudFormation StackSets dalam CloudFormation dokumentasi.

Alat

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi Layanan AWS melalui perintah di shell baris perintah Anda.

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- Amazon GuardDuty adalah layanan pemantauan keamanan berkelanjutan yang menganalisis dan memproses log untuk mengidentifikasi aktivitas tak terduga dan berpotensi tidak sah di lingkungan Anda AWS .
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke AWS sumber daya dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Epik

Buat CloudFormation template dan gunakan tumpukan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Simpan kode di Amazon S3.	1. Salin kode Python di bagian Informasi tambahan dari pola ini. 2. Tempel kode di editor teks. 3. Simpan file sebagai `index.py`. 4. Unggah file ke bucket Amazon Simple Storage Service (Amazon S3). Untuk petunjuk, lihat Mengunggah objek dalam dokumentasi Amazon S3.	AWS DevOps
Buat CloudFormation template.	1. Buka konsol CloudFormation .	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di panel navigasi kiri, pilih Infrastructure Composer. 3. Jika Anda tidak melihat kanvas kosong, buat proyek baru. 4. Seret dan jatuhkan AWS Lambda fungsi ke kanvas. 5. Dalam tampilan template, validasi bahwa fungsi Lambda dan grup log hadir. 6. `Runtime`Ubah fungsi Lambda ke versi terbaru Python. 7. Validasi bahwa `Handler` properti memiliki nilai `index.lambda_handler.` 8. Konfigurasikan `CodeUri` properti menjadi lokasi Amazon S3 tempat Anda mengunggah kode Python. Contoh nilai adalah `s3://amzn-s3-demo-bucket/key-name` . 9. Tambahkan `Policies` properti untuk sumber daya. Ikuti praktik terbaik keamanan untuk memberikan izin akses hak istimewa paling sedikit yang memungkinkan CloudFormation dan GuardDuty tindakan. Misalnya, Anda	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dapat menggunakan kebijakan `AWSLambda Execute` terkelola untuk log dan kustom `iam:CreateServiceLinkedRole` untuk GuardDuty. 10. Tambahkan definisi sumber daya khusus ke cuplikan dengan menavigasi ke tampilan templat. `CheckResourceExist: Type: 'Custom:: LambdaCustomResource' Properties: ServiceToken: !GetAtt Function.Arn` 11. Simpan template sebagai `sample.yaml` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat CloudFormation tumpukan.	1. Di AWS CLI, masukkan perintah berikut. Ini membuat CloudFormation tumpukan baru menggunakan `sample.yaml` file. Untuk informasi selengkapnya, lihat Membuat tumpukan dalam CloudFormation dokumentasi.
	`aws cloudformation create-stack \ --stack-name guardduty-cf-stack \ --template-body file://sample.yaml`
	2. Konfirmasikan nilai berikut muncul di AWS CLI, menunjukkan bahwa tumpukan telah berhasil dibuat. Jumlah waktu yang dibutuhkan untuk membuat tumpukan dapat bervariasi.
	`"StackStatus": "CREATE_COMPLETE",`
		AWS DevOps

Validasi yang GuardDuty diaktifkan untuk file. Akun AWS	1. Masuk ke Konsol Manajemen AWS dan buka GuardDuty konsol. 2. Verifikasi bahwa GuardDuty layanan diaktifkan.	Administrator cloud, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan akun atau Wilayah tambahan.	Sesuai kebutuhan untuk kasus penggunaan Anda, gunakan CloudFormation StackSet fitur ini untuk memperluas solusi ini ke beberapa Akun AWS dan Wilayah AWS. Untuk informasi selengkapnya, lihat Bekerja dengan AWS CloudFormation StackSets dalam CloudFormation dokumentasi.	Administrator cloud, administrator AWS

Sumber daya terkait

Referensi

- AWS CloudFormation dokumentasi
- AWS Lambda referensi jenis sumber daya
- CloudFormation jenis sumber daya: AWS: :IAM: :Peran
- CloudFormation jenis sumber daya: AWS:::GuardDuty: Detektor
- Empat cara untuk mengambil properti AWS layanan apa pun menggunakan AWS CloudFormation (posting blog)

Tutorial dan video

- Sederhanakan Manajemen Infrastruktur Anda Menggunakan AWS CloudFormation (Tutorial)
- Gunakan Amazon GuardDuty dan AWS Security Hub CSPM untuk mengamankan beberapa akun (AWS re:Invent 2020)
- Praktik terbaik untuk penulisan AWS CloudFormation (AWS re:Invent 2019)
- Deteksi Ancaman di AWS: Pengantar Amazon GuardDuty (reAWS : Inforce 2019)

Informasi tambahan

Kode Python

```
import boto3
import os
import json
from botocore.exceptions import ClientError
import cfnresponse
guardduty=boto3.client('guardduty')
cfn=boto3.client('cloudformation')
def lambda_handler(event, context):
    print('Event: ', event)
    if 'RequestType' in event:
      if event['RequestType'] in ["Create","Update"]:
          enabled=False
          try:
            response=guardduty.list_detectors()
            if "DetectorIds" in response and len(response["DetectorIds"])>0:
              enabled="AlreadyEnabled"
            elif "DetectorIds" in response and len(response["DetectorIds"])==0:
              cfn_response=cfn.create_stack(
                StackName='guardduty-cfn-stack',
                TemplateBody='{ "AWSTemplateFormatVersion": "2010-09-09",
   "Description": "Guard duty creation template",    "Resources":
 { "IRWorkshopGuardDutyDetector": {  "Type": "AWS::GuardDuty::Detector",
 "Properties": {   "Enable": true  }   } } }'
                )
              enabled="True"
          except Exception as e:
              print("Exception: ",e)
          responseData = {}
          responseData['status'] = enabled
          cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
 "CustomResourcePhysicalID" )
      elif event['RequestType'] == "Delete":
          cfn_response=cfn.delete_stack(
                  StackName='guardduty-cfn-stack')
          cfnresponse.send(event, context, cfnresponse.SUCCESS, {})
```

Aktifkan enkripsi data transparan di Amazon RDS for SQL Server

Ranga Cherukuri, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerapkan enkripsi data transparan (TDE) di Amazon Relational Database Service (Amazon RDS) untuk SQL Server untuk mengenkripsi data saat istirahat.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Instans Amazon RDS for SQL Server DB

Versi produk

Amazon RDS saat ini mendukung TDE untuk versi dan edisi SQL Server berikut:

- SQL Server 2016 Enterprise Edition
- SQL Server 2017 Enterprise Edition
- SQL Server 2019 Standard dan Enterprise Edition
- SQL Server 2022 Standard dan Enterprise Edition

Untuk informasi terbaru tentang versi dan edisi yang didukung, lihat Support for Transparent Data Encryption di SQL Server dalam dokumentasi Amazon RDS.

Arsitektur

Tumpukan teknologi

- Amazon RDS for SQL Server

Arsitektur

Alat

- Microsoft SQL Server Management Studio (SSMS) adalah lingkungan terintegrasi untuk mengelola infrastruktur SQL Server. Ini menyediakan antarmuka pengguna dan sekelompok alat dengan editor skrip kaya yang berinteraksi dengan SQL Server.

Epik

Buat grup opsi di konsol Amazon RDS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buka konsol Amazon RDS.	Masuk ke AWS Management Console dan buka konsol Amazon RDS.	Pengembang, DBA
Buat grup opsi.	Di panel navigasi, pilih Grup opsi, Buat grup. Pilih sqlserver-EE sebagai mesin DB, lalu pilih versi mesin.	Pengembang, DBA
Tambahkan opsi TRANSPARENT_DATA_ENCRYPTION.	Edit grup opsi yang Anda buat dan tambahkan opsi yang disebut`TRANSPARENT_DATA_ENCRYPTION` .	Pengembang, DBA

Kaitkan grup opsi dengan instans DB

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pilih instans DB.	Di konsol Amazon RDS, di panel navigasi, pilih Database, lalu pilih instans DB yang ingin Anda kaitkan dengan grup opsi.	Pengembang, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kaitkan instans DB dengan grup opsi.	Pilih Ubah, lalu gunakan pengaturan grup Opsi untuk mengaitkan instance SQL Server DB dengan grup opsi yang Anda buat sebelumnya.	Pengembang, DBA
Terapkan perubahan.	Terapkan perubahan segera atau selama jendela pemeliharaan berikutnya, seperti yang diinginkan.	Pengembang, DBA

Dapatkan nama sertifikat.	Dapatkan nama sertifikat default dengan menggunakan kueri berikut.
	`USE [master]`
	`GO`
	`SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'`
	`GO`
		Pengembang, DBA

Buat kunci enkripsi basis data

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke instans Amazon RDS for SQL Server DB menggunakan SSMS.	Untuk petunjuk, lihat Menggunakan SSMS di dokumentasi Microsoft.	Pengembang, DBA
Buat kunci enkripsi database dengan menggunakan sertifikat default.	Buat kunci enkripsi database dengan menggunakan nama sertifikat default yang Anda dapatkan sebelumnya. Gunakan query T-SQL	Pengembang, DBA

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	berikut untuk membuat kunci enkripsi database. Anda dapat menentukan algoritma AES_256 alih-alih AES_128.
	
	`USE [Databasename]`
	`GO`
	`CREATE DATABASE`
	`ENCRYPTION KEY`
	`WITH ALGORITHM = AES_128`
	`ENCRYPTION BY SERVER`
	`CERTIFICATE [certificatename]`
	`GO`	

Aktifkan enkripsi pada database.	Gunakan kueri T-SQL berikut untuk mengaktifkan enkripsi database.
	
	`ALTER DATABASE [Database Name]`
	`SET ENCRYPTION ON`
	`GO`
		Pengembang, DBA

Periksa status enkripsi.	Gunakan kueri T-SQL berikut untuk memeriksa status enkripsi.
	
	`SELECT DB_NAME(database_id) AS DatabaseName,`
	`encryption_state,`
	`percent_complete FROM sys.dm_database_encryption_keys`
		Pengembang, DBA

Sumber daya terkait

- Support untuk Enkripsi Data Transparan di SQL Server (dokumentasi Amazon RDS)
- Bekerja dengan Grup Opsi (dokumentasi Amazon RDS)
- Memodifikasi Instans Amazon RDS DB (dokumentasi Amazon RDS)
- Enkripsi Data Transparan untuk SQL Server (dokumentasi Microsoft)
- Menggunakan SSMS (dokumentasi Microsoft)

Memantau dan memulihkan penghapusan kunci AWS KMS yang dijadwalkan

Mikesh Khanal dan Ramya Pulipaka, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS) Cloud, menghapus kunci AWS Key Management Services (AWS KMS) dapat mengakibatkan hilangnya data. Penghapusan menghapus materi kunci dan semua metadata yang terkait dengan kunci AWS KMS, dan itu tidak dapat diubah. Setelah kunci AWS KMS dihapus, Anda tidak dapat lagi mendekripsi data yang dienkripsi di bawah kunci AWS KMS tersebut, sehingga data tidak dapat dipulihkan.

Pola ini mengatur pemantauan, dengan pemberitahuan saat aplikasi atau pengguna menjadwalkan kunci AWS KMS untuk dihapus. Jika Anda menerima pemberitahuan, Anda mungkin ingin membatalkan penghapusan kunci AWS KMS dan mempertimbangkan kembali keputusan Anda untuk menghapusnya. Pola ini menggunakan runbook otomatisasi AWS Systems Manager AWSConfigRemediation-CancelKeyDeletionuntuk memfasilitasi pembatalan penghapusan kunci AWS KMS.

> **Note**
>
> CloudFormation Template pola harus diterapkan di semua Wilayah AWS tempat Anda ingin memantau penghapusan kunci AWS KMS.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- Memahami layanan AWS berikut:
  - Amazon EventBridge
  - AWS KMS
  - Amazon Simple Notification Service (Amazon SNS)
  - AWS Systems Manager

Batasan

- Setiap penyesuaian solusi memerlukan pengetahuan tentang CloudFormation template AWS dan layanan AWS yang digunakan dalam pola ini.
- Saat ini, solusi ini menggunakan bus acara default, dan dapat disesuaikan sesuai dengan persyaratan. Untuk informasi selengkapnya tentang bus acara khusus, lihat dokumentasi AWS.

Arsitektur

Tumpukan teknologi target

- Amazon EventBridge
- AWS KMS
- Amazon SNS
- AWS Systems Manager
- Otomatisasi menggunakan yang berikut ini:
  - AWS Command Line Interface (AWS CLI) atau AWS SDK
  - AWS CloudFormation tumpukan

Arsitektur target

1. Penghapusan kunci AWS KMS dijadwalkan.
2. Acara penghapusan terjadwal dievaluasi oleh aturan. EventBridge
3. EventBridge Aturan tersebut melibatkan topik Amazon SNS.
4. EventBridge Aturan ini memulai otomatisasi Systems Manager dan runbook.
5. Runbook membatalkan penghapusan.

Otomatisasi dan skala

CloudFormation Tumpukan menyebarkan semua sumber daya dan layanan yang diperlukan agar solusi ini berfungsi. Pola dapat dijalankan secara independen dalam satu akun atau dijalankan menggunakan AWS CloudFormation StackSets untuk beberapa akun independen atau organisasi.

```
aws cloudformation create-stack --stack-name  <stack-name>\
```

```
    --template-body file://<Full-Path-of-file> \
    --parameters ParameterKey=,ParameterValue= \
    --capabilities CAPABILITY_NAMED_IAM
```

Alat

Alat

- AWS CloudFormation — AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan menyiapkan sumber daya Amazon Web Services sehingga Anda dapat menghabiskan lebih sedikit waktu untuk mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi yang berjalan di AWS. Anda dapat menggunakan CloudFormation template untuk membuat tumpukan di akun AWS di Wilayah AWS. Template menjelaskan semua sumber daya AWS yang Anda inginkan, serta menyediakan CloudFormation serta mengonfigurasi sumber daya tersebut untuk Anda.
- AWS CLI — AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) adalah alat open source yang dapat Anda gunakan untuk berinteraksi dengan layanan AWS menggunakan perintah di shell baris perintah Anda.
- Amazon EventBridge — Amazon EventBridge adalah layanan bus acara tanpa server yang menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge mengirimkan aliran data real-time dari aplikasi dan layanan AWS Anda sendiri, dan merutekan data tersebut ke target seperti AWS Lambda. EventBridge menyederhanakan proses membangun arsitektur berbasis peristiwa.
- AWS KMS — AWS Key Management Service (AWS KMS) adalah layanan terkelola untuk membuat dan mengendalikan kunci AWS KMS, kunci enkripsi yang digunakan untuk mengenkripsi data Anda.
- AWS SDKs — Alat AWS disertakan SDKs sehingga Anda dapat mengembangkan dan mengelola aplikasi di AWS dalam bahasa pemrograman pilihan Anda.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan (juga dikenal sebagai produsen dan konsumen). Penerbit berkomunikasi secara asinkron dengan pelanggan dengan mengirim pesan ke topik, yang merupakan titik akses logis dan saluran komunikasi.
- AWS Systems Manager — AWS Systems Manager adalah layanan AWS yang dapat Anda gunakan untuk melihat dan mengontrol infrastruktur di AWS. Menggunakan konsol Systems Manager, Anda dapat mengotomatiskan tugas operasional di seluruh sumber daya AWS Anda. Systems Manager membantu Anda menjaga keamanan dan kepatuhan dengan memindai instans

terkelola Anda dan melaporkan (atau mengambil tindakan korektif) setiap pelanggaran kebijakan yang terdeteksi.

Kode

- `alerting_ct_logs.yaml` CloudFormation Template untuk proyek terlampir.

Epik

Siapkan akun AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal dan konfigurasikan AWS CLI.	Instal AWS CLI versi 2. Kemudian konfigurasikan setelan kredensional keamanan untuk identitas , format output default, dan Wilayah AWS default yang digunakan AWS CLI untuk berinteraksi dengan AWS. Identitas harus memiliki izin yang diperlukan untuk melakukan tugas.	Pengembang, insinyur keamanan

Terapkan templat AWS CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh CloudFormation template.	Unduh lampiran ke jalur lokal di komputer Anda dan ekstrak file `alerting_ct_logs.yaml` template.	Pengembang, insinyur keamanan
Menyebarkan template.	Di jendela terminal tempat profil akun AWS telah	Pengembang, insinyur keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dikonfigurasi, jalankan perintah berikut. `aws cloudformation create-stack --stack-name <stack_name> \ --capabilities <Value> \ --template-body file://<Full_Path> \ --parameters ParameterKey=DestinationEmailAddress,ParameterValue=<Value> \ ParameterKey=SNSTopicName,ParameterValue=<Value> \ ParameterKey=EnableRemediation ,ParameterValue=<Value> \ ParameterKey=AutomationAssumeRole, ParameterValue=<Value>` Pada langkah berikutnya, masukkan nilai untuk parameter template.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi parameter template.	Masukkan nilai yang diperlukan untuk parameter. • `DestinationEmailAddress` — Alamat email untuk menerima peringatan saat kunci AWS KMS dijadwalkan untuk dihapus. • `SNSTopicName` — Nama topik Amazon SNS. • `EnableRemediation` — Pembatalan penghapusan kunci terjadwal menggunakan runbook Systems Manager. Nilai yang diizinkan adalah `true` dan `false`. • `AutomationAssumeRole` — Nama Sumber Daya Amazon (ARN) dari peran yang memungkinkan otomatisasi Systems Manager untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat bagian Izin IAM yang Diperlukan di dokumentasi. AWSConfigRemediation-CancelKeyDeletion • `Capabilities` — CloudFormation Agar AWS dapat membuat tumpukan, Anda harus secara eksplisit	Pengembang, insinyur keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengakui bahwa template tumpukan Anda berisi kemampuan tertentu.	

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Periksa kotak masuk email Anda dan pilih Konfirmasi langganan di pesan email yang Anda terima dari Amazon SNS. Jendela browser web akan terbuka dan menampilkan konfirmasi berlangganan dan ID langganan Anda.	Pengembang, insinyur keamanan

Sumber daya terkait

Referensi

- Membuat aturan untuk layanan AWS
- Membuat CloudWatch alarm Amazon untuk mendeteksi penggunaan kunci AWS KMS yang sedang menunggu penghapusan

Tutorial dan video

- Cara memulai dengan Amazon EventBridge
- Menyelam jauh di Amazon EventBridge (AWS Online Tech Talks)

Lokakarya AWS

- Bekerja dengan EventBridge aturan

Informasi tambahan

Kode berikut memberikan contoh untuk memperluas solusi untuk memantau dan memberi tahu Anda tentang perubahan apa pun dalam layanan AWS apa pun. Contohnya termasuk pola yang telah ditentukan dan pola kustom. Untuk informasi selengkapnya, lihat Peristiwa dan pola acara di EventBridge.

```
EventPattern:
        source:
        - aws.kms
        detail-type:
        - AWS API Call via CloudTrail
        detail:
          eventSource:
          - kms.amazonaws.com
          eventName:
          - ScheduleKeyDeletion
```

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Identifikasi bucket Amazon S3 publik dengan menggunakan Security Hub AWS Organizations CSPM

Mourad Cherfaoui, Arun Chandapillai, dan Parag Nagwekar, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara membuat mekanisme untuk mengidentifikasi bucket Amazon Simple Storage Service (Amazon S3) publik di akun Anda. AWS Organizations Mekanisme ini bekerja dengan menggunakan kontrol dari standar AWS Foundational Security Best Practices (FSBP) AWS Security Hub CSPM untuk memantau bucket Amazon S3. Anda dapat menggunakan Amazon EventBridge untuk memproses temuan CSPM Security Hub, lalu memposting temuan ini ke topik Amazon Simple Notification Service (Amazon SNS). Pemangku kepentingan di organisasi Anda dapat berlangganan topik dan mendapatkan pemberitahuan email langsung tentang temuan tersebut.

Bucket Amazon S3 baru dan objeknya tidak mengizinkan akses publik secara default. Anda dapat menggunakan pola ini dalam skenario di mana Anda harus mengubah konfigurasi Amazon S3 default berdasarkan persyaratan organisasi Anda. Misalnya, ini bisa menjadi skenario di mana Anda memiliki bucket Amazon S3 yang menghosting situs web atau file yang menghadap publik yang harus dapat dibaca oleh semua orang di internet dari bucket Amazon S3 Anda.

Security Hub CSPM sering digunakan sebagai layanan pusat untuk mengkonsolidasikan semua temuan keamanan, termasuk yang terkait dengan standar keamanan dan persyaratan kepatuhan. Ada Layanan AWS hal lain yang dapat Anda gunakan untuk mendeteksi bucket Amazon S3 publik, tetapi pola ini menggunakan penerapan CSPM Security Hub yang ada dengan konfigurasi minimal.

Prasyarat dan batasan

Prasyarat

- Penyiapan AWS multi-akun dengan akun administrator CSPM Security Hub khusus
- Security Hub CSPM dan AWS Config, diaktifkan di Wilayah AWS yang ingin Anda pantau

> **Note**
>
> Anda harus mengaktifkan agregasi Lintas wilayah di CSPM Security Hub jika Anda ingin memantau beberapa Wilayah dari satu Wilayah agregasi.

- Izin pengguna untuk mengakses dan memperbarui akun administrator CSPM Security Hub, membaca akses ke semua bucket Amazon S3 di organisasi, dan izin untuk mematikan akses publik (jika diperlukan)

Arsitektur

Diagram berikut menunjukkan arsitektur untuk menggunakan Security Hub CSPM untuk mengidentifikasi bucket Amazon S3 publik.

Diagram menunjukkan alur kerja berikut:

1. Security Hub CSPM memantau konfigurasi bucket Amazon S3 di AWS Organizations semua akun (termasuk akun administrator) dengan menggunakan kontrol S3.2 dan S3.3 dari standar keamanan FSBP, dan mendeteksi temuan jika bucket dikonfigurasi sebagai publik.
2. Akun administrator CSPM Security Hub mengakses temuan (termasuk untuk S3.2 dan S3.3) dari semua akun anggota.
3. Security Hub CSPM secara otomatis mengirimkan semua temuan baru dan semua pembaruan temuan yang ada ke Temuan CSPM EventBridge Security Hub - Acara yang diimpor. Ini termasuk peristiwa untuk temuan dari akun administrator dan anggota.
4. EventBridge Aturan menyaring temuan dari S3.2 dan S3.3 yang memiliki`FAILED`, `ComplianceStatus` status alur kerja`NEW`, dan a dari. `RecordState` `ACTIVE`
5. Aturan menggunakan pola peristiwa untuk mengidentifikasi peristiwa dan mengirimkannya ke topik Amazon SNS setelah dicocokkan.
6. Topik Amazon SNS mengirimkan acara ke pelanggannya (melalui email, misalnya).
7. Analis keamanan yang ditunjuk untuk menerima pemberitahuan email meninjau ember Amazon S3 yang dimaksud.
8. Jika bucket disetujui untuk akses publik, analis keamanan menetapkan status alur kerja dari temuan terkait di Security Hub CSPM ke. `SUPPRESSED` Jika tidak, analis menetapkan statusnya`NOTIFIED`. Ini menghilangkan notifikasi future untuk bucket Amazon S3 dan mengurangi noise notifikasi.
9. Jika status alur kerja diatur`NOTIFIED`, analis keamanan meninjau temuan tersebut dengan pemilik bucket untuk menentukan apakah akses publik dibenarkan dan mematuhi persyaratan privasi dan perlindungan data. Investigasi menghasilkan penghapusan akses publik ke ember atau

menyetujui akses publik. Dalam kasus terakhir, analis keamanan menetapkan status alur kerja ke`SUPPRESSED`.

> **Note**
>
> Diagram arsitektur berlaku untuk penerapan agregasi Wilayah dan Lintas wilayah tunggal. Dalam akun A, B, dan C dalam diagram, Security Hub CSPM dapat menjadi milik Wilayah yang sama dengan akun administrator atau milik Wilayah yang berbeda jika agregasi lintas wilayah diaktifkan.

Alat

- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. EventBridge memberikan aliran data real-time dari aplikasi Anda sendiri, perangkat lunak sebagai layanan (SaaS) aplikasi, dan. Layanan AWS EventBridge merutekan data tersebut ke target seperti topik dan AWS Lambda fungsi Amazon SNS jika data cocok dengan aturan yang ditentukan pengguna.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email. Pelanggan menerima semua pesan yang dipublikasikan ke topik tempat mereka berlangganan, dan semua pelanggan topik menerima pesan yang sama.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Security Hub CSPMmemberikan pandangan komprehensif tentang keadaan keamanan Anda di AWS. Security Hub CSPM juga membantu Anda memeriksa AWS lingkungan Anda terhadap standar industri keamanan dan praktik terbaik. Security Hub CSPM mengumpulkan data keamanan dari seluruh Akun AWS, layanan, dan produk mitra pihak ketiga yang didukung, dan kemudian membantu menganalisis tren keamanan dan mengidentifikasi masalah keamanan prioritas tertinggi.

Epik

Konfigurasikan akun CSPM Security Hub

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan CSPM Security Hub di AWS Organizations akun.	Untuk mengaktifkan CSPM Security Hub di akun organisasi tempat Anda ingin memantau bucket Amazon S3, lihat panduan dari Menetapkan akun administrator CSPM Security Hub (konsol) dan Mengelola akun anggota yang dimiliki organisasi dalam dokumentasi CSPM Security Hub.	Administrator AWS
(Opsional) Aktifkan agregasi lintas wilayah.	Jika Anda ingin memantau bucket Amazon S3 di beberapa Wilayah dari satu Wilayah, siapkan agregasi Lintas wilayah.	Administrator AWS
Aktifkan kontrol S3.2 dan S3.3 untuk standar keamanan FSBP.	Anda harus mengaktifkan kontrol S3.2 dan S3.3 untuk standar keamanan FSBP. 1. Untuk mengaktifkan kontrol S3.2, ikuti instruksi dari [S3.2] bucket S3 yang harus melarang akses baca publik dalam dokumentasi CSPM Security Hub. 2. Untuk mengaktifkan kontrol S3.3, ikuti instruksi dari [3] Bucket S3 harus melarang akses tulis publik	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dalam dokumentasi CSPM Security Hub.	

Mengatur lingkungan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan topik Amazon SNS dan langganan email.	1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon SNS. 2. Pada panel navigasi, silakan pilih Topik, lalu pilih Buat topik. 3. Untuk Tipe, pilih Standar. 4. Untuk Nama, masukkan nama untuk topik Anda (misalnya, public-s3-bucket). 5. Pilih Buat topik. 6. Pada tab Langganan untuk topik Anda, pilih Buat langganan. 7. Untuk Protokol, pilih Email. 8. Untuk Endpoint, masukkan alamat email yang akan menerima notifikasi. Anda dapat menggunakan alamat email AWS administr ator, profesional TI, atau profesional Infosec. 9. Pilih Buat langganan. Untuk membuat langganan email	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tambahan, ulangi langkah 6-8 sesuai kebutuhan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan EventBridge aturan.	1. Buka konsol EventBridge . 2. Di bagian Memulai, pilih EventBridge Aturan, lalu pilih Buat aturan. 3. Pada halaman Tentukan detail aturan, untuk Nama, masukkan nama untuk aturan Anda (misalnya , public-s3-bucket). Pilih Berikutnya. 4. Di bagian Pola acara, pilih Edit pola. 5. Salin kode berikut, tempel ke editor kode pola acara, lalu pilih Berikutnya. `{ "source": ["aws.securityhub"], "detail-type": ["Security Hub Findings - Imported"], "detail": { "findings": { "Compliance": { "Status": ["FAILED"] }, "RecordState": ["ACTIVE"], "Workflow": { "Status": ["NEW"] }, "ProductFields": {`	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"ControlId": ["S3.2", "S3.3"] } } } }</pre> 6. Pada halaman Pilih target, untuk Pilih target, pilih topik SNS sebagai target, lalu pilih topik yang Anda buat sebelumnya. 7. Pilih Berikutnya, pilih Berikutnya lagi, lalu pilih Buat aturan.	

Pemecahan masalah

Isu	Solusi
Saya memiliki bucket Amazon S3 dengan akses publik diaktifkan, tetapi saya tidak mendapatkan pemberitahuan email untuk itu.	Ini bisa jadi karena bucket dibuat di Wilayah lain dan agregasi Lintas wilayah tidak diaktifkan di akun administrator CSPM Security Hub. Untuk mengatasi masalah ini, aktifkan agregasi lintas wilayah atau terapkan solusi pola ini di Wilayah tempat bucket Amazon S3 Anda saat ini berada.

Sumber daya terkait

- Apa itu AWS Security Hub CSPM? (Dokumentasi CSPM Security Hub)
- AWS Standar Praktik Terbaik Keamanan Dasar (FSBP) (dokumentasi CSPM Security Hub)
- AWS Security Hub CSPM skrip aktifkan multi-akun (Labs)AWS
- Praktik terbaik keamanan untuk Amazon S3 (dokumentasi Amazon S3)

Informasi tambahan

Alur kerja untuk memantau ember Amazon S3 publik

Alur kerja berikut menggambarkan bagaimana Anda dapat memantau bucket Amazon S3 publik di organisasi Anda. Alur kerja mengasumsikan bahwa Anda menyelesaikan langkah-langkah dalam Konfigurasi topik Amazon SNS dan kisah langganan email dari pola ini.

1. Anda menerima pemberitahuan email saat bucket Amazon S3 dikonfigurasi dengan akses publik.
   - Jika bucket disetujui untuk akses publik, atur status alur kerja dari temuan terkait ke `SUPPRESSED` akun administrator CSPM Security Hub. Hal ini mencegah CSPM Security Hub mengeluarkan notifikasi lebih lanjut untuk bucket ini dan dapat menghilangkan peringatan duplikat.
   - Jika bucket tidak disetujui untuk akses publik, setel status alur kerja dari temuan terkait di akun administrator CSPM Security Hub. `NOTIFIED` Hal ini mencegah CSPM Security Hub mengeluarkan notifikasi lebih lanjut untuk bucket ini dari Security Hub CSPM dan dapat menghilangkan noise.
2. Jika bucket mungkin berisi data sensitif, matikan akses publik segera sampai peninjauan selesai. Jika Anda menonaktifkan akses publik, maka Security Hub CSPM mengubah status alur kerja menjadi. `RESOLVED` Kemudian, email notifikasi untuk bucket stop.
3. Temukan pengguna yang mengonfigurasi bucket sebagai publik (misalnya, dengan menggunakan AWS CloudTrail) dan mulai peninjauan. Hasil tinjauan dalam menghapus akses publik ke bucket atau menyetujui akses publik. Jika akses publik disetujui, maka atur status alur kerja dari temuan yang sesuai ke`SUPPRESSED`.

Menyerap dan menganalisis log keamanan AWS di Microsoft Sentinel

Ivan Girardi dan Sebastian Wenzel, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara mengotomatiskan konsumsi log AWS keamanan, seperti log, data Amazon Logs AWS CloudTrail , data Amazon VPC Flow CloudWatch Logs, dan temuan GuardDuty Amazon, ke Microsoft Sentinel. Jika organisasi Anda menggunakan Microsoft Sentinel sebagai sistem informasi keamanan dan manajemen peristiwa (SIEM), ini membantu Anda memantau dan menganalisis log secara terpusat untuk mendeteksi peristiwa terkait keamanan. Segera setelah log tersedia, log tersebut secara otomatis dikirim ke bucket Amazon Simple Storage Service (Amazon S3) dalam waktu kurang dari 5 menit. Ini dapat membantu Anda mendeteksi peristiwa keamanan di AWS lingkungan Anda dengan cepat.

Microsoft Sentinel menyerap CloudTrail log dalam format tabel yang menyertakan stempel waktu asli saat acara direkam. Struktur log tertelan memungkinkan kemampuan kueri dengan menggunakan Kusto Query Language di Microsoft Sentinel.

Pola ini menerapkan solusi pemantauan dan peringatan yang mendeteksi kegagalan konsumsi dalam waktu kurang dari 1 menit. Ini juga termasuk sistem notifikasi yang dapat dipantau oleh SIEM eksternal. Anda gunakan AWS CloudFormation untuk menyebarkan sumber daya yang diperlukan di akun logging.

Target audiens

Pola ini direkomendasikan untuk pengguna yang memiliki pengalaman dengan AWS Control Tower,, AWS Organizations CloudFormation, AWS Identity and Access Management (IAM), dan AWS Key Management Service (AWS KMS).

Prasyarat dan batasan

Prasyarat

Berikut ini adalah prasyarat untuk menerapkan solusi ini:

- Aktif Akun AWS yang dikelola sebagai organisasi di AWS Organizations dan merupakan bagian dari AWS Control Tower landing zone. Organisasi harus menyertakan akun khusus

untuk pencatatan. Untuk petunjuk, lihat Membuat dan mengonfigurasi organisasi dalam AWS Organizations dokumentasi.

- CloudTrail Jejak yang mencatat peristiwa untuk seluruh organisasi dan menyimpan log di bucket Amazon S3 di akun logging. Untuk petunjuk, lihat Membuat jejak untuk organisasi.
- Di akun logging, izin untuk mengambil peran IAM yang ada yang memiliki izin berikut:
  - Menyebarkan sumber daya yang ditentukan dalam CloudFormation template yang disediakan.
  - Menyebarkan CloudFormation template yang disediakan.
  - Ubah kebijakan AWS KMS kunci jika log dienkripsi dengan kunci yang dikelola pelanggan.
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi.
- Akun Microsoft Azure dengan langganan untuk menggunakan Microsoft Sentinel.
- Aktifkan dan atur Microsoft Sentinel. Untuk petunjuk, lihat Mengaktifkan Microsoft Sentinel serta fitur dan konten awal dalam dokumentasi Microsoft Sentinel.
- Memenuhi prasyarat untuk mengatur konektor Microsoft Sentinel S3.

Batasan

- Solusi ini meneruskan log keamanan dari bucket Amazon S3 di akun logging ke Microsoft Sentinel. Petunjuk tentang cara mengirim log ke Amazon S3 tidak disediakan secara eksplisit.
- Pola ini memberikan instruksi untuk penyebaran di AWS Control Tower landing zone. Namun, penggunaan AWS Control Tower tidak diperlukan.
- Solusi ini kompatibel dengan lingkungan di mana bucket logging Amazon S3 dibatasi dengan kebijakan kontrol layanan (SCPs), seperti Larang Perubahan pada Kebijakan Bucket untuk Bucket Amazon AWS Control Tower S3 yang Dibuat di Arsip Log.
- Pola ini memberikan instruksi untuk meneruskan CloudTrail log, tetapi Anda dapat menyesuaikan solusi ini untuk mengirim log lain yang didukung Microsoft Sentinel, seperti log dari Log, CloudWatch Amazon VPC Flow Logs, dan. GuardDuty
- Instruksi menggunakan AWS CLI untuk menyebarkan CloudFormation template, tetapi Anda juga bisa menggunakan. Konsol Manajemen AWS Untuk petunjuk, lihat Menggunakan AWS CloudFormation konsol. Jika Anda menggunakan konsol untuk menyebarkan tumpukan, terapkan tumpukan yang Wilayah AWS sama dengan bucket logging.
- Solusi ini menerapkan antrian Amazon Simple Queue Service (Amazon SQS) untuk mengirimkan notifikasi Amazon S3. Antrian berisi pesan dengan jalur objek yang diunggah di bucket Amazon S3, bukan data aktual. Antrian menggunakan enkripsi SSE-SQS untuk membantu melindungi konten pesan. Jika Anda ingin mengenkripsi antrian SQS dengan SSE-KMS, Anda dapat menggunakan

kunci KMS yang dikelola pelanggan. Untuk informasi selengkapnya, lihat Enkripsi saat istirahat di Amazon SQS.

Arsitektur

Bagian ini memberikan gambaran tingkat tinggi tentang arsitektur yang dibuat oleh kode sampel. Diagram berikut menunjukkan sumber daya yang digunakan di akun logging untuk menyerap log dari bucket Amazon S3 yang ada ke Microsoft Sentinel.

Diagram arsitektur menunjukkan interaksi sumber daya berikut:

1. Di akun logging, Microsoft Sentinel mengasumsikan peran IAM melalui OpenID Connect (OIDC) untuk mengakses log di bucket Amazon S3 tertentu dan antrian Amazon SQS.
2. Amazon Simple Notification Service (Amazon SNS) dan Amazon S3 digunakan untuk enkripsi. AWS KMS
3. Amazon S3 mengirimkan pesan notifikasi ke antrian Amazon SQS setiap kali menerima log baru.
4. Microsoft Sentinel memeriksa Amazon SQS untuk pesan baru. Antrian Amazon SQS menggunakan enkripsi SSE-SQS. Periode penyimpanan pesan diatur ke 14 hari.
5. Microsoft Sentinel menarik pesan dari antrian Amazon SQS. Pesan berisi jalur objek Amazon S3 yang diunggah. Microsoft Sentinel menyerap benda-benda itu dari ember Amazon S3 ke dalam akun Microsoft Azure.
6. CloudWatch Alarm memonitor antrian Amazon SQS. Jika pesan tidak diterima dan dihapus dari antrian Amazon SQS dalam waktu 5 menit, maka pesan tersebut memulai pemberitahuan Amazon SNS yang mengirim email.

AWS Control Tower membantu Anda mengatur struktur unit organisasi dasar (OU) dan memusatkan CloudTrail log di akun logging. Ini juga mengimplementasikan wajib SCPs untuk melindungi ember penebangan.

Kami telah menyediakan arsitektur target di AWS Control Tower landing zone, tetapi ini tidak sepenuhnya diperlukan. Dalam diagram ini, sumber daya dalam akun manajemen mencerminkan AWS Control Tower penyebaran dan CloudTrail jejak yang mencatat peristiwa untuk seluruh organisasi.

Pola ini berfokus pada penyebaran sumber daya di akun logging. Jika log yang disimpan di Amazon S3 di AWS Control Tower landing zone Anda dienkripsi dengan kunci KMS yang dikelola pelanggan,

maka Anda harus memperbarui kebijakan kunci untuk mengizinkan Microsoft Sentinel mendekripsi log. Di AWS Control Tower landing zone, Anda mengelola kebijakan utama dari akun manajemen, yang merupakan tempat kunci dibuat.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan Wilayah.
- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- AWS Control Towermembantu Anda mengatur dan mengatur lingkungan AWS multi-akun, mengikuti praktik terbaik.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon Simple Queue Service (Amazon Simple Queue Service) menyediakan antrian host yang aman, tahan lama, dan tersedia yang membantu Anda mengintegrasikan dan memisahkan sistem dan komponen perangkat lunak terdistribusi.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat lainnya

- Microsoft Sentinel adalah sistem SIEM cloud-native yang menyediakan orkestrasi keamanan, otomatisasi, dan respons (SOAR).

Repositori kode

Kode untuk pola ini tersedia di GitHub Ingest dan menganalisis log AWS keamanan di repositori Microsoft Sentinel.

Praktik terbaik

- Ikuti prinsip hak istimewa paling sedikit (dokumentasi IAM).
- Ikuti Praktik terbaik untuk AWS Control Tower administrator (AWS Control Tower dokumentasi).
- Ikuti praktik AWS CloudFormation terbaik (CloudFormation dokumentasi).
- Gunakan alat analisis kode, seperti cfn_nag, untuk memindai templat yang dihasilkan. CloudFormation Alat cfn_nag mengidentifikasi potensi masalah keamanan dalam CloudFormation template dengan mencari pola.

Epik

Hubungkan Microsoft Sentinel ke Amazon S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan konektor Microsoft Sentinel S3.	1. Di Microsoft Sentinel, pilih Konektor data. 2. Dari galeri konektor data, pilih Amazon Web Services S3. **Note** Jika Anda tidak melihat konektorn ya, instal solusi Amazon Web Services dari Content Hub di Microsoft Sentinel. 3. Di panel detail untuk konektor, pilih Buka halaman konektor.	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Di bagian Konfigurasi, salin ID Eksternal. Anda memerlukan ID ini nanti.	

Menyebarkan tumpukan CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Dalam shell bash, masukkan perintah berikut. Ini mengkloning Ingest dan menganalisis Log AWS Keamanan di repositori Microsoft Sentinel. `git clone https://github.com/aws-samples/ingest-and-analyze-aws-security-logs-in-microsoft-sentinel.git`	DevOps insinyur, AWS Umum
Asumsikan peran IAM dalam akun logging.	Di akun logging, asumsikan peran IAM yang memiliki izin untuk menyebarkan tumpukan. CloudFormation Untuk informasi selengkapnya tentang mengasumsikan peran IAM dalam AWS CLI, lihat Menggunakan peran IAM dalam. AWS CLI	DevOps insinyur, AWS Umum
Menyebarkan tumpukan.	Untuk menyebarkan CloudFormation tumpukan masukkan perintah berikut, di mana:	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `<Bucket name>`adalah nama bucket Amazon S3 logging. • `<Sentinel external ID>`adalah ID eksternal konektor Amazon S3 di Microsoft Sentinel. • `<Email address>`adalah alamat email yang valid tempat Anda ingin menerima pemberitahuan. • `<Customer managed key ARN>`adalah Nama Sumber Daya Amazon (ARN) dari kunci KMS yang dikelola pelanggan. Berikan parameter ini hanya jika log dienkripsi dengan kunci KMS yang dikelola pelanggan. • `<Suffix>`adalah parameter opsional untuk menghindari konflik nama sumber daya. • `<ARN for the OIDC provider>` adalah ARN dari penyedia OIDC jika sudah ada. Jika Anda tidak memberikan parameter ini, CloudFormation buat penyedia OIDC.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Important Jika AWS organisasi dipantau dengan Microsoft Code Defender, penyedia OIDC untuk Microsoft telah digunakan. Anda harus memberikan parameter ini dan ARN dari penyedia yang ada. `aws cloudformation deploy --stack-name cloudtrail-sentinel-integration \ --no-fail-on-empty-changeset \ --template-file template.yml \ --capabilities CAPABILITY_IAM CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \ --parameter-overrides \ ControlTowerS3BucketName="<Bucket name>" \ AzureWorkspaceID="<Sentinel external ID>" \`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`EmailAddress="<Email address>" \ KMSKeyArn="<Customer managed key ARN>" \ Suffix="<Suffix to avoid name conflicts>" \ OIDCProviderArn="<ARN for the OIDC provider>"`	
Salin output.	Dari output CloudFormation tumpukan, salin nilai untuk `SentinelRoleArn` dan`SentinelSQS` . Anda menggunakan nilai-nilai ini nanti untuk menyelesaikan konfigurasi di Microsoft Sentinel.	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Ubah kebijakan utama.	Jika Anda tidak menggunakan kunci KMS yang dikelola pelanggan untuk mengenkripsi log di bucket Amazon S3, Anda dapat melewati langkah ini. Jika log dienkripsi dengan kunci KMS yang dikelola pelanggan, ubah kebijakan kunci untuk memberikan izin Microsoft Sentinel untuk mendekripsi log. Berikut ini adalah contoh kebijakan kunci. Kebijakan contoh ini memungkinkan akses lintas akun jika kunci KMS ada di akun lain. Akun AWS `{ "Version": "2012-10-17", "Id": "key-policy", "Statement": [ ... { "Sid": "Grant access to decrypt", "Effect": "Allow", "Principal": { "AWS": "<SentinelRoleArn>" }, "Action": "kms:Decrypt",`	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>"Resource": "<KeyArn>" }] }</pre>	

Konfigurasikan konektor di Microsoft Sentinel

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lengkapi konfigurasi di Microsoft Sentinel.	1. Di Microsoft Sentinel, pilih Konektor data. 2. Dari galeri konektor data, pilih Amazon Web Services S3. 3. Di panel detail untuk konektor, pilih Buka halaman konektor. 4. Di bagian Konfigurasi, lakukan hal berikut: a. Di Peran untuk menambahkan, masukkan `SentinelRoleArn` nilai yang Anda salin. b. Di SQS URL, masukkan `SentinelSQS` nilai yang Anda salin. c. Dalam daftar tabel Tujuan, pilih`AWSCloudTrail` . 5. Pilih Tambahkan koneksi.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kirim pemberitahuan acara Amazon S3 ke Amazon SQS.	Ikuti petunjuk dalam Mengaktifkan dan mengonfigurasi pemberitahuan peristiwa menggunakan konsol Amazon S3 untuk mengonfigurasi bucket logging Amazon S3 untuk mengirim pemberitahuan peristiwa ke antrean Amazon SQS. Jika CloudTrail telah dikonfigurasi untuk seluruh organisasi, log di bucket ini memiliki awalan`<OrgID>/AWSLogs/<OrgID>/` , di `<OrgID>` mana ID organisasi. Untuk informasi selengkapnya, lihat Melihat detail tentang organisasi Anda.	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan bahwa log tertelan.	1. Tunggu hingga log tertelan di Microsoft Sentinel. Ini dapat memakan waktu beberapa menit. 2. Di Microsoft Sentinel, buka halaman Konektor Data Amazon S3, lalu lakukan hal berikut: • Konfirmasikan bahwa status Konektor Data Amazon S3 adalah. `Connected` • Periksa volume data dalam grafik Data yang diterima. Untuk informasi selengkapnya tentang memeriksa aktivitas konektor data, lihat Konektor data dalam dokumentasi Microsoft.	DevOps insinyur

Validasi solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bandingkan CloudWatch dan log Sentinel.	Dalam konfigurasi default AWS Control Tower, CloudTrail log dikirim ke Amazon CloudWatch dan disimpan di akun AWS Control Tower manajemen. Untuk informasi selengkapnya, lihat Logging dan monitoring in	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	AWS Control Tower. Gunakan langkah-langkah berikut untuk mengonfirmasi bahwa log secara otomatis tertelan ke Microsoft Sentinel:
	1. Buka konsol CloudWatch .
	2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
	3. Untuk Pilih grup log, pilih grup log tempat CloudTrail log disimpan, seperti `aws-controltower/CloudTrailLogs` .
	4. Di kotak editor kueri, masukkan `fields eventID`.
	5. Pilih Run query (Jalankan kueri).
	6. Pilih Ekspor hasil, lalu pilih Salin tabel ke clipboard (CSV).
	7. Tempel hasilnya ke editor teks.
	8. Ubah format output sehingga dapat digunakan dalam kueri Microsoft Sentinel. Berikut ini adalah contoh yang menggunakan Bahasa Kueri Kusto:
	`AWSCloudTrail`
	`| where AwsEventId in`
	`(`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>'aa08b5fe-3bfb-391a-a14e-5fcebe14dab2', '9decd805-269c-451c-b75b-762f5dce59f9')</pre> 9. Di Microsoft Sentinel, buka halaman Konektor Data Amazon S3. Di samping grafik Data diterima, pilih Pergi ke analisis log. 10. Di kotak editor kueri, masukkan kueri, lalu pilih Jalankan. 11. Di Microsoft Sentinel dan CloudWatch, verifikasi bahwa jumlah entri sama. Sesuaikan rentang waktu jika diperlukan.	

Sumber daya terkait

AWS dokumentasi dan sumber daya

- AWS CLI Referensi Perintah (AWS CLI dokumentasi)
- Konfigurasi opsional AWS KMS keys (AWS Control Tower dokumentasi)
- Enkripsi saat istirahat di Amazon SQS (dokumentasi Amazon SQS)
- Bagaimana cara menjaga penerima milis berhenti berlangganan semua orang di daftar dari email topik Amazon SNS saya? (Pusat AWS Pengetahuan)

Dokumentasi Microsoft

- Hubungkan Microsoft Sentinel ke Amazon Web Services untuk menyerap data log layanan AWS
- Kusto Query Language di Microsoft Sentinel

Mengelola AWS Organizations kebijakan sebagai kode dengan menggunakan AWS CodePipeline dan Amazon Bedrock

Andre Cavalcante dan Mariana Pessoa de Queiroz, Amazon Web Services

Ringkasan

Anda dapat menggunakan kebijakan otorisasi AWS Organizations untuk mengonfigurasi dan mengelola akses untuk prinsipal dan sumber daya secara terpusat di akun anggota Anda. Kebijakan kontrol layanan (SCPs) menentukan izin maksimum yang tersedia untuk peran AWS Identity and Access Management (IAM) dan pengguna di organisasi Anda. Kebijakan kontrol sumber daya (RCPs) menentukan izin maksimum yang tersedia untuk sumber daya di organisasi Anda.

Pola ini membantu Anda mengelola SCPs dan RCPs sebagai infrastruktur sebagai kode (IAc) yang Anda terapkan melalui pipeline integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD). Dengan menggunakan AWS CloudFormation atau Hashicorp Terraform untuk mengelola kebijakan ini, Anda dapat mengurangi beban yang terkait dengan membangun dan memelihara beberapa kebijakan otorisasi.

Pola ini mencakup fitur-fitur berikut:

- Anda membuat, menghapus, dan memperbarui kebijakan otorisasi dengan menggunakan file manifes (`scp-management.json`dan`rcp-management.json`).
- Anda bekerja dengan pagar pembatas alih-alih kebijakan. Anda menentukan pagar pembatas dan targetnya dalam file manifes.
- Pipeline, yang menggunakan AWS CodeBuild dan AWS CodePipeline, menggabungkan dan mengoptimalkan pagar pembatas dalam file manifes. Untuk setiap pernyataan dalam file manifes, pipeline menggabungkan pagar pembatas menjadi satu SCP atau RCP dan kemudian menerapkannya ke target yang ditentukan.
- AWS Organizations menerapkan kebijakan untuk target Anda. Target dapat berupa Akun AWS, unit organisasi (OU), lingkungan (yang merupakan grup akun atau OUs yang Anda tentukan dalam `environments.json` file), atau sekelompok akun yang berbagi AWS tag.
- Amazon Bedrock membaca log pipeline dan merangkum semua perubahan kebijakan.
- Pipa membutuhkan persetujuan manual. Penyetuju dapat meninjau ringkasan eksekutif yang disiapkan Amazon Bedrock, yang membantu mereka memahami perubahan.

Prasyarat dan batasan

Prasyarat

- Beberapa Akun AWS yang dikelola sebagai organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Membuat organisasi.
- Fitur SCP dan RCP diaktifkan di. AWS Organizations Untuk informasi selengkapnya, lihat Mengaktifkan jenis kebijakan.
- Terraform versi 1.9.8 atau yang lebih baru diinstal.
- Jika Anda tidak menerapkan solusi ini melalui pipeline Terraform, file status Terraform harus disimpan dalam bucket Amazon Simple Storage Service (Amazon S3) di Akun AWS tempat Anda menerapkan pipeline manajemen kebijakan.
- Python versi 3.13.3 atau yang lebih baru diinstal.

Batasan

- Anda tidak dapat menggunakan pola ini untuk mengelola SCPs atau RCPs yang dibuat di luar CI/CD pipeline ini. Namun, Anda dapat membuat ulang kebijakan yang ada melalui pipeline. Untuk informasi selengkapnya, lihat Memigrasi kebijakan yang ada ke pipeline di bagian Informasi tambahan pada pola ini.
- Jumlah akun OUs, dan kebijakan di setiap akun tunduk pada kuota dan batas layanan untuk AWS Organizations.
- Pola ini tidak dapat digunakan untuk mengonfigurasi kebijakan manajemen AWS Organizations, seperti kebijakan cadangan, kebijakan tag, kebijakan aplikasi obrolan, atau kebijakan deklaratif.

Arsitektur

Diagram berikut menunjukkan alur kerja pipa manajemen kebijakan dan sumber daya yang terkait.

Diagram menunjukkan alur kerja berikut:

1. Seorang pengguna melakukan perubahan pada file `scp-management.json` atau `rcp-management.json` manifes di cabang utama repositori jarak jauh.
2. Perubahan ke `main` cabang memulai pipa di AWS CodePipeline.

3. CodePipeline memulai `Validate-Plan` CodeBuild proyek. Proyek ini menggunakan skrip Python di repositori jarak jauh untuk memvalidasi kebijakan dan file manifes kebijakan. CodeBuild Proyek ini melakukan hal berikut:
   a. Memeriksa bahwa file manifes SCP dan RCP berisi pernyataan unik IDs ()`Sid`.
   b. Menggunakan skrip `scp-policy-processor/main.py` dan `rcp-policy-processor/main.py` Python untuk menggabungkan pagar pembatas di folder pagar pembatas menjadi satu kebijakan RCP atau SCP. Ini menggabungkan pagar pembatas yang memiliki hal yang sama`Resource`,`Action`, dan. `Condition`
   c. Menggunakan AWS Identity and Access Management Access Analyzer untuk memvalidasi kebijakan akhir yang dioptimalkan. Jika ada temuan, pipa berhenti.
   d. Membuat `scps.json` dan `rcps.json` file, yang digunakan Terraform untuk membuat sumber daya.
   e. Menjalankan `terraform plan` perintah, yang membuat rencana eksekusi Terraform.
4. (Opsional) `Validate-Plan` CodeBuild Proyek menggunakan `bedrock-prompt/prompt.py` skrip untuk mengirim prompt ke Amazon Bedrock. Anda menentukan prompt dalam `bedrock-prompt/prompt.txt` file. Amazon Bedrock menggunakan Anthropic Claude Sonnet 3.5 untuk menghasilkan ringkasan perubahan yang diusulkan dengan menganalisis log Terraform dan Python.
5. CodePipeline menggunakan topik Amazon Simple Notification Service (Amazon SNS) untuk memberi tahu pemberi persetujuan bahwa perubahan harus ditinjau. Jika Amazon Bedrock menghasilkan ringkasan perubahan, notifikasi menyertakan ringkasan ini.
6. Penyetuju kebijakan menyetujui tindakan tersebut. CodePipeline Jika Amazon Bedrock membuat ringkasan perubahan, pemberi persetujuan dapat meninjau ringkasan CodePipeline sebelum menyetujui.
7. CodePipeline memulai `Apply` CodeBuild proyek. Proyek ini menggunakan Terraform untuk menerapkan perubahan RCP dan SCP di. AWS Organizations

Template IAc yang terkait dengan arsitektur ini juga menyebarkan sumber daya berikut yang mendukung pipeline manajemen kebijakan:

- Bucket Amazon S3 untuk menyimpan CodePipeline artefak dan skrip, seperti dan `scp-policy-processor/main.py` `bedrock-prompt/prompt.py`
- Kunci AWS Key Management Service (AWS KMS) yang mengenkripsi sumber daya yang dibuat oleh solusi ini

Alat

Layanan AWS

- Amazon Bedrock adalah layanan AI yang dikelola sepenuhnya yang membuat banyak model fondasi berkinerja tinggi tersedia untuk digunakan melalui API terpadu.
- AWS CodeBuildadalah layanan build yang dikelola sepenuhnya yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodePipelinembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Alat-alat lainnya

- HashiCorp Terraform adalah alat IAc yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.

Repositori kode

Kode untuk pola ini tersedia di organizations-policy-pipeline GitHub repositori. Berikut ini adalah file kunci yang terdapat dalam `sample-repository` folder:

- Dalam `environments` folder, `environments.json` berisi daftar lingkungan. Lingkungan adalah sekelompok target, dan mereka dapat berisi Akun AWS IDs atau unit organisasi (OUs).
- Dalam `rcp-management` folder:
  - `guardrails`Folder berisi pagar pembatas individual untuk Anda RCPs
  - `policies`Folder berisi individu RCPs
  - File `rcp-management.json` manifes membantu Anda mengelola pagar pembatas RCP, penuh RCPs, dan target terkaitnya.

- Dalam `scp-management` folder:
  - `guardrails`Folder berisi pagar pembatas individual untuk Anda SCPs
  - `policies`Folder berisi individu SCPs
  - File `scp-management.json` manifes membantu Anda mengelola pagar pembatas SCP, penuh SCPs, dan target terkaitnya.
- `utils`Folder berisi skrip yang dapat membantu Anda memigrasikan saat ini SCPs RCPs sehingga Anda dapat mengelolanya melalui pipeline. Untuk informasi selengkapnya, lihat bagian Informasi tambahan dari pola ini.

Praktik terbaik

- Sebelum Anda mengatur pipeline, kami sarankan Anda memverifikasi bahwa Anda belum mencapai batas AWS Organizations kuota Anda.
- Kami menyarankan Anda menggunakan akun AWS Organizations manajemen hanya untuk tugas-tugas yang harus dilakukan di akun itu. Untuk informasi selengkapnya, lihat Praktik terbaik untuk akun manajemen.

Epik

Siapkan akun target

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat repositori.	Buat repositori tempat tim operasi keamanan Anda akan mengelola kebijakan. Gunakan salah satu penyedia repositori pihak ketiga yang AWS CodeConnections mendukung.	DevOps insinyur
Delegasikan administrasi kebijakan.	Delegasikan administrasi AWS Organizations kebijakan ke akun anggota tempat Anda menerapkan pipeline. Untuk	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	petunjuk, lihat Membuat kebijakan delegasi berbasis sumber daya dengan. AWS Organizations Untuk kebijakan sampel, lihat Contoh kebijakan delegasi berbasis sumber daya di bagian Informasi tambahan dari pola ini.	
(Opsional) Aktifkan model pondasi.	Jika Anda ingin membuat ringkasan perubahan kebijakan, aktifkan akses ke model dasar Anthropic Claude 3.5 Sonnet di Amazon Bedrock di Akun AWS tempat Anda menerapkan pipeline. Untuk petunjukn ya, lihat Menambahkan atau menghapus akses ke model foundation Amazon Bedrock.	AWS Umum

Menyebarkan sumber daya untuk pipa

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Masukkan perintah berikut untuk mengkloning organizat ions-policy-pipeline repositori dari: GitHub `git clone https://g ithub.com/aws-samp les/organizations- policy-pipeline.git`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan metode penerapan Anda.	1. Di repositori kloning, buka file. `variables.tf` 2. Untuk`project_name` , masukkan awalan yang ingin Anda terapkan ke nama sumber daya yang digunakan. 3. Untuk`provider_type` , masukkan penyedia repositori jarak jauh. Nilai yang valid disediakan dalam file. 4. Untuk`full_repository_name` , masukkan nama repositori jarak jauh. 5. Untuk`branch_name` , masukkan nama cabang Git yang akan Anda gunakan untuk menyebarkan kebijakan. Dorongan atau penggabungan di cabang ini memulai pipa. Biasanya, ini adalah cabang utama. 6. Untuk`terraform_version` , masukkan versi Terraform yang Anda gunakan. 7. Untuk`enable_bedrock` , masukkan `true` jika Anda ingin Amazon Bedrock meringkas perubahan. Masukkan `false` jika Anda	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	tidak ingin menghasilkan ringkasan perubahan. 8. Untuk`tags`, masukkan pasangan nilai kunci yang ingin Anda tetapkan sebagai tag ke sumber daya yang diterapkan. 9. Simpan dan tutup file `variables.tf` .	

Menyebarkan pipa.	1. Masukkan perintah berikut untuk membuat rencana dan meninjau perubahan:
	`terraform plan`
	2. Masukkan perintah berikut untuk menerapkan rencana dan membuat infrastruktur pipa:
	`terraform apply`
		DevOps insinyur, Terraform

Hubungkan repositori jarak jauh.	Pada langkah sebelumnya, Terraform membuat CodeConnections koneksi ke repositori pihak ketiga. Di konsol Alat AWS Pengembang, ubah status koneksi dari `PENDING` ke`AVAILABLE` . Untuk petunjuk, lihat Memperbarui koneksi yang tertunda.	AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Langganan topik Amazon SNS.	Terraform membuat topik Amazon SNS. Berlangganan titik akhir ke topik dan konfirmasikan langganan sehingga pemberi persetujuan menerima pemberitahuan tentang tindakan persetujuan yang tertunda dalam pipeline. Untuk petunjuknya, lihat Membuat langganan ke topik Amazon SNS.	AWS Umum

Tentukan pagar dan kebijakan Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Isi repositori jarak jauh.	Dari repositori kloning, salin isi `sample-repository` folder ke repositori jarak jauh Anda. Ini termasuk `environments` , `rcp-management` , `scp-management` , dan `utils` folder.	DevOps insinyur
Tentukan lingkungan Anda.	1. Di `environments` folder, buka `environments.json` file. Ini adalah file di mana Anda menentukan target Akun AWS dan OUs untuk Anda RCPs dan SCPs. 2. Hapus contoh lingkungan.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Tambahkan lingkungan target Anda dalam format berikut: `[ { "ID": "<environment-name>", "Target": [ "<ou-name>:<ou-id>", "<account-name>:<account-id>" ] } ]` Di mana: • `<environment-name>` adalah nama yang Anda tetapkan ke grup OUs dan akun AWS. Anda dapat menggunakan nama ini dalam file manifes untuk menentukan di mana Anda ingin menerapkan kebijakan Anda. • `<ou-name>` adalah nama target OU. • `<ou-id>`adalah ID dari target OU.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `<account-name>` adalah nama targetnya Akun AWS. • `<account-id>` adalah ID target Akun AWS. Sebagai contoh, lihat repositori kode sumber. 4. Simpan dan tutup file `environments.json` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan pagar pembatas Anda.	1. Arahkan ke `rcp-management/guardrails` folder di repositori jarak jauh Anda. Ini adalah folder tempat Anda menentukan pagar pembatas untuk file manifes RCP Anda. Setiap pagar pembatas harus dalam file individual. File guardrail dapat berisi satu atau lebih pernyataan. **Note** Anda dapat menggunakan pagar pembatas yang sama dalam beberapa pernyataan dalam file manifes untuk SCPs dan. RCPs Jika Anda memodifikasi pagar pembatas, kebijakan apa pun yang mencakup pagar pembatas ini akan terpengaruh. 2. Hapus semua contoh pagar pembatas yang disalin dari repositori kode sumber. 3. Buat file.json baru dan beri nama deskriptif.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Buka file.json yang Anda buat. 5. Tentukan pagar pembatas dalam format berikut: `[ { "Sid": "<guardrail-name>", "Effect": "<effect-value>", "Action": [ "<action-name>" ], "Resource": "<resource-arn>", "Condition": { "<condition-operator>": { "<condition-key>": [ "<condition-value>" ] } } } ]` Di mana: • `<guardrail-name>` adalah nama unik untuk pagar pembatas. Nama ini tidak dapat digunakan untuk pagar pembatas lainnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `<effect-value>` harus `Allow` atau`Deny`. Untuk informasi lebih lanjut, lihat Efek. • `<action-name>` harus merupakan nama yang valid dari tindakan yang didukung layanan. Untuk informasi selengkapnya, lihat Tindakan. • `<resource-arn>`adalah Nama Sumber Daya Amazon (ARN) dari sumber daya yang diterapkan pagar pembatas. Anda juga dapat menggunakan karakter wildcard, seperti `*` atau`?`. Untuk informasi selengkapnya, lihat Sumber Daya. • `<condition-operator>` adalah operator kondisi yang valid. Untuk informasi selengkapnya, lihat Operator kondisi. • `<condition-key>`adalah kunci konteks kondisi global yang valid atau kunci konteks khusus layanan. Untuk informasi lebih lanjut, lihat Kondisi.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `<condition-value>`adalah nilai spesifik yang digunakan dalam suatu kondisi untuk mengevaluasi apakah pagar pembatas berlaku. Untuk informasi lebih lanjut, lihat Kondisi. Misalnya pagar pembatas RCP, lihat repositori kode sumber. 6. Simpan dan tutup file.json. 7. Ulangi langkah-langkah ini untuk membuat pagar pembatas RCP sebanyak yang diperlukan. 8. Ulangi langkah-langkah ini di `scp-management/guardrails` folder untuk membuat pagar pembatas sebanyak yang Anda butuhkan untuk Anda. SCPs Misalnya pagar pembatas SCP, lihat repositori kode sumber.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan kebijakan Anda.	1. Arahkan ke `rcp-management/policies` folder di repositori jarak jauh Anda. Ini adalah folder tempat Anda menentukan kebijakan lengkap untuk file manifes RCP Anda. Setiap kebijakan harus berupa file individual. **Note** Jika Anda mengubah kebijakan di folder ini, perubahan kebijakan memengaruhi akun apa pun atau OUs kebijakan ini diterapkan, sebagaimana didefinisikan dalam file manifes. 2. Hapus kebijakan contoh apa pun yang disalin dari repositori kode sumber. 3. Buat file.json baru dan beri nama deskriptif. 4. Buka file.json yang Anda buat. 5. Tentukan RCP. Misalnya RCPs, lihat repositori kode	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	sumber atau lihat Contoh kebijakan kontrol sumber daya dalam dokumentasi. AWS Organizations 6. Simpan dan tutup file.json. 7. Ulangi langkah-langkah ini untuk membuat RCPs sebanyak yang diperlukan. 8. Ulangi langkah-langkah ini di `scp-management/policies` folder untuk membuat SCPs sebanyak yang diperlukan. Misalnya SCPs, lihat repositori kode sumber atau lihat Contoh kebijakan kontrol layanan dalam dokumentasi. AWS Organizations	

Menggunakan file manifes untuk mengelola kebijakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan file manifes.	1. Di `rcp-management` folder, buka `rcp-management.json` file. Ini adalah file tempat Anda menentukan pagar pembatas RCP mana dan sepenuhnya RCPs berlaku untuk lingkungan target Anda. Untuk contoh file ini, lihat repositori kode sumber.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Hapus pernyataan contoh. 3. Tambahkan pernyataan baru dalam format berikut: `[ { "SID": "<statement-name>", "Target": { "Type": "<target-type>", "ID": "<target-name>" }, "Guardrails": [ "<guardrail-name>" ], "Policy": "<policy-name>", "Comments": "<comment-text>" } ]` Di mana: • `<statement-name>` adalah nama unik untuk pernyataan itu. • `<target-type>` adalah jenis target di mana Anda ingin menerapkan kebijakan. Nilai yang valid adalah `Account`, `OU`, `Environment`, atau `Tag`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	• `<target-name>` adalah pengenal target di mana Anda ingin menerapka n kebijakan. Masukkan salah satu dari berikut ini:
	  • Untuk Akun AWS, masukkan pengenal sebagai`<account-name>:<account-id>` .
	  • Untuk OU, masukkan pengenal sebagai`<OU-name>:<ou-id>` .
	  • Untuk lingkungan, masukkan nama unik yang Anda tentukan dalam `environme nts.json` file.
	  • Untuk tag, masukkan pasangan kunci-nilai sebagai. `<tag-key> :<tag-value>`
	• `<guardrail-name>`adalah nama unik dari pagar pembatas RCP yang Anda tentukan di folder. `rcp-manag ement/guardrails` Anda dapat menambahk an beberapa pagar pembatas dalam elemen ini. Anda dapat membiarkan bidang ini	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kosong jika Anda tidak ingin menerapkan pagar pembatas. • `<policy-name>` adalah nama unik dari RCP yang Anda tentukan di `rcp-management/policies` folder. Anda hanya dapat menambahkan satu kebijakan dalam elemen ini. Anda dapat membiarkan bidang ini kosong jika Anda tidak ingin menerapkan kebijakan. • `<comment-text>` adalah deskripsi yang dapat Anda masukkan untuk tujuan dokumentasi. Bidang ini tidak digunakan selama pemrosesan pipa. Anda dapat membiarkan bidang ini kosong jika Anda tidak ingin menambahkan komentar. 4. Ulangi langkah-langkah ini untuk menambahkan pernyataan sebanyak yang diperlukan RCPs untuk mengonfigurasi organisasi Anda. 5. Simpan dan tutup file `rcp-management.json` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	6. Di `scp-management` folder, ulangi langkah-l angkah ini dalam `scp-management.json` file. Ini adalah file tempat Anda menentukan pagar pembatas SCP mana dan sepenuhnya SCPs berlaku untuk lingkungan target Anda. Untuk contoh file ini, lihat repositori kode sumber.	
Mulai pipa.	Komit dan dorong perubahan ke cabang repositori jarak jauh yang Anda tentukan dalam file. `variables.tf` Biasanya, ini adalah `main` cabang. CI/CD Pipa secara otomatis dimulai. Jika ada kesalahan pipeline, lihat bagian Pemecahan Masalah dari pola ini.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyetujui perubahan.	Ketika `Validate-Plan` CodeBuild proyek selesai, pemberi persetujuan kebijakan menerima pemberitahuan melalui topik Amazon SNS yang sebelumnya Anda konfigurasikan. Lakukan hal-hal berikut: 1. Buka pesan notifikasi. 2. Jika tersedia, tinjau ringkasan perubahan kebijakan. 3. Ikuti instruksi di Menyetujui atau menolak tindakan persetujuan di. CodePipeline	AWS Umum, Penyetuju kebijakan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi penerapan.	1. Masuk ke AWS Organizations konsol di akun yang merupakan administrator yang didelegasikan. AWS Organizations 2. Pada halaman Kebijakan kontrol layanan, konfirmasikan SCPs bahwa yang Anda buat terdaftar. 3. Pilih SCP yang dikelola melalui pipeline dan konfirmasikan bahwa itu berlaku untuk target yang dimaksud. 4. Pada halaman Kebijakan kontrol sumber daya, konfirmasikan RCPs bahwa yang Anda buat terdaftar. 5. Pilih RCP yang dikelola melalui pipeline dan konfirmasikan bahwa itu berlaku untuk target yang dimaksud.	AWS Umum

Pemecahan masalah

Isu	Solusi
Kesalahan file manifes dalam `Validate-Plan` fase pipa	Pesan “Kesalahan pipa dalam fase Validasi & Rencana untuk file manifes” muncul di output pipeline jika ada kesalahan dalam `rcp-management.json` file `scp-management.json` atau. Kemungkinan kesalahan termasuk

Isu	Solusi
	nama lingkungan yang salah, kolom atau nilai duplikat SIDs, atau tidak valid. Lakukan hal-hal berikut: 1. Ikuti petunjuk di Lihat detail build di AWS CodeBuild. 2. Di log build, temukan kesalahan validasi. Kesalahan memberikan informasi lebih lanjut tentang apa yang menyebabkan build gagal. 3. Perbarui file.json yang sesuai. 4. Komit dan dorong file yang diperbarui ke repositori jarak jauh. Pipa dimulai ulang. 5. Pantau status untuk mengonfirmasi bahwa kesalahan validasi telah teratasi.

Isu	Solusi
Temuan IAM Access Analyzer dalam `Validate-Plan` fase pipa	Pesan “Temuan di IAM Access Analyzer selama fase Validasi & Rencana” muncul di output pipeline jika ada kesalahan dalam pagar pembatas atau definisi kebijakan. Pola ini menggunakan IAM Access Analyzer untuk memvalidasi kebijakan akhir. Lakukan hal-hal berikut: 1. Ikuti petunjuk di Lihat detail build di AWS CodeBuild. 2. Di log build, temukan error validasi IAM Access Analyzer. Kesalahan memberikan informasi lebih lanjut tentang apa yang menyebabkan build gagal. Untuk informasi selengkapnya tentang jenis temuan, lihat referensi pemeriksaan validasi kebijakan IAM. 3. Perbarui file.json yang sesuai untuk pagar pembatas atau kebijakan. 4. Komit dan dorong file yang diperbarui ke repositori jarak jauh. Pipa dimulai ulang. 5. Pantau status untuk mengonfirmasi bahwa kesalahan validasi telah teratasi.

Sumber daya terkait

- Referensi elemen kebijakan JSON (dokumentasi IAM)
- Kebijakan kontrol sumber daya (AWS Organizations dokumentasi)
- Kebijakan kontrol layanan (AWS Organizations dokumentasi)
- Menambah atau menghapus akses ke model fondasi Amazon Bedrock (dokumentasi Amazon Bedrock)
- Menyetujui atau menolak tindakan persetujuan dalam CodePipeline (dokumentasi) CodePipeline

Informasi tambahan

Contoh kebijakan delegasi berbasis sumber daya

Berikut ini adalah contoh kebijakan delegasi berbasis sumber daya untuk. AWS Organizations Ini memungkinkan akun pengelola yang didelegasikan untuk mengelola SCPs dan RCPs untuk organisasi. Dalam contoh kebijakan berikut, ganti `<MEMBER_ACCOUNT_ID>` dengan ID akun tempat Anda menerapkan pipeline manajemen kebijakan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DelegationToAudit",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<MEMBER_ACCOUNT_ID>:root"
      },
      "Action": [
        "organizations:ListTargetsForPolicy",
        "organizations:CreatePolicy",
        "organizations:DeletePolicy",
        "organizations:AttachPolicy",
        "organizations:DetachPolicy",
        "organizations:DisablePolicyType",
        "organizations:EnablePolicyType",
        "organizations:UpdatePolicy",
        "organizations:DescribeEffectivePolicy",
        "organizations:DescribePolicy",
        "organizations:DescribeResourcePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

Migrasi kebijakan yang ada ke pipeline

Jika sudah ada SCPs atau RCPs yang ingin Anda migrasi dan kelola melalui pipeline ini, Anda dapat menggunakan skrip Python di folder `sample-repository/utils` repositori kode. Skrip ini meliputi:

- `check-if-scp-exists-in-env.py`— Skrip ini memeriksa apakah kebijakan tertentu berlaku untuk target apa pun di lingkungan tertentu, yang Anda tentukan dalam `environments.json` file. Masukkan perintah berikut untuk menjalankan skrip ini:

```
python3 check-if-scp-exists-in-env.py \
   --policy-type <POLICY_TYPE> \
   --policy-name <POLICY_NAME> \
   --env-id <ENV_ID>
```

  Ganti yang berikut ini dalam perintah ini:

  - `<POLICY_TYPE>`adalah `scp` atau `rcp`
  - `<POLICY_NAME>`adalah nama SCP atau RCP
  - `<ENV_ID>`adalah ID lingkungan yang Anda tentukan dalam `environments.json` file
- `create-environments.py`— Skrip ini membuat file environments.json berdasarkan saat ini SCPs dan RCPs di lingkungan Anda. Ini tidak termasuk kebijakan yang diterapkan melalui. AWS Control Tower Masukkan perintah berikut untuk menjalankan skrip ini, di `<POLICY_TYPE>` mana `scp` atau`rcp`:

```
python create-environments.py --policy-type <POLICY_TYPE>
```

- `verify-policies-capacity.py`— Skrip ini memeriksa setiap lingkungan yang Anda tentukan untuk menentukan berapa banyak kapasitas yang tersisa untuk setiap kuota AWS Organizations terkait kebijakan. Anda menentukan lingkungan untuk memeriksa dalam `environments.json` file. Masukkan perintah berikut untuk menjalankan skrip ini, di `<POLICY_TYPE>` mana `scp` atau`rcp`:

```
python verify-policies-capacity.py --policy-type <POLICY_TYPE>
```

Kelola set AWS IAM Identity Center izin sebagai kode dengan menggunakan AWS CodePipeline

Andre Cavalcante dan Claison Amorim, Amazon Web Services

Ringkasan

AWS IAM Identity Center membantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua aplikasi dan aplikasi Anda. Akun AWS Anda dapat membuat dan mengelola identitas pengguna di Pusat Identitas IAM, atau Anda dapat menghubungkan sumber identitas yang ada, seperti domain Microsoft Active Directory atau penyedia identitas eksternal (iDP). IAM Identity Center menyediakan pengalaman administrasi terpadu untuk menentukan, menyesuaikan, dan menetapkan akses halus ke lingkungan Anda AWS dengan menggunakan set izin. Set izin berlaku untuk pengguna dan grup federasi dari toko identitas Pusat Identitas IAM atau iDP eksternal Anda.

Pola ini membantu Anda mengelola set izin IAM Identity Center sebagai kode di lingkungan multi-akun Anda yang dikelola sebagai organisasi. AWS Organizations Dengan pola ini, Anda dapat mencapai yang berikut:

- Membuat, menghapus, dan memperbarui set izin
- Buat, perbarui, atau hapus penetapan set izin ke target Akun AWS, unit organisasi (OUs), atau root organisasi Anda.

Untuk mengelola izin dan penugasan Pusat Identitas IAM sebagai kode, solusi ini menerapkan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang menggunakan dan. AWS CodeBuild AWS CodePipeline Anda mengelola set izin dan tugas dalam template JSON yang Anda simpan di repositori jarak jauh. Ketika EventBridge aturan Amazon mendeteksi perubahan pada repositori atau mendeteksi modifikasi pada akun di OU target, itu memulai fungsi. AWS Lambda Fungsi Lambda memulai CI/CD pipeline yang memperbarui set izin dan penetapan di IAM Identity Center.

Prasyarat dan batasan

Prasyarat

- Lingkungan multi-akun yang dikelola sebagai organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Membuat organisasi.

- Pusat Identitas IAM, diaktifkan dan dikonfigurasi dengan sumber identitas. Untuk informasi selengkapnya, lihat Memulai di dokumentasi Pusat Identitas IAM.
- Akun anggota yang terdaftar sebagai administrator yang didelegasikan untuk hal-hal berikut: Layanan AWS
  - Pusat Identitas IAM — Untuk petunjuk, lihat Mendaftarkan akun anggota di dokumentasi Pusat Identitas IAM.
  - AWS Organizations Untuk petunjuk, lihat Administrator yang didelegasikan untuk AWS Organizations. Akun ini harus memiliki izin untuk membuat daftar dan mendeskripsikan akun dan OUs.

> **Note**
>
> Anda harus menggunakan akun yang sama dengan administrator yang didelegasikan untuk kedua layanan.

- Izin untuk menerapkan AWS CloudFormation tumpukan di akun administrator yang didelegasikan Pusat Identitas IAM dan di akun manajemen organisasi. Untuk informasi selengkapnya, lihat Mengontrol akses dalam CloudFormation dokumentasi.
- Bucket Amazon Simple Storage Service (Amazon S3) di akun administrator yang didelegasikan IAM Identity Center. Anda mengunggah kode artefak ke dalam ember ini. Untuk petunjuknya, lihat Membuat bucket di dokumentasi Amazon S3.
- ID akun akun manajemen organisasi. Untuk petunjuk, lihat Menemukan Akun AWS ID Anda.
- Sebuah repositori di host kode sumber Anda, seperti. GitHub

Batasan

- Pola ini tidak dapat digunakan untuk mengelola atau menetapkan set izin untuk lingkungan akun tunggal atau untuk akun yang tidak dikelola sebagai organisasi. AWS Organizations
- Izin menetapkan nama, penugasan IDs, dan tipe utama Pusat Identitas IAM dan IDs tidak dapat dimodifikasi setelah penerapan.
- Pola ini membantu Anda membuat dan mengelola izin khusus. Anda tidak dapat menggunakan pola ini untuk mengelola atau menetapkan izin yang telah ditentukan sebelumnya.
- Pola ini tidak dapat digunakan untuk mengelola izin yang ditetapkan untuk akun manajemen organisasi.

Arsitektur

Arsitektur target

Diagram menunjukkan alur kerja berikut:

1. Seorang pengguna membuat salah satu dari perubahan berikut:
   - Melakukan satu atau beberapa perubahan ke repositori jarak jauh, seperti GitHub
   - Memodifikasi akun di OU di AWS Organizations
2. Jika pengguna melakukan perubahan ke repositori jarak jauh ke cabang utama, pipeline dimulai.

   Jika pengguna memodifikasi akun di OU, maka `MoveAccount` EventBridge aturan mendeteksi perubahan dan memulai fungsi Lambda di akun manajemen organisasi.
3. Fungsi Lambda yang dimulai memulai pipeline masuk CI/CD . CodePipeline
4. CodePipeline memulai `TemplateValidation` CodeBuild proyek. `TemplateValidation` CodeBuild Proyek ini menggunakan skrip Python di repositori jarak jauh untuk memvalidasi template set izin. CodeBuild memvalidasi yang berikut:
   - Nama set izin unik.
   - Pernyataan penugasan IDs (`Sid`) unik.
   - Definisi kebijakan dalam `CustomPolicy` parameter dan valid. (Validasi ini menggunakan AWS Identity and Access Management Access Analyzer.)
   - Nama Sumber Daya Amazon (ARNs) dari kebijakan terkelola valid.
5. Grup `PermissionSet` tindakan dalam `Deploy` CodeBuild proyek digunakan AWS SDK for Python (Boto3) untuk menghapus, membuat, atau memperbarui set izin di Pusat Identitas IAM. Hanya set izin dengan `SSOPipeline:true` tag yang terpengaruh. Semua set izin yang dikelola melalui pipeline ini memiliki tag ini.
6. Grup `Assignments` tindakan dalam `Deploy` CodeBuild proyek menggunakan Terraform untuk menghapus, membuat, atau memperbarui tugas di Pusat Identitas IAM. File status backend Terraform disimpan dalam bucket Amazon S3 di akun yang sama.
7. CodeBuild memperbarui set izin dan tugas di IAM Identity Center.

Otomatisasi dan skala

Karena semua akun baru di lingkungan multi-akun dipindahkan ke unit organisasi tertentu di AWS Organizations, solusi ini secara otomatis berjalan dan memberikan set izin yang diperlukan ke semua akun yang Anda tentukan dalam templat penetapan. Tidak diperlukan otomatisasi tambahan atau tindakan penskalaan.

Di lingkungan yang besar, jumlah permintaan API ke IAM Identity Center dapat menyebabkan solusi ini berjalan lebih lambat. Terraform dan Boto3 secara otomatis mengelola pelambatan untuk meminimalkan penurunan kinerja apa pun.

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- AWS CodeBuildadalah layanan build terkelola penuh yang membantu Anda mengkompilasi kode sumber, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan.
- AWS CodePipelinemembantu Anda dengan cepat memodelkan dan mengkonfigurasi berbagai tahapan rilis perangkat lunak dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perubahan perangkat lunak secara terus menerus.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS IAM Identity Centermembantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua aplikasi Anda Akun AWS dan cloud.
- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- AWS SDK for Python (Boto3)adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, perpustakaan, atau skrip Python Anda. Layanan AWS
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Kode untuk pola ini tersedia di repositori aws-iam-identity-center-pipeline. Folder template dalam repositori menyertakan contoh template untuk set izin dan tugas. Ini juga mencakup AWS CloudFormation template untuk menyebarkan CI/CD pipeline dan AWS sumber daya di akun target.

Praktik terbaik

- Sebelum Anda mulai memodifikasi set izin dan templat penetapan, sebaiknya Anda merencanakan set izin untuk organisasi Anda. Pertimbangkan izin apa yang seharusnya, akun atau OUs set izin mana yang harus diterapkan, dan prinsip Pusat Identitas IAM (pengguna atau grup) mana yang harus dipengaruhi oleh kumpulan izin. Izin menetapkan nama, asosiasi IDs, dan tipe utama Pusat Identitas IAM dan IDs tidak dapat dimodifikasi setelah penerapan.
- Patuhi prinsip hak istimewa paling sedikit dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Memberikan hak istimewa terkecil dan praktik terbaik Keamanan dalam dokumentasi AWS Identity and Access Management (IAM).

Epik

Rencanakan set izin dan tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Dalam shell bash, masukkan perintah berikut. Ini mengkloning repositori aws-iam-identity-center-pipeline dari. GitHub `git clone https://github.com/aws-samples/aws-iam-identity-center-pipeline.git`	DevOps insinyur
Tentukan set izin.	1. Di repositori kloning, navigasikan ke `templates/permissionsets` folder, lalu buka salah satu templat yang tersedia.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Dalam `Name` parameter, masukkan nama untuk set izin. Nilai ini harus unik, dan tidak dapat diubah setelah penerapan. 3. Dalam `Description` parameter, jelaskan secara singkat set izin, seperti kasus penggunaannya. 4. Dalam `SessionDuration` parameter, tentukan lamanya waktu pengguna dapat masuk ke file Akun AWS. Gunakan format durasi ISO-8601 (Wikipedia), seperti selama 4 jam. `PT4H` Jika tidak ada nilai yang ditentukan, default di IAM Identity Center adalah 1 jam. 5. Dalam `RelayState` parameter, tentukan URL yang menyediakan akses cepat ke konsol yang paling sesuai untuk peran pengguna. 6. Sesuaikan kebijakan dalam set izin. Semua parameter berikut adalah opsional dan dapat dimodifikasi setelah penerapan. Anda harus menggunakan setidaknya satu parameter untuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	menentukan kebijakan dalam kumpulan izin:
	• Dalam `ManagedPolicies` parameter, masukkan kebijakan ARNs AWS terkelola apa pun yang ingin Anda tetapkan.
	• Dalam `CustomerManagedPolicies` parameter, masukkan nama kebijakan yang dikelola pelanggan yang ingin Anda tetapkan. Jangan gunakan ARN.
	• Dalam `PermissionBoundary` parameter, lakukan hal berikut untuk menetapkan batas izin:
	  • Jika Anda menggunakan kebijakan AWS terkelola sebagai batas izin, masuk, masuk`PolicyType`, dan masuk `AWSPolicy`, masukkan ARN kebijakan tersebut.
	  • Jika Anda menggunakan kebijakan yang dikelola pelanggan sebagai batas izin, masuk, masuk	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`PolicyType Customer`, dan masuk`Policy`, masukkan nama kebijakan. Jangan gunakan ARN. • Dalam `CustomPolicy` parameter, tentukan kebijakan kustom, berformat JSON yang ingin Anda tetapkan. Untuk informasi selengkapnya tentang struktur kebijakan JSON, lihat Ikhtisar kebijakan JSON. 7. Simpan dan tutup template set izin. Kami menyarankan Anda menyimpan file dengan nama yang cocok dengan nama set izin. 8. Ulangi proses ini untuk membuat set izin sebanyak yang diperlukan untuk organisasi Anda, dan hapus templat sampel apa pun yang tidak diperlukan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan tugas.	1. Di repositori kloning, arahkan ke `templates/assignments` folder, lalu buka. `iam-identitycenter-assigments.json` File ini menjelaskan bagaimana Anda ingin menetapkan set izin ke Akun AWS atau OUs. 2. Dalam `SID` parameter, masukkan pengidentifikasi untuk penugasan. Nilai ini harus unik, dan tidak dapat dimodifikasi setelah penerapan. 3. Dalam `Target` parameter , tentukan akun atau organisasi tempat Anda ingin menerapkan set izin. Nilai yang valid adalah akun IDs, OUs, atau`root`. `root`memberikan izin yang ditetapkan ke semua akun anggota di organisasi, tidak termasuk akun manajemen. Masukkan nilai dalam tanda kutip ganda, dan pisahkan beberapa nilai dengan koma. Akun IDs dan OUs harus mengikuti pola: `{{account_name}}:{{account_id}}` atau`{{ou_name}}:`	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`{{ou_id}}` . Jika Anda ingin menetapkan izin secara rekursif ke nested OUs, gunakan pola OU dengan wildcard di bagian akhir. Contoh: `{{ou_name}}:{{ou_id}}:*` 4. Dalam `PrincipalType` parameter, masukkan tipe utama IAM Identity Center yang akan terpengaruh oleh set izin. Nilai-nilai yang valid adalah `USER` atau `GROUP`. Nilai ini tidak dapat diubah setelah penerapan. 5. Dalam `PrincipalID` parameter, masukkan nama pengguna atau grup di toko identitas Pusat Identitas IAM yang akan terpengaruh oleh set izin. Nilai ini tidak dapat diubah setelah penerapan. 6. Dalam `PermissionSetName` parameter, masukkan nama set izin yang ingin Anda tetapkan. 7. Ulangi langkah 2—6 untuk membuat tugas sebanyak yang diperlukan dalam file ini. Biasanya, ada satu tugas untuk setiap set izin. Hapus tugas sampel apa pun yang tidak diperlukan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	8. Simpan dan tutup file `iam-identitycenter-assigments.json` .	

Menyebarkan set izin dan tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan sumber daya di akun administrator yang didelegasikan IAM Identity Center.	1. Di akun administrator yang didelegasikan IAM Identity Center, buka konsol.CloudFormation 2. Menyebarkan `iam-identitycenter-pipeline.yaml` template. Berikan tumpukan nama yang jelas dan deskriptif, dan perbarui parameter seperti yang diinstruksikan. Untuk petunjuk, lihat Membuat tumpukan dalam CloudFormation dokumentasi.	DevOps insinyur
Menyebarkan sumber daya di akun AWS Organizations manajemen.	1. Masuk ke akun manajemen organisasi. 2. Buka konsol AWS CloudFormation. 3. Di bilah navigasi, pilih nama yang saat ini ditampilkan Wilayah AWS. Kemudian pilih `us-east-1` Region. Wilayah ini diperlukan agar `MoveAccount` EventBrid	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ge aturan dapat mendeteksi AWS CloudTrail peristiwa yang terkait dengan perubahan organisasi. 4. Menyebarkan `iam-identitycenter-organization` template. Berikan tumpukan nama yang jelas dan deskripti f, dan perbarui parameter seperti yang diinstruksikan. Untuk petunjuk, lihat Membuat tumpukan dalam CloudFormation dokumentasi.	
Selesaikan pengaturan repositori jarak jauh.	Ubah status AWS CodeConnections koneksi dari `PENDING` ke`AVAILABLE` . Koneksi ini dibuat saat Anda menerapkan CloudFormation tumpukan. Untuk petunjuk, lihat Memperbarui koneksi yang tertunda dalam CodeConnections dokumentasi.	DevOps insinyur
Unggah file ke repositori jarak jauh.	Unggah semua file yang telah Anda unduh dari `aws-samples` repositori dan edit pada langkah sebelumnya ke repositori jarak jauh. Perubahan pada `main` cabang memulai pipeline, yang membuat atau memperbarui set izin dan penugasan.	DevOps insinyur

Memperbarui set izin dan tugas

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui set izin dan tugas.	Saat EventBridge aturan `MoveAccount` Amazon mendeteksi modifikasi pada akun di organisasi, CI/CD pipeline akan dimulai secara otomatis dan memperbarui set izin. Misalnya, jika Anda menambahkan akun ke OU yang ditentukan dalam file JSON tugas, maka CI/CD pipeline akan menerapkan izin yang disetel ke akun baru. Jika Anda ingin memodifik asi set izin dan penetapan yang diterapkan, perbarui file JSON dan kemudian komit ke repositori jarak jauh. Perhatikan hal berikut saat menggunakan CI/CD pipeline untuk mengelola kumpulan izin dan asosiasi yang diterapkan sebelumnya: • Jika Anda mengubah nama set izin, CI/CD pipeline akan menghapus set izin asli dan membuat yang baru. • Pipeline ini hanya mengelola set izin yang memiliki `SSOPipeli ne:true` tag.	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Anda dapat memiliki beberapa set izin dan templat penetapan di folder yang sama di repositori. • Jika Anda menghapus templat, pipeline akan menghapus set penetapan atau izin. • Jika Anda menghapus seluruh blok JSON penetapan, pipeline akan menghapus penugasan dari IAM Identity Center. • Anda tidak dapat menghapus set izin yang ditetapkan ke Akun AWS. Pertama, Anda harus membatalkan penetapan set izin.	

Pemecahan masalah

Isu	Solusi
Akses kesalahan ditolak	Konfirmasikan bahwa Anda memiliki izin yang diperlukan untuk menyebarkan CloudForm ation templat dan sumber daya yang ditentuka n di dalamnya. Untuk informasi selengkapnya, lihat Mengontrol akses dalam CloudFormation dokumentasi.
Kesalahan saluran pipa dalam fase validasi	Kesalahan ini muncul jika ada kesalahan dalam set izin atau templat penugasan.

Isu	Solusi
	1. Di CodeBuild, lihat detail build. 2. Di log build, temukan error validasi yang memberikan informasi lebih lanjut tentang apa yang menyebabkan build gagal. 3. Perbarui set izin atau templat penugasan, lalu komit ke repositori. 4. CI/CD Pipa memulai ulang CodeBuild proyek. Pantau status untuk mengonfirmasi bahwa kesalahan validasi telah teratasi.

Sumber daya terkait

- Set izin (dokumentasi Pusat Identitas IAM)

Kelola kredensil menggunakan AWS Secrets Manager

Durga Prasad Cheepuri, Amazon Web Services

Ringkasan

Pola ini memandu Anda menggunakan AWS Secrets Manager untuk secara dinamis mengambil kredensi database untuk aplikasi Java Spring.

Di masa lalu, ketika Anda membuat aplikasi kustom yang mengambil informasi dari database, Anda biasanya harus menanamkan kredensi (rahasia) untuk mengakses database secara langsung dalam aplikasi. Ketika tiba waktunya untuk memutar kredensil, Anda harus menginvestasikan waktu untuk memperbarui aplikasi untuk menggunakan kredensil baru, dan kemudian mendistribusikan aplikasi yang diperbarui. Jika Anda memiliki beberapa aplikasi yang berbagi kredensil dan Anda melewatkan memperbarui salah satunya, aplikasi akan gagal. Karena risiko ini, banyak pengguna memilih untuk tidak secara teratur memutar kredensialnya, yang secara efektif menggantikan satu risiko dengan risiko lainnya.

Secrets Manager memungkinkan Anda mengganti kredensi hard-code dalam kode Anda (termasuk kata sandi) dengan panggilan API untuk mengambil rahasia secara terprogram. Ini membantu memastikan bahwa rahasia tidak dapat dikompromikan oleh seseorang yang memeriksa kode Anda, karena rahasianya tidak ada. Anda juga dapat mengonfigurasi Secrets Manager untuk memutar rahasia secara otomatis sesuai dengan jadwal yang Anda tentukan. Ini memungkinkan Anda untuk mengganti rahasia jangka panjang dengan rahasia jangka pendek, yang membantu mengurangi risiko kompromi secara signifikan. Untuk informasi selengkapnya, lihat dokumentasi AWS Secrets Manager.

Prasyarat dan batasan

Prasyarat

- Akun AWS dengan akses ke Secrets Manager
- Aplikasi Java Spring

Arsitektur

Tumpukan teknologi sumber

- Aplikasi Java Spring dengan kode yang mengakses database, dengan kredensi DB dikelola dari file application.properties.

Tumpukan teknologi target

- Aplikasi Java Spring dengan kode yang mengakses database, dengan kredensi DB dikelola di Secrets Manager. File application.properties menyimpan rahasia ke Secrets Manager.

Integrasi Secrets Manager dengan aplikasi

Alat

- Secrets Manager — AWS Secrets Manager adalah layanan AWS yang memudahkan Anda mengelola rahasia. Rahasia dapat berupa kredensial basis data, kata sandi, kunci API pihak ketiga, dan bahkan teks acak. Anda dapat menyimpan dan mengontrol akses ke rahasia ini secara terpusat dengan menggunakan konsol Secrets Manager, antarmuka baris perintah Secrets Manager (CLI), atau Secrets Manager API dan. SDKs

Epik

Simpan rahasia di Secrets Manager

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Simpan kredensil DB sebagai rahasia di Secrets Manager.	Simpan Amazon Relational Database Service (Amazon RDS) atau kredensi DB lainnya sebagai rahasia di Secrets Manager dengan mengikuti langkah-langkah dalam Membuat rahasia dalam dokumentasi Secrets Manager.	Sys Admin

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tetapkan izin untuk aplikasi Spring untuk mengakses Secrets Manager.	Tetapkan izin yang sesuai berdasarkan bagaimana aplikasi Java Spring menggunakan Secrets Manager. Untuk mengontrol akses ke rahasia, buat kebijakan berdasarkan informasi yang disediakan dalam dokumentasi Secrets Manager, di bagian Menggunakan kebijakan berbasis identitas (Kebijakan IAM) dan ABAC for Secrets Manager dan Menggunakan kebijakan berbasis sumber daya untuk Secrets Manager. Ikuti langkah-langkah di bagian Mengambil nilai rahasia dalam dokumentasi Secrets Manager.	Sys Admin

Perbarui aplikasi Musim Semi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan dependensi JAR untuk menggunakan Secrets Manager.	Lihat bagian Informasi tambahan untuk detailnya.	Pengembang Java
Tambahkan detail rahasia ke aplikasi Spring.	Perbarui file application.properties dengan nama rahasia, titik akhir, dan Wilayah AWS. Sebagai contoh, lihat bagian Informasi tambahan.	Pengembang Java

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Perbarui kode pengambilan kredensi DB di Java.	Dalam aplikasi, perbarui kode Java yang mengambil kredensi DB untuk mengambil detail tersebut dari Secrets Manager. Misalnya kode, lihat bagian Informasi tambahan.	Pengembang Java

Sumber daya terkait

- Dokumentasi AWS Secrets Manager
- Menggunakan kebijakan berbasis identitas (Kebijakan IAM) dan ABAC for Secrets Manager
- Menggunakan kebijakan berbasis sumber daya untuk Secrets Manager
- Kode sampel

Informasi tambahan

Menambahkan dependensi JAR untuk menggunakan Secrets Manager

Maven:

```
<groupId>com.amazonaws</groupId>
    <artifactId>aws-java-sdk-secretsmanager</artifactId>
    <version>1.11. 355 </version>
```

Gradle:

```
compile group: 'com.amazonaws', name: 'aws-java-sdk-secretsmanager', version:
 '1.11.355'
```

Memperbarui file application.properties dengan detail rahasianya

```
spring.aws.secretsmanager.secretName=postgres-local
spring.aws.secretsmanager.endpoint=secretsmanager.us-east-1.amazonaws.com
spring.aws.secretsmanager.region=us-east-1
```

Memperbarui kode pengambilan kredensi DB di Java

```
String  secretName  =  env.getProperty("spring.aws.secretsmanager.secretName");
String  endpoints  =  env.getProperty("spring.aws.secretsmanager.endpoint");
String  AWS Region  =  env.getProperty("spring.aws.secretsmanager.region");
AwsClientBuilder.EndpointConfiguration  config  =  new
  AwsClientBuilder.EndpointConfiguration(endpoints, AWS Region);
AWSSecretsManagerClientBuilder  clientBuilder  =
  AWSSecretsManagerClientBuilder.standard();
clientBuilder.setEndpointConfiguration(config);
AWSSecretsManager  client  =  clientBuilder.build();

ObjectMapper  objectMapper  =  new  ObjectMapper();

JsonNode  secretsJson  =  null;

ByteBuffer  binarySecretData;

GetSecretValueRequest  getSecretValueRequest  =  new
  GetSecretValueRequest().withSecretId(secretName);

GetSecretValueResult  getSecretValueResponse  =  null;

try  {
     getSecretValueResponse  =  client.getSecretValue(getSecretValueRequest);
    }

catch  (ResourceNotFoundException  e)  {
     log.error("The requested secret "  +  secretName  +  " was not found");
    }

catch  (InvalidRequestException  e)  {
     log.error("The request was invalid due to: "  +  e.getMessage());
     }

catch  (InvalidParameterException  e)  {
     log.error("The request had invalid params: "  +  e.getMessage());
     }
if  (getSecretValueResponse  ==  null)  {
     return  null;
     }  // Decrypted secret using the associated KMS key // Depending on whether the
 secret was a string or binary, one of these fields will be populated
```

```
String secret = getSecretValueResponse.getSecretString();

if (secret != null) {
     try {
                secretsJson  =  objectMapper.readTree(secret);
           }

     catch  (IOException  e)  {
                log.error("Exception while retrieving secret values: "  +
  e.getMessage());
           }
}

else  {
     log.error("The Secret String returned is null");

     return null;

     }
     String  host  =  secretsJson.get("host").textValue();
     String  port  =  secretsJson.get("port").textValue();
     String  dbname  =  secretsJson.get("dbname").textValue();
     String  username  =  secretsJson.get("username").textValue();
     String  password  =  secretsJson.get("password").textValue();
}
```

Pantau ElastiCache kluster Amazon untuk enkripsi saat istirahat

Abhishek Adawane, Amazon Web Services

Ringkasan

Amazon ElastiCache adalah layanan Amazon Web Services (AWS) yang menyediakan solusi caching berkinerja tinggi, terukur, dan hemat biaya untuk mendistribusikan penyimpanan data dalam memori atau lingkungan cache di cloud. Ini mengambil data dari throughput tinggi dan latensi rendah, penyimpanan data dalam memori. Fungsionalitas ini menjadikannya pilihan populer untuk kasus penggunaan real-time seperti caching, toko sesi, game, layanan geo-spasial, analitik real-time, dan antrian. ElastiCache menawarkan penyimpanan data Redis dan Memcached, yang keduanya menyediakan waktu respons sub-milidetik.

Enkripsi data membantu mencegah pengguna yang tidak sah membaca data sensitif yang tersedia di kluster Redis Anda dan sistem penyimpanan cache terkait mereka. Ini termasuk data yang disimpan ke media persisten, yang dikenal sebagai data saat istirahat, dan data yang dapat dicegat saat melakukan perjalanan melalui jaringan antara klien dan server cache, yang dikenal sebagai data dalam perjalanan.

Anda dapat mengaktifkan enkripsi saat istirahat untuk ElastiCache (Redis OSS) saat Anda membuat grup replikasi, dengan menyetel parameter ke. `AtRestEncryptionEnabled` **`true`** Ketika parameter ini diaktifkan, parameter ini mengenkripsi disk selama operasi sinkronisasi, pencadangan, dan swap, dan mengenkripsi cadangan yang disimpan di Amazon Simple Storage Service (Amazon S3). Anda tidak dapat mengaktifkan enkripsi saat istirahat pada grup replikasi yang ada. Saat Anda membuat grup replikasi, Anda dapat mengaktifkan enkripsi saat istirahat dengan dua cara berikut:

- Dengan memilih opsi Default, yang menggunakan enkripsi yang dikelola layanan saat istirahat.
- Dengan menggunakan kunci yang dikelola pelanggan dan memberikan ID kunci atau Nama Sumber Daya Amazon (ARN) dari AWS Key Management Service (AWS KMS).

Pola ini menyediakan kontrol keamanan yang memantau panggilan API dan menghasilkan EventBridge peristiwa Amazon Events pada `CreateReplicationGroup` operasi. Acara ini memanggil AWS Lambda fungsi, yang menjalankan skrip Python. Fungsi mendapatkan ID grup replikasi dari input JSON peristiwa, dan melakukan pemeriksaan berikut untuk menentukan apakah ada cluster yang tidak terenkripsi:

- Memeriksa apakah `AtRestEncryptionEnabled` kuncinya ada.

- Jika `AtRestEncryptionEnabled` ada, periksa nilainya untuk melihat apakah itu`true`.
- Jika `AtRestEncryptionEnabled` nilai disetel ke`false`, tetapkan variabel yang melacak pelanggaran dan mengirimkan pesan pelanggaran ke alamat email yang Anda berikan, dengan menggunakan notifikasi Amazon Simple Notification Service (Amazon SNS).

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.
- Bucket S3 untuk mengunggah kode Lambda yang disediakan.
- Alamat email tempat Anda ingin menerima pemberitahuan pelanggaran.
- ElastiCache logging diaktifkan, untuk akses ke semua log API.

Batasan

- Kontrol detektif ini bersifat regional dan harus digunakan di setiap Wilayah AWS yang ingin Anda pantau.
- Kontrol mendukung grup replikasi yang berjalan di cloud pribadi virtual (VPC).
- Kontrol mendukung grup replikasi yang menjalankan jenis node berikut:
  - R7g, R6gd, R6g, R5, R4, R3
  - M7g, M6g, M5, M4, M3
  - T4g, T3, T2
  - C7gN

Versi produk

- Mendukung ElastiCache (Redis OSS) versi 3.2.6 atau yang lebih baru, dan Valkey 7.2 atau yang lebih baru

Arsitektur

Arsitektur alur kerja

1. Pengguna meluncurkan grup ElastiCache replikasi melalui panggilan Konsol Manajemen AWS, the AWS Command Line Interface (AWS CLI), atau API.
2. ElastiCache menghasilkan EventBridge peristiwa saat `CreateReplicationGroup` API dipanggil.
3. EventBridge Aturan memicu dan memanggil fungsi Lambda untuk pemeriksaan kepatuhan.
4. Fungsi Lambda memproses peristiwa dan memeriksa apakah enkripsi saat istirahat diaktifkan di cluster. ElastiCache
5. Jika pelanggaran enkripsi terdeteksi, fungsi Lambda menerbitkan pesan notifikasi ke topik SNS.
6. Amazon SNS mengirimkan pemberitahuan email kepada administrator tentang pelanggaran kepatuhan enkripsi.

Otomatisasi dan skala

- Jika Anda menggunakan AWS Organizations, Anda dapat menggunakan AWS CloudFormation StackSetsuntuk menyebarkan template ini di beberapa akun yang ingin Anda pantau.

Alat

Layanan AWS

- Amazon ElastiCache memudahkan untuk mengatur, mengelola, dan menskalakan lingkungan cache dalam memori terdistribusi di AWS Cloud. Ini menyediakan cache dalam memori berkinerja tinggi, dapat diubah ukurannya, dan hemat biaya sambil menghilangkan kompleksitas yang terkait dengan penerapan dan pengelolaan lingkungan cache terdistribusi. ElastiCache bekerja dengan mesin Redis dan Memcached.
- AWS CloudFormationmembantu Anda memodelkan dan menyiapkan AWS sumber daya Anda, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Anda dapat mengelola dan menyediakan tumpukan di beberapa Akun AWS dan Wilayah AWS.
- Amazon EventBridge memberikan aliran peristiwa sistem yang mendekati real-time yang menggambarkan perubahan AWS sumber daya. EventBridge menyadari perubahan operasional saat terjadi dan mengambil tindakan korektif seperlunya, dengan mengirim pesan untuk merespons lingkungan, mengaktifkan fungsi, membuat perubahan, dan menangkap informasi negara.

- AWS Lambdaadalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya bila diperlukan dan menskalakan secara otomatis dari beberapa permintaan per hari menjadi ribuan per detik. Anda hanya membayar untuk waktu komputasi yang Anda gunakan—tidak ada biaya saat kode Anda tidak berjalan.
- Amazon SNS mengkoordinasikan dan mengelola pengiriman pesan antara penerbit dan klien, termasuk server web dan alamat email. Pelanggan menerima semua pesan yang dipublikasikan ke topik tempat mereka berlangganan, dan semua pelanggan topik menerima pesan yang sama.

Kode

Kode untuk pola ini tersedia di ElastiCache cluster GitHub Monitor Amazon untuk repositori enkripsi saat istirahat. Lihat bagian Epics untuk informasi tentang cara menggunakan file di repositori.

Praktik terbaik

Deployment

- Pastikan bahwa AWS CloudTrail mencatat panggilan ElastiCache API sebelum Anda menerapkan kontrol ini.
- Ini adalah kontrol regional; menyebarkan kontrol di setiap Wilayah AWS tempat yang Anda gunakan ElastiCache.
- Validasi solusi di dev/test lingkungan sebelum Anda menerapkannya ke produksi.

Keamanan

- Untuk meningkatkan kontrol atas kunci enkripsi, gunakan kunci KMS yang dikelola pelanggan.
- Tinjau izin AWS Identity and Access Management (IAM) untuk memastikan akses hak istimewa paling sedikit untuk peran eksekusi Lambda.
- Siapkan peringatan untuk pesan dalam antrian surat mati.

Operasi

- Tetapkan retensi log yang sesuai untuk menyeimbangkan kebutuhan kepatuhan dengan biaya.
- Setel konkurensi Lambda yang dicadangkan untuk menyesuaikan berdasarkan frekuensi pembuatan ElastiCache Anda.

- Berlangganan beberapa alamat email ke Amazon SNS untuk pemberitahuan tim.

Pemantauan

- Tinjau CloudWatch alarm Amazon untuk memastikan bahwa ambang alarm sesuai dengan kebutuhan operasional Anda.
- Pantau durasi eksekusi metrik Lambda dan tingkat kesalahan secara teratur.
- Audit pelanggaran secara teratur untuk meninjau pemberitahuan kepatuhan enkripsi.

Epik

Menyebarkan kontrol keamanan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh kode dari GitHub.	Kloning atau unduh repositori kode dari. GitHub Repositori berisi file dan file`index.py`. `elasticache_encryption_at_rest.yml`	Arsitek awan
Buat paket penyebaran Lambda.	Buat dua file.zip dari kode Python: • Buat`ElastiCache-EncryptionAtRest.zip` , yang berisi`index.py`. • Gunakan perintah berikut ini. `zip ElastiCache-EncryptionAtRest.zip index.py`	Arsitek awan
Unggah kode ke bucket S3.	1. Buat bucket S3 baru atau gunakan bucket S3 yang	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	ada untuk mengunggah kode Lambda. 2. Zip kode Lambda (`index.py`) dan beri nama. `ElastiCache-EncryptionAtRest.zip` 3. Unggah file.zip ke bucket S3. Bucket ini harus Wilayah AWS sama dengan sumber daya yang ingin Anda evaluasi.	
Menyebarkan CloudFormation template.	Buka CloudFormation konsol Wilayah AWS sama dengan bucket S3, dan terapkan `elasticache_encryption_at_rest.yml` file yang disediakan di repositori kode. Dalam epik berikutnya, berikan nilai untuk parameter template.	Arsitek awan

Lengkapi parameter dalam CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan nama bucket S3.	Masukkan nama bucket S3 yang Anda buat atau pilih di epik pertama. Bucket S3 ini berisi file.zip untuk kode Lambda dan harus Wilayah AWS sama dengan CloudFormation template	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dan sumber daya yang akan dievaluasi.	
Berikan kunci S3.	Berikan lokasi file kode Lambda .zip di bucket S3 Anda, tanpa garis miring (misalnya, atau). `ElasticCache-EncryptionAtRest.zip controls/ElasticCache-EncryptionAtRest.zip`	Arsitek awan
Berikan alamat email.	Berikan alamat email aktif tempat Anda ingin menerima pemberitahuan pelanggaran.	Arsitek awan
Tentukan tingkat logging.	Tentukan tingkat logging dan verbositas. • `Info`menunjuk pesan informasi rinci tentang kemajuan aplikasi dan harus digunakan hanya untuk debugging. • `Error`menunjuk peristiwa kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. • `Warning`menunjuk situasi yang berpotensi berbahaya.	Arsitek awan

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan langganan email.	Ketika CloudFormation template berhasil digunakan , ia mengirimkan pesan berlangganan ke alamat email yang Anda berikan. Untuk menerima pemberitahuan, Anda harus mengonfirmasi langganan email ini.	Arsitek awan

Pemecahan masalah

Isu	Solusi
Fungsi Lambda tidak dipicu	Gejala: Tidak ada log masuk CloudWatch setelah Anda membuat atau memodifikasi ElastiCache cluster. Solusi: • Verifikasi status EventBridge aturan: `aws events describe-rule --name <rule-name>` • Konfirmasikan bahwa kebijakan berbasis sumber daya Lambda mengizinkan pemanggilan. EventBridge • Konfirmasikan bahwa pola EventBridge acara cocok dengan panggilan ElastiCache API.

Isu	Solusi
Tidak ada pemberitahuan email	Gejala: Fungsi Lambda berhasil berjalan, tetapi Anda tidak menerima pemberitahuan email apa pun. Solusi: • Konfirmasikan langganan Amazon SNS Anda melalui tautan konfirmasi email. • Periksa folder spam atau sampah untuk email Amazon SNS. • Verifikasi Nama Sumber Daya Amazon (ARN) untuk topik Amazon SNS di variabel lingkungan Lambda: `OUTBOUND_TOPIC_ARN` • Uji Amazon SNS secara manual: `aws sns publish --topic-arn <arn> --message "Test"`
Masalah izin	Gejala: Akses ditolak kesalahan dalam log fungsi CloudWatch Lambda. Solusi: • Verifikasi bahwa peran eksekusi Lambda memiliki izin ElastiCache baca yang diperlukan. • Konfirmasikan bahwa kebijakan AWS KMS kunci menyertakan peran eksekusi Lambda. • Pastikan bahwa kebijakan topik Amazon SNS memungkinkan Lambda untuk mempublikasikan.

Sumber daya terkait

- Buat tumpukan dari CloudFormation konsol (CloudFormation dokumentasi)
- Enkripsi AT-rest di ElastiCache (Redis OSS) (dokumentasi) ElastiCache

Pantau aktivitas pengguna root IAM

JJ Sung dan Mostefa Brougui, Amazon Web Services

Ringkasan

Setiap akun Amazon Web Services (AWS) memiliki pengguna root. Sebagai praktik keamanan terbaik untuk AWS Identity and Access Management (IAM), kami menyarankan Anda menggunakan pengguna root untuk menyelesaikan tugas yang hanya dapat dilakukan oleh pengguna root. Untuk daftar lengkapnya, lihat Tugas yang memerlukan kredensil pengguna root di Panduan Referensi Manajemen Akun AWS. Karena pengguna root memiliki akses penuh ke semua sumber daya AWS dan informasi penagihan Anda, sebaiknya Anda tidak menggunakan akun ini dan memantaunya untuk aktivitas apa pun, yang mungkin menunjukkan bahwa kredensil pengguna root telah disusupi.

Dengan menggunakan pola ini, Anda menyiapkan arsitektur berbasis peristiwa yang memantau pengguna root IAM. Pola ini menyiapkan hub-and-spoke solusi yang memantau beberapa akun AWS, akun spoke, dan memusatkan manajemen dan pelaporan dalam satu akun, akun hub.

Saat kredensi pengguna root IAM digunakan, Amazon dan CloudWatch AWS CloudTrail masing-masing merekam aktivitas di log dan jejak. Di akun spoke, EventBridge aturan Amazon mengirimkan acara ke bus acara pusat di akun hub. Di akun hub, EventBridge aturan mengirimkan acara ke fungsi AWS Lambda. Fungsi ini menggunakan topik Amazon Simple Notification Service (Amazon SNS) yang memberi tahu Anda tentang aktivitas pengguna root.

Dalam pola ini, Anda menggunakan CloudFormation templat AWS untuk menerapkan layanan pemantauan dan penanganan peristiwa di akun spoke. Anda menggunakan templat HashiCorp Terraform untuk menerapkan manajemen acara dan layanan notifikasi di akun hub.

Prasyarat dan batasan

Prasyarat

1. Izin untuk menerapkan sumber daya AWS di lingkungan AWS Anda.
2. Izin untuk menyebarkan set CloudFormation tumpukan. Untuk informasi selengkapnya, lihat Prasyarat untuk operasi set tumpukan (dokumentasi). CloudFormation
3. Terraform diinstal dan siap digunakan. Untuk informasi selengkapnya, lihat Memulai — AWS (dokumentasi Terraform).
4. Jejak yang ada di setiap akun bicara. Untuk informasi selengkapnya, lihat Memulai AWS CloudTrail (CloudTrail dokumentasi).

5. Jejak dikonfigurasi untuk mengirim acara ke CloudWatch Log. Untuk informasi selengkapnya, lihat Mengirim peristiwa ke CloudWatch Log (CloudTrail dokumentasi).
6. Akun hub dan spoke Anda harus dikelola oleh AWS Organizations.

Arsitektur

Diagram berikut menggambarkan blok bangunan implementasi.

1. Saat kredensil pengguna root IAM digunakan, CloudWatch dan CloudTrail rekam aktivitas di log dan jejak, masing-masing.
2. Di akun spoke, EventBridge aturan mengirimkan acara ke bus acara pusat di akun hub.
3. Di akun hub, EventBridge aturan mengirimkan acara ke fungsi Lambda.
4. Fungsi Lambda menggunakan topik Amazon SNS yang memberi tahu Anda tentang aktivitas pengguna root.

Alat

Layanan AWS

- AWS CloudFormation membantu Anda menyiapkan sumber daya AWS, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya di seluruh akun dan Wilayah AWS.
- AWS CloudTrail membantu Anda mengaudit tata kelola, kepatuhan, dan risiko operasional akun AWS Anda.
- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, dan layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, fungsi AWS Lambda, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di akun AWS lainnya.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.

- AWS Lambda adalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.

Alat dan layanan lainnya

- Terraform adalah aplikasi CLI untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud dengan menggunakan kode, dalam bentuk file konfigurasi.

Repositori kode

Kode sumber dan template untuk pola ini tersedia di GitHub repositori. Pola ini menyediakan dua template:

- Template Terraform yang berisi sumber daya yang Anda terapkan di akun hub
- CloudFormation Template yang Anda terapkan sebagai instance set tumpukan di akun spoke

Repositori memiliki struktur keseluruhan berikut.

```
.
 |__README.md
 |__spoke-stackset.yaml
 |__hub.tf
 |__root-activity-monitor-module
     |__main.tf  # contains Terraform code to deploy resources in the Hub account
     |__iam      # contains IAM policies JSON files
         |__ lambda-assume-policy.json          # contains trust policy of the IAM role
 used by the Lambda function
         |__ lambda-policy.json                 # contains the IAM policy attached to
 the IAM role used by the Lambda function
     |__outputs  # contains Lambda function zip code
```

Bagian Epics memberikan step-by-step instruksi untuk menyebarkan template.

Epik

Menyebarkan sumber daya ke akun hub

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode sampel.	1. Buka repositori AWS IAM Root Activity Monitor. 2. Pada tab Kode, di atas daftar file, pilih Kode, lalu salin URL HTTPS. 3. Di antarmuka baris perintah, ubah direktori kerja Anda ke lokasi tempat Anda ingin menyimpan file sampel. 4. Masukkan perintah berikut: `git clone <repoURL>`	AWS Umum
Perbarui template Terraform.	1. Ambil ID organisasi Anda. Untuk petunjuknya, lihat Melihat detail organisasi dari akun manajemen (dokumentasi AWS Organizations). 2. Di repositori kloning, buka. `hub.tf` 3. Perbarui yang berikut ini dengan nilai yang sesuai untuk lingkungan Anda: • `OrganizationId` — Tambahkan ID organisasi Anda.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `SNSTopicName` — Tambahkan nama untuk topik Amazon SNS. • `SNSSubscriptions` — Tambahkan email ke mana notifikasi Amazon SNS harus dikirim. • `Region`— Tambahkan kode Wilayah AWS tempat Anda menggunakan sumber daya. Misalnya, `eu-west-1` . • `Tags`— Tambahkan tag Anda. Untuk informasi selengkapnya, lihat Menandai sumber daya AWS (AWS General Reference). 4. Simpan dan tutup file `hub.tf`.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Menerapkan sumber daya ke akun hub AWS.	1. Di antarmuka baris perintah Terraform, arahkan ke folder root dari repositor i kloning, lalu masukkan perintah berikut.
	`terraform init && terraform plan`
	2. Tinjau output dan konfirmas ikan bahwa Anda ingin membuat sumber daya yang dijelaskan.
	3. Masukkan perintah berikut.
	`terraform apply`
	4. Saat diminta, konfirmas ikan penyebaran dengan memasukkan. `yes`
		AWS Umum

Menerapkan sumber daya ke akun bicara Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	1. Masuk ke AWS Management Console, dan buka CloudFormation konsol. 2. Dari panel navigasi, pilih StackSets. 3. Di bagian atas StackSets halaman, pilih Buat StackSet.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	4. Di bawah Izin, pilih Izin yang dikelola layanan. CloudFormation secara otomatis mengonfigurasi izin yang diperlukan untuk menerapkan ke akun target yang dikelola oleh AWS Organizations. 5. Di bawah Prasyarat - Siapkan templat, pilih Template sudah siap. 6. Di bawah Tentukan Templat, pilih Unggah file templat. 7. Pilih Pilih file, dan kemudian di repositori kloning, pilih. `spoke-stackset.yaml` 8. Pilih Berikutnya. 9. Pada halaman Tentukan StackSet detail, masukkan nama untuk kumpulan tumpukan. 10. Di bawah Parameter, masukkan ID akun akun hub, lalu pilih Berikutnya. 11. Pada halaman Konfigurasi StackSet opsi, di bawah Tag, tambahkan tag Anda. 12. Di bawah Konfigurasi eksekusi, pilih Tidak Aktif, lalu pilih Berikutnya.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	13. Pada halaman Setel opsi penerapan, tentukan unit organisasi dan Wilayah tempat Anda ingin menerapkan kumpulan tumpukan, lalu pilih Berikutnya. 14. Pada halaman Tinjauan, pilih Saya mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM, lalu pilih Kirim. CloudFormation mulai menerapkan set tumpukan Anda. Untuk informasi dan petunjuk selengkapnya, lihat Membuat kumpulan tumpukan (CloudFormation dokumentasi).	

(Opsional) Uji notifikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Gunakan kredensil pengguna root.	1. Masuk ke akun spoke atau akun hub dengan menggunakan kredensi pengguna root. 2. Konfirmasikan bahwa akun email yang Anda tentukan	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	menerima pemberitahuan Amazon SNS.	

Sumber daya terkait

- Praktik terbaik keamanan (dokumentasi IAM)
- Bekerja dengan StackSets (CloudFormation dokumentasi)
- Memulai (dokumentasi Terraform)

Informasi tambahan

Amazon GuardDuty adalah layanan pemantauan keamanan berkelanjutan yang menganalisis dan memproses log untuk mengidentifikasi aktivitas tak terduga dan berpotensi tidak sah di lingkungan AWS Anda. Sebagai alternatif untuk solusi ini, jika Anda telah mengaktifkan GuardDuty, itu dapat mengingatkan Anda ketika kredenial pengguna root telah digunakan. GuardDuty Temuannya adalah`Policy:IAMUser/RootCredentialUsage`, dan tingkat keparahan defaultnya Rendah. Untuk informasi selengkapnya, lihat Mengelola GuardDuty temuan Amazon.

Kirim pemberitahuan saat pengguna IAM dibuat

Mansi Suratwala dan Sergiy Shevchenko, Amazon Web Services

Ringkasan

Di Amazon Web Services (AWS), Anda dapat menggunakan pola ini untuk menerapkan CloudFormation template AWS agar menerima notifikasi secara otomatis saat pengguna AWS Identity and Access Management (IAM) dibuat.

Dengan menggunakan IAM, Anda dapat mengelola akses ke layanan dan sumber daya AWS dengan aman. Anda dapat membuat dan mengelola pengguna dan grup AWS, serta menggunakan izin untuk mengizinkan dan menolak akses pengguna dan grup tersebut ke sumber daya AWS.

CloudFormation Template membuat CloudWatch acara Amazon Events dan fungsi AWS Lambda. Acara ini menggunakan AWS CloudTrail untuk memantau setiap pengguna IAM yang dibuat di akun AWS. Jika pengguna dibuat, CloudWatch acara Acara akan memulai fungsi Lambda, yang mengirimkan pemberitahuan Amazon Simple Notification Service (Amazon SNS) yang memberi tahu Anda tentang peristiwa pembuatan pengguna baru.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif
- CloudTrail Jejak AWS dibuat dan digunakan

Batasan

- CloudFormation Template AWS harus digunakan `CreateUser` hanya untuk.

Arsitektur

Tumpukan teknologi target

- IAM
- AWS CloudTrail

- CloudWatch Acara Amazon
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon SNS

Arsitektur target

Otomatisasi dan skala

Anda dapat menggunakan CloudFormation template AWS beberapa kali untuk Wilayah dan akun AWS yang berbeda. Anda hanya perlu menjalankannya sekali di setiap Wilayah atau akun. Untuk mengotomatiskan penerapan ke beberapa akun, gunakan AWS. CloudFormation StackSets CloudFormation Template akan dapat menyebarkan semua sumber daya yang diperlukan di setiap akun.

Alat

Alat

- IAM — AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke sumber daya AWS dengan aman. Anda menggunakan IAM untuk mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan sumber daya.
- AWS CloudFormation — AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya Amazon Web Services sehingga Anda dapat menghabiskan lebih sedikit waktu untuk mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi yang berjalan di AWS. Anda membuat templat yang menjelaskan semua sumber daya AWS yang Anda inginkan, serta CloudFormation menangani penyediaan dan konfigurasi sumber daya tersebut untuk Anda.
- AWS CloudTrail — AWS CloudTrail membantu Anda mengelola tata kelola, kepatuhan, serta audit operasional dan risiko akun AWS Anda. Tindakan yang diambil oleh pengguna, peran, atau layanan AWS dicatat sebagai peristiwa di CloudTrail. Peristiwa mencakup tindakan yang diambil di AWS Management Console, AWS Command Line Interface, dan AWS SDKs dan APIs.
- CloudWatch Acara Amazon — CloudWatch Acara Amazon menghadirkan near-real-time aliran peristiwa sistem yang menjelaskan perubahan sumber daya AWS.

- AWS Lambda — AWS Lambda adalah layanan komputasi yang mendukung menjalankan kode tanpa menyediakan atau mengelola server. Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah penyimpanan untuk internet. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web.
- Amazon SNS - Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan menggunakan Lambda, HTTP, email, notifikasi push seluler, dan pesan teks seluler (SMS).

Kode

File.zip proyek tersedia sebagai lampiran.

Epik

Buat bucket S3 untuk skrip Lambda

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan bucket S3.	Buka konsol Amazon S3, dan pilih atau buat bucket S3. Bucket S3 ini akan meng-host file kode Lambda .zip. Nama bucket S3 tidak dapat berisi garis miring di depan.	Arsitek awan

Unggah kode Lambda ke bucket S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unggah kode Lambda.	Unggah file kode Lambda .zip yang disediakan di bagian Lampiran ke bucket S3 yang Anda tentukan.	Arsitek awan

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation template.	Di CloudFormation konsol, terapkan CloudFormation `createIAMuser.yaml` template yang disediakan sebagai lampiran ke pola ini. Dalam epik berikutnya, berikan nilai untuk parameter template.	Arsitek awan

Lengkapi parameter dalam CloudFormation template

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Berikan nama bucket S3.	Masukkan nama bucket S3 yang Anda buat atau pilih di epik pertama.	Arsitek awan
Berikan kunci S3.	Berikan lokasi file kode Lambda .zip di bucket S3 Anda, tanpa garis miring di depan (misalnya,). `<directory>/<file-name>.zip`	Arsitek awan
Berikan alamat email.	Berikan alamat email aktif untuk menerima notifikasi Amazon SNS.	Arsitek awan
Tentukan tingkat logging.	Tentukan tingkat logging dan frekuensi untuk fungsi Lambda Anda. `Info`menunjuk pesan informasi rinci tentang kemajuan aplikasi.	Arsitek awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`Error`menunjuk peristiwa kesalahan yang masih memungkinkan aplikasi untuk terus berjalan. `Warning`menunjuk situasi yang berpotensi berbahaya.	

Konfirmasi langganan.

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasi langganan.	Ketika template berhasil digunakan, ia mengirimkan pesan email berlangganan ke alamat email yang disediaka n. Untuk menerima pemberita huan, Anda harus mengonfir masi langganan email ini.	Arsitek awan

Sumber daya terkait

- Membuat jejak
- Membuat bucket S3
- Mengunggah file ke bucket S3
- Menerapkan template CloudFormation
- Membuat pengguna IAM
- Membuat aturan CloudWatch Acara yang dipicu pada panggilan AWS API menggunakan AWS CloudTrail

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Mencegah akses internet di tingkat akun dengan menggunakan kebijakan kontrol layanan

Sergiy Shevchenko, Sean O'Sullivan, dan Victor Mazeo Whitaker, Amazon Web Services

Ringkasan

Organizations sering ingin membatasi akses internet untuk sumber daya akun yang harus tetap pribadi. Dalam akun ini, sumber daya di awan pribadi virtual (VPCs) tidak boleh mengakses internet dengan cara apa pun. Banyak organisasi memilih arsitektur inspeksi terpusat. Untuk lalu lintas timur-barat (VPC-to-VPC) dalam arsitektur inspeksi terpusat, Anda perlu memastikan bahwa akun spoke dan sumber dayanya tidak memiliki akses ke internet. Untuk lalu lintas utara-selatan (jalan keluar internet dan lokal), Anda ingin mengizinkan akses internet hanya melalui VPC inspeksi.

Pola ini menggunakan kebijakan kontrol layanan (SCP) untuk membantu mencegah akses internet. Anda dapat menerapkan SCP ini di tingkat akun atau unit organisasi (OU). SCP membatasi konektivitas internet dengan mencegah hal-hal berikut:

- Membuat atau melampirkan gateway IPv4 atau IPv6 internet yang memungkinkan akses internet langsung ke VPC
- Membuat atau menerima koneksi peering VPC yang memungkinkan akses internet tidak langsung melalui VPC lain
- Membuat atau memperbarui AWS Global Acceleratorkonfigurasi yang memungkinkan akses internet langsung ke sumber daya VPC

Prasyarat dan batasan

Prasyarat

- Satu atau beberapa Akun AWS dikelola sebagai organisasi di AWS Organizations.
- Semua fitur diaktifkan di AWS Organizations.
- SCPs diaktifkan dalam organisasi.
- Izin untuk:
  - Akses akun manajemen organisasi.
  - Buat SCPs. Untuk informasi selengkapnya tentang izin minimum, lihat Membuat SCP.

- Lampirkan SCP ke akun target atau unit organisasi (OUs). Untuk informasi selengkapnya tentang izin minimum, lihat Melampirkan dan melepaskan kebijakan kontrol layanan.

Batasan

- SCPs tidak memengaruhi pengguna atau peran di akun manajemen. SCP tersebut hanya mempengaruhi akun anggota di organisasi Anda.
- SCPs hanya mempengaruhi AWS Identity and Access Management (IAM) pengguna dan peran yang dikelola oleh akun yang merupakan bagian dari organisasi. Untuk informasi selengkapnya, lihat efek SCP pada izin.

Alat

Layanan AWS

- AWS Organizationsadalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat. Dalam pola ini, Anda menggunakan kebijakan kontrol layanan (SCPs) di AWS Organizations.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Praktik terbaik

Setelah membuat SCP ini di organisasi Anda, pastikan untuk sering memperbaruinya untuk mengatasi fitur baru Layanan AWS atau yang mungkin memengaruhi akses internet.

Epik

Buat dan lampirkan SCP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat SCP.	1. Masuk ke konsol AWS Organizations tersebut.	Administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda harus masuk ke akun manajemen organisasi. 2. Di panel kiri, pilih Kebijakan. 3. Pada halaman kebijakan , pilih Kebijakan kontrol layanan. 4. Pada halaman Kebijakan kontrol layanan, pilih Buat kebijakan. 5. Pada halaman Buat kebijakan kontrol layanan baru, masukkan nama Kebijakan dan deskripsi Kebijakan opsional. 6. (Opsional) Tambahkan AWS tag ke kebijakan Anda. 7. Di editor JSON, hapus kebijakan placeholder. 8. Tempelkan kebijakan berikut ke editor JSON. `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "ec2:AttachInternetGateway", "ec2:CreateInternetGateway", "ec2:CreateVpcPeeringConnection",`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>"ec2:Acce
	ptVpcPeeringConnec
	tion",
	"ec2:Crea
	teEgressOnlyIntern
	etGateway"
	],
	"Resource":
	"*",
	"Effect": "Deny"
	},
	{
	"Action": [
	"globalac
	celerator:Create*",
	"globalac
	celerator:Update*"
	],
	"Resource":
	"*",
	"Effect": "Deny"
	}
	]
	}</pre> 9. Pilih Buat kebijakan.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Lampirkan SCP.	1. Pada halaman Kebijakan kontrol layanan, pilih kebijakan yang Anda buat. 2. Di tab Target, pilih Lampirkan. 3. Pilih OU atau akun yang ingin Anda lampirkan kebijakan. Anda mungkin harus memperluas OUs untuk menemukan OU atau akun yang Anda inginkan. 4. Pilih Lampirkan kebijakan.	Administrator AWS

Sumber daya terkait

- AWS Organizations dokumentasi
- Kebijakan kontrol layanan (SCPs)
- Arsitektur inspeksi terpusat dengan AWS Gateway Load Balancer AWS Transit Gateway dan AWS (posting blog)

Mengekspor laporan identitas AWS IAM Identity Center dan tugasnya dengan menggunakan PowerShell

Jorge Pava, Frank Allotta, Manideep Reddy Gillela, dan Chad Miles, Amazon Web Services

Ringkasan

Bila Anda menggunakan AWS IAM Identity Center (penerus AWS Single Sign-On) untuk mengelola akses masuk tunggal (SSO) secara terpusat ke semua akun Amazon Web Services (AWS) dan aplikasi cloud Anda, melaporkan dan mengaudit tugas tersebut melalui AWS Management Console dapat membosankan dan memakan waktu. Hal ini terutama berlaku jika Anda melaporkan izin untuk pengguna atau grup di lusinan atau ratusan akun AWS.

Bagi banyak orang, alat yang ideal untuk melihat informasi ini adalah dalam aplikasi spreadsheet, seperti Microsoft Excel. Ini dapat membantu Anda memfilter, mencari, dan memvisualisasikan data untuk seluruh organisasi Anda, yang dikelola oleh AWS Organizations.

Pola ini menjelaskan cara menggunakan AWS Tools PowerShell untuk menghasilkan laporan konfigurasi identitas SSO di IAM Identity Center. Laporan diformat sebagai file CSV, dan mencakup nama identitas (prinsipal), tipe identitas (pengguna atau grup), akun yang dapat diakses identitas, dan set izin. Setelah membuat laporan ini, Anda dapat membukanya di aplikasi pilihan Anda untuk mencari, memfilter, dan mengaudit data sesuai kebutuhan. Gambar berikut menunjukkan data sampel dalam aplikasi spreadsheet.

Penting: Karena laporan ini berisi informasi sensitif, kami sangat menyarankan Anda menyimpan nya dengan aman dan membagikannya hanya atas need-to-know dasar.

Prasyarat dan batasan

Prasyarat

- IAM Identity Center dan AWS Organizations, dikonfigurasi dan diaktifkan.
- PowerShell, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal PowerShell (dokumentasi Microsoft).
- AWS Tools untuk PowerShell, diinstal, dan dikonfigurasi. Untuk alasan kinerja, kami sangat menyarankan Anda menginstal AWS Tools versi termodulasi untuk PowerShell, yang

disebut. `AWS.Tools` Setiap layanan AWS didukung oleh modul kecil individualnya sendiri. Di PowerShell shell, masukkan perintah berikut untuk menginstal modul yang diperlukan untuk pola ini:`AWS.Tools.Installer`,`Organizations`,`SSOAdmin`, dan`IdentityStore`.

```
Install-Module AWS.Tools.Installer
Install-AWSToolsModule -Name Organizations, SSOAdmin, IdentityStore
```

Untuk informasi selengkapnya, lihat Menginstal AWS.tools di Windows atau Menginstal AWS.tools di Linux atau macOS (AWS Tools untuk dokumentasi). PowerShell Jika Anda menerima kesalahan saat menginstal modul, lihat bagian Pemecahan Masalah dari pola ini.

- AWS Command Line Interface (AWS CLI) atau AWS SDK harus dikonfigurasi sebelumnya dengan kredensyal yang berfungsi dengan melakukan salah satu hal berikut:
  - Gunakan AWS CLI `aws configure` Untuk informasi selengkapnya, lihat Konfigurasi cepat (dokumentasi AWS CLI).
  - Konfigurasikan AWS CLI atau AWS Cloud Development Kit (AWS CDK) untuk mendapatkan akses sementara melalui peran AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat Mendapatkan kredensyal peran IAM untuk akses CLI (dokumentasi Pusat Identitas IAM).
- Profil bernama untuk AWS CLI yang telah menyimpan kredensyal untuk prinsipal IAM yang:
  - Memiliki akses ke akun manajemen AWS Organizations atau akun administrator yang didelegasikan untuk IAM Identity Center
  - Apakah kebijakan terkelola `AWSSSOReadOnly` dan `AWSSSODirectoryReadOnly` AWS telah diterapkan padanya

  Untuk informasi selengkapnya, lihat Menggunakan profil bernama (dokumentasi AWS CLI) dan kebijakan terkelola AWS (dokumentasi IAM).

Batasan

- Akun AWS target harus dikelola sebagai organisasi di AWS Organizations.

Versi produk

- Untuk semua sistem operasi, disarankan agar Anda menggunakan PowerShell versi 7.0 atau yang lebih baru.

Arsitektur

Arsitektur target

1. Pengguna menjalankan skrip dalam PowerShell baris perintah.
2. Skrip mengasumsikan profil bernama untuk AWS CLI. Ini memberikan akses ke Pusat Identitas IAM.
3. Skrip mengambil konfigurasi identitas SSO dari IAM Identity Center.
4. Skrip menghasilkan file CSV di direktori yang sama di workstation lokal tempat skrip disimpan.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS IAM Identity Center membantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua akun AWS dan aplikasi cloud Anda.
- AWS Tools for PowerShell adalah seperangkat PowerShell modul yang membantu Anda menjalankan skrip pada sumber daya AWS Anda dari baris PowerShell perintah.

Alat-alat lainnya

- PowerShelladalah program manajemen otomatisasi dan konfigurasi Microsoft yang berjalan di Windows, Linux, dan macOS.

Epik

Hasilkan laporan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Siapkan naskahnya.	1. Salin PowerShell skrip di bagian Informasi tambahan dari pola ini.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	2. Di `Param` bagian ini, untuk lingkungan AWS Anda, tentukan nilai untuk variabel berikut: • `OutputFile` — Nama file laporan. • `ProfileName` — Profil bernama AWS CLI yang ingin Anda gunakan untuk menghasilkan laporan. • `Region`— Wilayah AWS tempat Pusat Identitas IAM digunakan. Untuk daftar lengkap Wilayah dan kodenya, lihat Titik akhir Regional. 3. Simpan skrip dengan nama file`SSO-Report.ps1` .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Jalankan penulisan.	Disarankan agar Anda menjalankan skrip kustom Anda di PowerShell shell dengan perintah berikut.
	`.\SSO-Report.ps1`
	Atau, Anda dapat menjalank an skrip dari shell lain dengan memasukkan perintah berikut.
	`pwsh .\SSO-Report.ps1`
	Skrip menghasilkan file CSV di direktori yang sama dengan file skrip.
		Administrator awan

Menganalisis data laporan.	File CSV keluaran memiliki header AccountName,, Principal PermissionSet, dan Type. Buka file ini di aplikasi spreadsheet pilihan Anda. Anda dapat membuat tabel data untuk memfilter dan mengurutkan output.	Administrator awan

Pemecahan masalah

Isu	Solusi
`The term 'Get-<parameter>' is not recognized as the name of a cmdlet, function, script file, or operable program.` kesalahan	AWS Tools untuk PowerShell atau modulnya tidak diinstal. Di PowerShell shell, masukkan perintah berikut untuk menginstal AWS Tools PowerShell dan modul yang diperluka

Isu	Solusi

	n untuk pola ini:`AWS.Tools.Installer` ,`Organizations` ,`SSOAdmin`, dan`IdentityStore` .
	
	`Install-Module AWS.Tools.Installer`
	`Install-AWSToolsModule -Name Organizations, SSOAdmin, IdentityStore`

`No credentials specified or obtained from persisted/shell defaults` kesalahan	Di Siapkan skrip di bagian Epik, konfirmasikan bahwa Anda telah memasukkan `Region` variabel `ProfileName` dan dengan benar. Pastikan bahwa pengaturan dan kredensyal di profil bernama memiliki izin yang cukup untuk mengelola Pusat Identitas IAM.
`Authenticode Issuer …` kesalahan saat menginstal modul AWS.tools	Tambahkan `-SkipPublisherCheck` parameter ke akhir `Install-AWSToolsModule` perintah.

Isu	Solusi

`Get-ORGAccountList : Assembly AWSSDK.SSO could not be found or loaded.` kesalahan	Kesalahan ini dapat terjadi ketika profil AWS CLI bernama ditentukan, AWS CLI dikonfigu rasi untuk mengautentikasi pengguna dengan IAM Identity Center, dan AWS CLI dikonfigu rasi untuk secara otomatis mengambil token otentikasi yang diperbarui. Untuk mengatasi kesalahan ini, lakukan hal berikut:
	1. Masukkan perintah berikut untuk mengonfir masi bahwa `SSOOIDC` modul `SSO` dan telah diinstal.
	`Install-AWSToolsModule SSO, SSOOIDC`
	2. Masukkan baris berikut ke dalam skrip di bawah `param()` blok.
	`Import-Module AWS.Tools.SSO`
	`Import-Module AWS.Tools.SSOOIDC`

Sumber daya terkait

- Di mana pengaturan konfigurasi disimpan? (Dokumentasi AWS CLI)
- Mengonfigurasi AWS CLI untuk menggunakan AWS IAM Identity Center (dokumentasi AWS CLI)
- Menggunakan profil bernama (dokumentasi AWS CLI)

Informasi tambahan

Dalam skrip berikut, tentukan apakah Anda perlu memperbarui nilai untuk parameter berikut:

- Jika Anda menggunakan profil bernama di AWS CLI untuk mengakses akun tempat Pusat Identitas IAM dikonfigurasi, perbarui nilainya. `$ProfileName`

- Jika Pusat Identitas IAM diterapkan di Wilayah AWS yang berbeda dari Wilayah default untuk konfigurasi AWS CLI atau AWS SDK Anda, perbarui `$Region` nilainya untuk menggunakan Wilayah tempat Pusat Identitas IAM diterapkan.
- Jika tidak satu pun dari situasi ini berlaku, maka tidak diperlukan pembaruan skrip.

```
param (
    # The name of the output CSV file
    [String] $OutputFile  = "SSO-Assignments.csv",
    # The AWS CLI named profile
    [String] $ProfileName = "",
    # The AWS Region in which IAM Identity Center is configured
    [String] $Region      = ""
)
$Start = Get-Date; $OrgParams = @{}
If ($Region){ $OrgParams.Region = $Region}
if ($ProfileName){$OrgParams.ProfileName = $ProfileName}
$SSOParams   = $OrgParams.Clone(); $IdsParams = $OrgParams.Clone()
$AccountList = Get-ORGAccountList @OrgParams | Select-Object Id, Name
$SSOinstance = Get-SSOADMNInstanceList @OrgParams
$SSOParams['InstanceArn']       = $SSOinstance.InstanceArn
$IdsParams['IdentityStoreId']   = $SSOinstance.IdentityStoreId
$PSsets       = @{}; $Principals   = @{}
$Assignments  = @(); $AccountCount = 1; Write-Host ""
foreach ($Account in $AccountList) {
    $Duration = New-Timespan -Start $Start -End (Get-Date) | ForEach-Object
 {[Timespan]::New($_.Days, $_.Hours, $_.Minutes, $_.Seconds)}
    Write-Host "`r$Duration - Account $AccountCount of $($AccountList.Count)
 (Assignments:$($Assignments.Count))        " -NoNewline
    $AccountCount++
    foreach ($PS in Get-SSOADMNPermissionSetsProvisionedToAccountList -AccountId
 $Account.Id @SSOParams) {
        if (-not $PSsets[$PS]) {$PSsets[$PS] = (Get-SSOADMNPermissionSet @SSOParams -
PermissionSetArn $PS).Name;$APICalls++}
        $AssignmentsResponse = Get-SSOADMNAccountAssignmentList @SSOParams -
PermissionSetArn $PS -AccountId $Account.Id
        if ($AssignmentsResponse.NextToken) {$AccountAssignments =
 $AssignmentsResponse.AccountAssignments}
        else {$AccountAssignments = $AssignmentsResponse}
        While ($AssignmentsResponse.NextToken) {
            $AssignmentsResponse = Get-SSOADMNAccountAssignmentList @SSOParams -
PermissionSetArn $PS -AccountId $Account.Id -NextToken $AssignmentsResponse.NextToken
            $AccountAssignments += $AssignmentsResponse.AccountAssignments}
```

```
        foreach ($Assignment in $AccountAssignments) {
            if (-not $Principals[$Assignment.PrincipalId]) {
                $AssignmentType = $Assignment.PrincipalType.Value
                $Expression     = "Get-IDS"+$AssignmentType+" @IdsParams -"+
$AssignmentType+"Id "+$Assignment.PrincipalId
                $Principal      = Invoke-Expression $Expression
                if ($Assignment.PrincipalType.Value -eq "GROUP")
 { $Principals[$Assignment.PrincipalId] = $Principal.DisplayName }
                else { $Principals[$Assignment.PrincipalId] = $Principal.UserName }
            }
            $Assignments += [PSCustomObject]@{
                AccountName     = $Account.Name
                PermissionSet   = $PSsets[$PS]
                Principal       = $Principals[$Assignment.PrincipalId]
                Type            = $Assignment.PrincipalType.Value}
        }
    }
}
$Duration = New-Timespan -Start $Start -End (Get-Date) | ForEach-Object
 {[Timespan]::New($_.Days, $_.Hours, $_.Minutes, $_.Seconds)}
Write-Host "`r$($AccountList.Count) accounts done in $Duration. Outputting result to
 $OutputFile"
$Assignments | Sort-Object Account | Export-CSV -Path $OutputFile -Force
```

Membatasi akses berdasarkan alamat IP atau geolokasi dengan menggunakan AWS WAF

Louis Hourcade, Amazon Web Services

Ringkasan

AWS WAFadalah firewall aplikasi web yang membantu melindungi aplikasi web dan APIs terhadap eksploitasi web umum dan bot yang dapat memengaruhi ketersediaan, membahayakan keamanan, atau mengkonsumsi sumber daya yang berlebihan. Daftar kontrol akses web (web ACLs) AWS WAF memberi Anda kontrol atas bagaimana lalu lintas mencapai aplikasi Anda. Di ACL web, Anda menambahkan aturan atau grup aturan yang dirancang untuk mengizinkan lalu lintas yang sah, mengontrol lalu lintas bot, dan memblokir pola serangan umum. Untuk informasi selengkapnya, lihat Cara AWS WAF kerja.

Anda dapat mengaitkan jenis aturan berikut ke AWS WAF web Anda ACLs:

- Grup aturan AWS terkelola — Tim dan AWS Marketplace penjual Aturan Terkelola menawarkan set aturan yang telah dikonfigurasi sebelumnya. Beberapa grup aturan terkelola dirancang untuk membantu melindungi jenis aplikasi web tertentu. Lainnya menawarkan perlindungan luas terhadap ancaman yang diketahui atau kerentanan umum.
- Aturan kustom dan grup aturan khusus — Anda juga dapat membuat aturan dan grup aturan yang menyesuaikan akses ke aplikasi web Anda dan APIs. Misalnya, Anda dapat membatasi lalu lintas berdasarkan daftar alamat IP tertentu atau pada daftar negara.

Dengan menggunakan pola ini dan repositori kode terkait, Anda dapat menggunakan AWS Cloud Development Kit (AWS CDK)untuk menyebarkan AWS WAF web ACLs dengan aturan khusus. Aturan ini membatasi akses ke sumber daya aplikasi web berdasarkan alamat IP atau geolokasi pengguna akhir. Anda juga dapat melampirkan beberapa grup aturan terkelola secara opsional.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Izin untuk menyebarkan sumber daya AWS WAF
- AWS CDK, diinstal dan dikonfigurasi di akun Anda

- Git, diinstal

Batasan

- Anda dapat menggunakan pola ini hanya di Wilayah AWS tempat AWS WAF yang tersedia. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah.

Alat

Layanan AWS

- AWS Cloud Development Kit (AWS CDK)adalah kerangka pengembangan perangkat lunak yang membantu Anda menentukan dan menyediakan AWS Cloud infrastruktur dalam kode.
- AWS WAFadalah firewall aplikasi web yang membantu Anda memantau permintaan HTTP dan HTTPS yang diteruskan ke sumber daya aplikasi web Anda yang dilindungi.

Repositori kode

Kode untuk pola ini tersedia di GitHub IP dan pembatasan geolokasi dengan AWS WAF repositori. Kode ini menyebarkan dua AWS WAF web ACLs. Yang pertama adalah ACL web regional yang ditujukan untuk sumber daya Amazon API Gateway. Yang kedua adalah ACL web global untuk CloudFront sumber daya Amazon. Kedua web ACLs berisi aturan kustom berikut:

- `IPMatch`memblokir permintaan dari alamat IP yang tidak diizinkan.
- `GeoMatch`memblokir permintaan dari negara yang tidak diizinkan.

Selama penerapan, Anda dapat secara opsional melampirkan semua grup aturan terkelola berikut ke web Anda: ACLs

- Core rule set (CRS) - Grup aturan ini berisi aturan yang umumnya berlaku untuk aplikasi web. Ini membantu melindungi terhadap eksploitasi berbagai kerentanan, termasuk beberapa risiko tinggi dan kerentanan yang umum terjadi yang dijelaskan dalam publikasi OWASP, seperti OWASP Top 10.
- Perlindungan admin - Grup aturan ini berisi aturan yang membantu Anda memblokir akses eksternal ke halaman administratif yang terbuka.

- Masukan buruk yang diketahui - Grup aturan ini membantu memblokir pola permintaan yang diketahui tidak valid dan terkait dengan eksploitasi atau penemuan kerentanan.
- Daftar reputasi IP Amazon - Grup aturan ini berisi aturan yang didasarkan pada intelijen ancaman internal Amazon. Ini membantu Anda memblokir alamat IP yang biasanya terkait dengan bot atau ancaman lainnya.
- Grup aturan terkelola sistem operasi Linux - Grup aturan ini membantu memblokir pola permintaan yang terkait dengan eksploitasi kerentanan Linux, termasuk serangan Local File Inclusion (LFI) khusus Linux.
- Grup aturan terkelola database SQL - Grup aturan ini membantu memblokir pola permintaan yang terkait dengan eksploitasi database SQL, seperti serangan injeksi SQL.

Epik

Konfigurasikan AWS WAF web ACLs

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Masukkan perintah berikut untuk mengkloning pembatasan IP dan geolokasi dengan AWS WAF repositori ke workstation lokal Anda: `git clone https://github.com/aws-samples/ip-and-geolocation-restriction-with-waf-cdk.git`	Git
Konfigurasikan aturan.	1. Di repositori kloning, buka file app.py. 2. Ubah nilai variabel berikut untuk menyesuaikan aturan: `aws_acccount = "AWS_ACCOUNT"`	AWS Umum, Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`region = "AWS_REGION"`
	`ip_list = ["CIDR_RANGE_1", "CIDR_RANGE_2"]`
	`geo_list = ["COUNTRY_CODE_1", "COUNTRY_CODE_2"]`
	`aws_managed_rules = True`
	
	Di mana:
	• `aws_account` adalah ID target Akun AWS.
	• `region`adalah target Wilayah AWS untuk ACL web untuk sumber daya API Gateway.
	
	Note
	ACL web untuk CloudFront sumber daya bersifat global dan akan digunakan di Wilayah. `us-east-1`
	
	• `ip_list`adalah daftar rentang CIDR yang diizinkan akses.
	• `geo_list`adalah daftar negara yang diizinkan mengakses. Untuk nilai	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang valid, lihat AWS WAF dokumentasi. • `aws_managed_rules` mengontrol apakah grup aturan terkelola ditambahkan ke ACL web. Jika nilai ini`True`, mereka ditambahkan. Jika nilai ini`False`, mereka dikecualikan. 3. Simpan dan tutup file app.py.	

Bootstrap dan terapkan kode

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Bootstrap AWS lingkungan Anda.	Jika belum selesai, Anda perlu bootstrap AWS lingkungan Anda sebelum Anda dapat menyebarkan AWS CDK aplikasi. 1. Di AWS CDK CLI, masukkan perintah berikut untuk bootstrap Wilayah: `us-east-1` `cdk bootstrap aws://<account-id>/us-east-1` 2. Jika Anda menerapkan ACL web untuk API Gateway di	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Wilayah selain`us-east-1` , masukkan perintah berikut untuk mem-boots trap Wilayah target: `cdk bootstrap aws://<account-id>/<region>`	
Menyebarkan AWS CDK aplikasi.	1. Masukkan perintah berikut untuk menyebarkan AWS CDK aplikasi: `cdk deploy --all` 2. Tunggu hingga penerapan AWS CloudFormation tumpukan selesai.	AWS Umum

Validasi penerapan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan bahwa web ACLs berhasil digunakan.	1. Masuk ke Konsol Manajemen AWS, lalu buka AWS WAF konsol. 2. Di panel navigasi, pilih Web ACLs. 3. Dalam daftar Wilayah AWS, pilih Global (CloudFront). 4. Verifikasi bahwa ACL CloudFront web baru telah digunakan, dan konfirmasikan bahwa ACL	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memiliki alamat IP dan aturan geolokasi yang Anda tentukan. Nama default dari ACL web ini adalah`WebACLClo udfront-<ID>` . 5. Dalam daftar Wilayah AWS, pilih Wilayah tempat Anda menerapkan tumpukan. 6. Verifikasi bahwa ACL web baru untuk sumber daya API Gateway telah diterapkan. Konfirmasikan bahwa ia memiliki alamat IP dan aturan geolokasi yang Anda tentukan. Nama default dari ACL web ini adalah`WebACLApiGW-<ID>`.	
(Opsional) Kaitkan web ACLs ke sumber daya Anda.	Kaitkan AWS WAF web ACLs dengan AWS sumber daya Anda, seperti Application Load Balancer, API Gateway, atau CloudFront distribusi. Untuk petunjuk, lihat Mengaitkan atau memisahkan ACL web dengan sumber daya.AWS Sebagai contoh, lihat class CfnWeb ACLAssociation (construct) dalam AWS CDK dokumentasi.	AWS Umum

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus tumpukan.	1. Putuskan hubungan ACL web dari sumber daya apa pun AWS . Untuk instruksi, lihat AWS WAF dokumentasi. 2. Di AWS CDK CLI, masukkan perintah berikut untuk menghapus aplikasi. AWS CDK `cdk destroy --all`	AWS Umum

Sumber daya terkait

- Referensi API (AWS CDK dokumentasi)
- aws-cdk-lib.aws_wafv2 modul (dokumentasi)AWS CDK
- Bekerja dengan web ACLs (AWS WAF dokumentasi)
- Mengelola grup aturan Anda sendiri (AWS WAF dokumentasi)
- Aturan (AWS WAF dokumentasi)

Pindai repositori Git untuk informasi sensitif dan masalah keamanan dengan menggunakan git-secret

Saurabh Singh, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menggunakan alat git-secret open-source dari AWS Labs untuk memindai repositori sumber Git dan menemukan kode yang berpotensi menyertakan informasi sensitif, seperti kata sandi pengguna atau kunci AWS akses, atau yang memiliki masalah keamanan lainnya.

`git-secrets`memindai commit, mengkomit pesan, dan menggabungkan untuk mencegah informasi sensitif seperti rahasia ditambahkan ke repositori Git Anda. Misalnya, jika komit, pesan komit, atau komit apa pun dalam riwayat gabungan cocok dengan salah satu pola ekspresi reguler yang dikonfigurasi dan dilarang, komit akan ditolak.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Repositori Git yang membutuhkan pemindaian keamanan
- Klien Git (versi 2.37.1 dan yang lebih baru) diinstal

Arsitektur

Arsitektur target

- Git
- `git-secrets`

Alat

- git-secrets adalah alat yang mencegah Anda memasukkan informasi sensitif ke dalam repositori Git.

- Git adalah sistem kontrol versi terdistribusi open-source.

Praktik terbaik

- Selalu pindai repositori Git dengan menyertakan semua revisi:

```
git secrets --scan-history
```

Epik

Connect ke sebuah EC2 instance

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke sebuah EC2 instance dengan menggunak an SSH.	Connect ke instans Amazon Elastic Compute Cloud (Amazon EC2) menggunakan SSH dan file key pair. Anda dapat melewati langkah ini jika Anda memindai repositori di mesin lokal Anda.	AWS Umum

Instal Git

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Instal Git.	Instal Git dengan menggunak an perintah: `yum install git -y` Jika Anda menggunakan mesin lokal Anda, Anda dapat menginstal klien Git untuk versi OS tertentu. Untuk	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	informasi selengkapnya, lihat situs web Git.	

Kloning repositori sumber dan instal git-secret

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori sumber Git.	Untuk mengkloning repositori Git yang ingin Anda pindai, pilih perintah klon Git dari direktori home Anda.	AWS Umum
Kloning git-rahasia.	Kloning `git-secrets` repositori Git. `git clone https://github.com/awslabs/git-secrets.git` Tempatkan di `git-secrets` suatu tempat di Anda `PATH` sehingga Git mengambilnya saat Anda menjalankan`git-secrets .`	AWS Umum
Instal git-secret.	Untuk Unix dan varian (Linux/macOS): Anda dapat menggunakan `install` target `Makefile` (disediakan dalam `git-secrets` repositori) untuk menginstal alat. Anda dapat menyesuaikan jalur instalasi dengan menggunakan	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`MANPREFIX` variabel `PREFIX` dan.
	`make install`
	Untuk Windows:
	Jalankan PowerShell `install.ps1` skrip yang disediakan di `git-secrets` repositori. Skrip ini menyalin file instalasi ke direktori instalasi (secara `%USERPROFILE%/.git-secrets` default) dan menambahkan direktori ke pengguna saat iniPATH.
	`PS > ./install.ps1`
	Untuk Homebrew (pengguna macOS):
	Jalankan:
	`brew install git-secrets`	

Pindai repositori kode git

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Pergi ke repositori sumber.	Beralih ke direktori untuk repositori Git yang ingin Anda pindai:	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`cd my-git-repository`	
Daftarkan set aturan AWS (Git hooks).	Untuk mengonfigurasi `git-secrets` untuk memindai repositori Git Anda pada setiap komit, jalankan perintah: `git secrets --register-aws`	AWS Umum
Pindai repositori.	Jalankan perintah berikut untuk mulai memindai repositori Anda: `git secrets --scan`	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Tinjau file output.	Alat ini menghasilkan file keluaran jika menemukan kerentanan di repositori Git Anda. Contoh:
	
	`example.sh:4:AWS_SECRET_ACCESS_KEY = *********`
	
	`[ERROR] Matched one or more prohibited patterns`
	
	`Possible mitigations:`
	`- Mark false positives as allowed using: git config --add secrets.allowed ...`
	`- Mark false positives as allowed by adding regular expressions to .gitallowed at repository's root directory`
	`- List your configured patterns: git config --get-all secrets.patterns`
	`- List your configured allowed patterns: git config --get-all secrets.allowed`
	`- List your configured allowed patterns in .gitallowed at repository's root directory`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`- Use --no-verify if this is a one-time false positive`	

Sumber daya terkait

- alat git-rahasia

Amankan transfer file dengan menggunakan Transfer Family, Amazon Cognito, dan GuardDuty

Manoj Kumar, Amazon Web Services

Ringkasan

Solusi ini membantu Anda mentransfer file dengan aman melalui server SFTP dengan menggunakan. AWS Transfer Family Ini termasuk kemampuan pemindaian malware otomatis melalui Perlindungan Malware untuk S3, fitur Amazon GuardDuty. Ini dirancang untuk organisasi yang perlu bertukar file dengan aman dengan pihak eksternal dan memvalidasi bahwa semua file yang masuk dipindai untuk malware sebelum diproses.

Templat infrastruktur sebagai kode (IAc) yang disediakan dengan pola ini membantu Anda menerapkan yang berikut:

- Server SFTP yang aman dengan otentikasi Amazon Cognito melalui AWS Lambda
- Bucket Amazon Simple Storage Service (Amazon S3) untuk upload dan file masuk yang telah dipindai untuk malware
- Arsitektur berbasis virtual private cloud (VPC) dengan subnet publik dan pribadi di beberapa Availability Zone
- Kontrol akses berbasis IP untuk lalu lintas masuk dan keluar, dengan daftar izinkan dan tolak yang dapat dikonfigurasi
- Pemindaian malware otomatis melalui GuardDuty
- Perutean file cerdas berdasarkan hasil pemindaian melalui Amazon EventBridge dan Lambda
- Pemberitahuan real-time untuk insiden keamanan melalui Amazon Simple Notification Service (Amazon SNS)
- Enkripsi untuk bucket Amazon S3 dan variabel lingkungan Lambda melalui () AWS Key Management Service AWS KMS
- Titik akhir Amazon Virtual Private Cloud (Amazon VPC) untuk akses tanpa eksposur internet
- Pencatatan komprehensif melalui CloudWatch integrasi Amazon

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Izin dalam AWS Identity and Access Management (IAM) untuk melakukan tindakan yang dijelaskan dalam pola ini, termasuk menerapkan AWS CloudFormation templat yang menyediakan peran IAM
- GuardDuty, diaktifkan di akun target
- Perlindungan Malware untuk S3, diaktifkan di akun target
- Kuota layanan memungkinkan Anda untuk membuat yang berikut di akun target:
  - Satu VPC
  - Satu subnet pribadi
  - Satu subnet publik
  - Tiga alamat IP elastis
  - Batas konkurensi Lambda yang cukup
- Alamat email yang valid untuk notifikasi terkait keamanan
- (Opsional) Daftar alamat IP atau rentang CIDR yang ingin Anda izinkan atau tolak
- (Opsional) AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi

Batasan

- Perlindungan Malware untuk S3 tunduk pada kuota, seperti ukuran file maksimum. Untuk informasi selengkapnya, lihat Kuota dalam Perlindungan Malware untuk S3 dan Dukungan fitur Amazon S3 dalam dokumentasi. GuardDuty
- Solusi ini hanya menggunakan autentikasi nama pengguna dan kata sandi Amazon Cognito. Metode otentikasi berbasis sertifikat atau lainnya tidak didukung dalam template ini. Secara default, solusi ini tidak mengkonfigurasi otentikasi multi-faktor (MFA).
- Solusinya mengimplementasikan kontrol akses berbasis IP hanya melalui grup keamanan.

Arsitektur

Diagram arsitektur berikut menunjukkan sumber daya yang digunakan dalam pola ini. Solusi ini menggunakan Amazon Cognito untuk otentikasi dan otorisasi pengguna. Server AWS Transfer Family SFTP digunakan untuk upload file. File disimpan dalam bucket Amazon S3, dan Amazon GuardDuty memindai file untuk malware. Amazon SNS mengirimkan pemberitahuan email jika malware terdeteksi.

Diagram menunjukkan alur kerja berikut:

1. Seorang pengguna terhubung ke titik akhir server SFTP di. AWS Transfer Family Ini memulai proses otentikasi dengan kumpulan pengguna Amazon Cognito.
2. Fungsi Lambda memulai proses otentikasi dan otorisasi dan memvalidasi kredensi pengguna dengan Amazon Cognito.
3. Fungsi Lambda mengembalikan bucket Amazon `UploadBucket` S3 sebagai direktori home. Pengguna mengasumsikan peran IAM untuk server Transfer Family, dan fungsi Lambda memberi tahu pengguna bahwa mereka telah berhasil diautentikasi.
4. Pengguna mengunggah file ke server SFTP Transfer Family. File disimpan di bucket `UploadBucket` Amazon S3.
5. GuardDuty memindai file untuk malware. Hasil pemindaian potensial adalah`NO_THREATS_FOUND`,`THREATS_FOUND`,`UNSUPPORTED`,`ACCESS_DENIED`, dan`FAILED`. Untuk hasil sampel, lihat hasil pemindaian objek S3 dalam GuardDuty dokumentasi.
6. EventBridge Aturan mendeteksi peristiwa hasil pemindaian.
7. EventBridge memulai fungsi Lambda perutean file.
8. Fungsi Lambda memproses acara dan memfilter file berdasarkan hasil pemindaian sebagai berikut:
   - File yang memiliki hasil `NO_THREATS_FOUND` pemindaian dikirim ke bucket `CleanBucket` Amazon S3.
   - File yang memiliki hasil `THREATS_FOUND` pemindaian dikirim ke bucket `MalwareBucket` Amazon S3.
   - File yang memiliki hasil `UNSUPPORTED` pemindaian dikirim ke bucket `ErrorBucket` Amazon S3.
   - File yang memiliki hasil `ACCESS_DENIED` pemindaian dikirim ke bucket `ErrorBucket` Amazon S3.
   - File yang memiliki hasil `FAILED` pemindaian dikirim ke bucket `ErrorBucket` Amazon S3.

   Semua file dienkripsi dengan file. AWS KMS key
9. Jika file dikirim ke bucket `MalwareBucket` Amazon S3, fungsi Lambda memulai topik Amazon SNS. Topik Amazon SNS mengirimkan pemberitahuan email ke alamat email yang Anda konfigurasikan.

Alat

Layanan AWS

- Amazon CloudWatch membantu Anda memantau metrik sumber AWS daya Anda dan aplikasi yang Anda jalankan AWS secara real time.
- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data real-time dari berbagai sumber. Misalnya, AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- Amazon GuardDuty adalah layanan pemantauan keamanan berkelanjutan yang menganalisis dan memproses log untuk mengidentifikasi aktivitas tak terduga dan berpotensi tidak sah di lingkungan Anda AWS .
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS Transfer Familymembantu Anda mentransfer file masuk dan keluar dari layanan AWS penyimpanan melalui protokol SFTP, FTPS, atau FTP.
- Amazon Virtual Private Cloud (Amazon VPC) membantu Anda meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaat menggunakan infrastruktur yang dapat diskalakan. AWS

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub AWS Transfer Family dan GuardDuty Malware Scanning Solution.

Praktik terbaik

CloudFormation Template yang disediakan dirancang untuk menggabungkan banyak praktik AWS terbaik, seperti izin hak istimewa terkecil untuk peran dan kebijakan IAM, enkripsi saat istirahat dan dalam perjalanan, dan rotasi kunci otomatis. Untuk lingkungan produksi, pertimbangkan untuk menerapkan rekomendasi tambahan berikut:

- Aktifkan MFA untuk pengguna Amazon Cognito
- AWS ShieldMelaksanakan perlindungan penolakan layanan (DDoS) terdistribusi
- Konfigurasikan AWS Configuntuk pemantauan kepatuhan berkelanjutan
- Implementasikan AWS CloudTrailuntuk pencatatan API yang komprehensif
- Siapkan Amazon GuardDuty untuk deteksi ancaman di luar pemindaian malware
- Implementasikan AWS Security Hub CSPMuntuk manajemen keamanan terpusat
- Gunakan AWS Secrets Manageruntuk manajemen kredensi
- Menerapkan pemantauan lalu lintas jaringan dengan Traffic Mirroring
- Konfigurasikan Amazon Macie untuk penemuan dan perlindungan data sensitif di Amazon S3
- Menerapkan penilaian keamanan reguler dan pengujian penetrasi
- Menetapkan rencana respon insiden formal
- Menerapkan penambalan otomatis untuk semua komponen
- Melakukan pelatihan keamanan reguler untuk administrator
- Siapkan AWS Organizationsuntuk manajemen keamanan multi-akun

Epik

Menyebarkan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Masukkan perintah berikut untuk mengkloning repositori solusi pemindaian GuardDuty malware ke workstation lokal Anda:AWS Transfer Family	Pengembang aplikasi, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`git clone https://github.com/aws-samples/sample-secure-transfer-family-code.git`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat CloudFormation tumpukan.	1. Ikuti petunjuk dalam CloudFormation dokumentasi untuk membuat tumpukan melalui CloudFormation konsol atau melalui file AWS CLI. Konfigurasikan parameter berikut: • `UploadBucketName` - Unggah nama bucket • `CleanBucketName` - Bersihkan nama bucket file • `MalwareBucketName` - Nama ember file malware • `ErrorBucketName` - Kesalahan file nama bucket • `SecurityTeamEmail` - Alamat email untuk peringatan • `VpcCIDR`- Rentang CIDR untuk VPC • `PublicSubnet1CIDR` - Rentang CIDR untuk subnet publik 1 • `PublicSubnet2CIDR` - Rentang CIDR untuk subnet publik 2 • `PrivateSubnet1CIDR` - Rentang CIDR untuk subnet pribadi 1	Administrator cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `PrivateSubnet2CIDR` - Rentang CIDR untuk subnet pribadi 2 • `AllowedIPAddresses` - Alamat IP yang diizinkan atau rentang CIDR • `EnableEgressRules` — Jika Anda ingin mengaktifkan aturan jalan keluar untuk grup keamanan, masukkan. `true` Jika Anda ingin menonaktifkan aturan jalan keluar, masukkan. `false` 2. Tunggu hingga penerapan tumpukan selesai. 3. Dalam output pembuatan tumpukan, catat titik akhir server SFTP. Anda membutuhkan nilai ini nanti.	

Konfigurasikan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan perlindungan malware.	1. Masuk ke Konsol Manajemen AWS dan buka GuardDuty konsol. 2. Di panel navigasi, pilih Perlindungan Malware untuk S3.	Administrator cloud, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	3. Di bagian Ember yang dilindungi, pilih Aktifkan. 4. Di bawah Masukkan detail bucket S3, masukkan nama bucket Amazon S3. Atau, pilih Browse S3 untuk memilih bucket. Pilih ember `UploadBucket` Amazon S3. 5. Di bawah Awalan, pilih Semua objek di bucket S3. 6. Di bawah Tag objek yang dipindai, pilih objek Tag. 7. Di bagian Akses layanan, pilih Buat dan gunakan peran layanan baru. 8. Pilih Aktifkan.	
Tambahkan pengguna ke kumpulan pengguna.	Tambahkan satu atau beberapa pengguna ke kumpulan pengguna Amazon Cognito. Untuk petunjuknya, lihat Mengelola pengguna di kumpulan pengguna Anda di dokumentasi Amazon Cognito.	Administrator cloud, administrator AWS

Uji server SFTP

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Connect ke endpoint server SFTP.	1. Masukkan perintah berikut untuk mendapatkan endpoint server SFTP, di mana nama `<stack-`	Pengembang aplikasi, administrator Cloud, arsitek Cloud, DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	name> tumpukan Anda: CloudFormation
	
	`TRANSFER_ENDPOINT=$(aws cloudformation describe-stacks \ --stack-name <stack-name> \ --query 'Stacks[0].Outputs[?OutputKey==`TransferServerEndpoint`].OutputValue' \ --output text)`
	
	2. Masukkan perintah berikut untuk terhubung ke server SFTP:
	
	`sftp user@company.com@$TRANSFER_ENDPOINT`
	
	3. Validasi bahwa Anda menerima prompt berikut:
	
	`sftp>`
	
	Ini menunjukkan bahwa Anda sekarang dapat memasukkan perintah untuk melakukan transfer file.	

Pemecahan masalah

Isu	Solusi
Otentikasi pengguna gagal	1. Verifikasi bahwa pengguna ada di kumpulan pengguna Amazon Cognito. 2. Validasi bahwa pengguna dikonfirmasi dan diaktifkan. 3. Pastikan pengguna memberikan kata sandi yang benar. 4. Periksa kesalahan di CloudWatch Log. Untuk daftar AWS CLI perintah yang dapat membantu Anda melakukan langkah pemecahan masalah ini, lihat Perintah berguna untuk pemecahan masalah di bagian Informasi tambahan.
Otentikasi SFTP gagal	1. Verifikasi bahwa pengguna ada di kumpulan pengguna Amazon Cognito. 2. Validasi bahwa pengguna dikonfirmasi dan diaktifkan. 3. Pastikan pengguna memberikan kata sandi yang benar. 4. Verifikasi bahwa alamat IP diizinkan dalam grup keamanan untuk AWS Transfer Family server. 5. Periksa kesalahan dalam log untuk fungsi Lambda otentikasi. 6. Verifikasi bahwa peran IAM untuk AWS Transfer Family memiliki izin yang benar. Untuk daftar AWS CLI perintah yang dapat membantu Anda melakukan langkah

Isu	Solusi
	pemecahan masalah ini, lihat Perintah berguna untuk pemecahan masalah di bagian Informasi tambahan.
Akses unggahan file ditolak	1. Verifikasi bahwa izin AWS KMS kunci dikonfigurasi dengan benar. 2. Periksa kebijakan bucket Amazon S3. 3. Konfirmasikan bahwa izin peran IAM telah dikonfigurasi dengan benar. 4. Verifikasi bahwa titik akhir VPC dikonfigurasi dengan benar 5. Verifikasi bahwa peran IAM untuk AWS Transfer Family memiliki izin yang benar. Untuk daftar AWS CLI perintah yang dapat membantu Anda melakukan langkah pemecahan masalah ini, lihat Perintah berguna untuk pemecahan masalah di bagian Informasi tambahan.

Isu	Solusi
Tidak ada pemindaian malware	1. Aktifkan GuardDuty di akun Anda. 2. Aktifkan Perlindungan Malware untuk S3. 3. Verifikasi bahwa EventBridge aturan Amazon aktif. 4. Periksa kesalahan dalam log untuk fungsi Lambda perutean file. 5. Verifikasi bahwa notifikasi dikonfigurasi dengan benar di Amazon SNS. Untuk daftar AWS CLI perintah yang dapat membantu Anda melakukan langkah pemecahan masalah ini, lihat Perintah berguna untuk pemecahan masalah di bagian Informasi tambahan.
Kesalahan fungsi Lambda	1. Periksa konfigurasi VPC. 2. Verifikasi bahwa titik akhir VPC dikonfigurasi dengan benar. 3. Verifikasi bahwa peran IAM untuk AWS Transfer Family memiliki izin yang benar. 4. Periksa kesalahan di CloudWatch Log. 5. Verifikasi bahwa izin AWS KMS kunci dikonfigurasi dengan benar. Untuk daftar AWS CLI perintah yang dapat membantu Anda melakukan langkah pemecahan masalah ini, lihat Perintah berguna untuk pemecahan masalah di bagian Informasi tambahan.

Sumber daya terkait

- Aplikasi web Transfer Family (Dokumentasi Transfer Family)

Informasi tambahan

Perintah yang berguna untuk pemecahan masalah

Periksa status CloudFormation tumpukan:

```
aws cloudformation describe-stacks \
  --stack-name <STACK_NAME>
```

Daftar semua pengguna di kumpulan pengguna Amazon Cognito:

```
aws cognito-idp list-users \
  --user-pool-id <USER_POOL_ID>
```

Lihat log untuk fungsi Lambda:

```
aws logs describe-log-groups \
  --log-group-name-prefix /aws/lambda/
```

Periksa status GuardDuty:

```
aws guardduty list-detectors
```

Periksa aturan grup keamanan:

```
aws ec2 describe-security-groups \
  --group-ids <SECURITY_GROUP_ID> \
  --output table
```

Periksa status AWS Transfer Family server:

```
aws transfer describe-server \
  --server-id <SERVER_ID>
```

Daftar semua file dalam ember Amazon S3:

```
aws s3 ls s3://<BUCKET_NAME>/ \
  --recursive
```

Periksa status EventBridge aturan:

```
aws events describe-rule \
  --name <RULE_NAME>
```

Amankan data sensitif di CloudWatch Log dengan menggunakan Amazon Macie

Anisha Salunkhe, Omar Franco, dan David Guardiola, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menggunakan Amazon Macie untuk mendeteksi data sensitif secara otomatis dalam grup CloudWatch log Amazon Logs dengan menerapkan alur kerja pemantauan keamanan yang komprehensif. Solusinya menggunakan Amazon Data Firehose untuk mengalirkan entri CloudWatch Log ke Amazon Simple Storage Service (Amazon S3). Macie secara berkala memindai bucket ini untuk informasi identitas pribadi (PII), data keuangan, dan konten sensitif lainnya. Infrastruktur digunakan melalui AWS CloudFormation template yang menyediakan semua yang diperlukan Layanan AWS dan konfigurasi.

CloudWatch Log sering berisi data aplikasi yang secara tidak sengaja dapat menyertakan informasi pengguna yang sensitif. Ini dapat menciptakan risiko kepatuhan dan keamanan. Pendekatan pemantauan log tradisional tidak memiliki kemampuan deteksi data sensitif otomatis. Hal ini dapat membuat sulit untuk mengidentifikasi dan menanggapi potensi eksposur data secara real-time.

Pola ini membantu tim keamanan dan petugas kepatuhan menjaga kerahasiaan data dengan menyediakan deteksi dan peringatan otomatis untuk data sensitif dalam sistem logging. Solusi ini memungkinkan respons insiden proaktif melalui notifikasi Amazon Simple Notification Service (Amazon SNS), dan secara otomatis mengisolasi data sensitif ke bucket Amazon S3 yang aman. Anda dapat menyesuaikan pola deteksi dan mengintegrasikan alur kerja dengan proses operasi keamanan yang ada.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Izin untuk membuat tumpukan CloudFormation
- Grup CloudWatch log Log yang ingin Anda pantau
- Alamat email aktif untuk menerima pemberitahuan dari Amazon SNS
- Akses ke AWS CloudShell
- (Opsional) Akses ke AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi

Batasan

- Macie tunduk pada kuota layanan. Untuk informasi lebih lanjut, lihat Kuota untuk Macie di dokumentasi Macie.

Arsitektur

Arsitektur target

Diagram berikut menunjukkan alur kerja untuk menggunakan Macie untuk memeriksa entri CloudWatch log Log untuk data sensitif.

Alur kerja menunjukkan langkah-langkah berikut:

1. Grup CloudWatch log log menghasilkan log, yang tunduk pada filter langganan.
2. Filter langganan meneruskan log ke Amazon Data Firehose.
3. Log dienkripsi dengan kunci AWS Key Management Service (AWS KMS) saat melewati aliran pengiriman Amazon Data Firehose.
4. Aliran pengiriman mengirimkan log ke bucket log yang diekspor di Amazon S3.
5. Pada pukul 4 pagi setiap hari, Amazon EventBridge memulai AWS Lambda fungsi yang memulai pemindaian Macie untuk data sensitif di bucket log yang diekspor.
6. Jika Macie mengidentifikasi data sensitif di bucket, fungsi Lambda akan menghapus log dari bucket log yang diekspor dan mengenkripsinya dengan file. AWS KMS key
7. Fungsi Lambda mengisolasi log yang berisi data sensitif dalam bucket isolasi data.
8. Identifikasi data sensitif memulai topik Amazon SNS.
9. Amazon SNS mengirimkan pemberitahuan email ke alamat email yang Anda konfigurasikan dengan informasi tentang log yang berisi data sensitif.

Sumber daya yang digunakan

CloudFormation Template menyebarkan sumber daya berikut di target Anda Akun AWS dan Wilayah AWS:

- Dua ember Amazon S3:

  - Bucket log yang diekspor untuk menyimpan data CloudWatch Log
  - Bucket isolasi data untuk menyimpan informasi sensitif
- EventBridge Aturan Amazon yang menanggapi temuan Macie
- AWS Lambda fungsi yang memulai peristiwa dan mengekspor log ke bucket Amazon S3
- Topik dan langganan Amazon SNS
- Aliran Amazon Data Firehose
- Sesi Macie
- Pengidentifikasi data kustom Macie
- Filter langganan CloudWatch Log
- AWS KMS kunci untuk mengenkripsi log yang disimpan dalam ember
- Peran dan kebijakan yang diperlukan AWS Identity and Access Management (IAM) untuk solusi

Alat

Layanan AWS

- AWS CloudFormationmembantu Anda menyiapkan AWS sumber daya, menyediakannya dengan cepat dan konsisten, dan mengelolanya sepanjang siklus hidupnya di seluruh Akun AWS dan. Wilayah AWS
- Amazon CloudWatch Logs membantu Anda memusatkan log dari semua sistem, aplikasi, Layanan AWS sehingga Anda dapat memantau dan mengarsipkannya dengan aman.
- Amazon Data Firehose membantu Anda mengirimkan data streaming real-time ke titik akhir HTTP kustom lainnya Layanan AWS, dan titik akhir HTTP yang dimiliki oleh penyedia layanan pihak ketiga yang didukung.
- Amazon EventBridge adalah layanan bus acara tanpa server yang membantu Anda menghubungkan aplikasi Anda dengan data waktu nyata dari berbagai sumber. Misalnya, sumber seperti AWS Lambda fungsi, titik akhir pemanggilan HTTP menggunakan tujuan API, atau bus acara di tempat lain. Akun AWS
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk membantu melindungi data Anda.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

- Amazon Macie membantu Anda menemukan data sensitif, memberikan visibilitas terhadap risiko keamanan data, dan memungkinkan perlindungan otomatis terhadap risiko tersebut.
- Amazon Simple Notification Service (Amazon SNS) membantu Anda mengoordinasikan dan mengelola pertukaran pesan antara penayang dan klien, termasuk server web dan alamat email.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub sample-macie-for-securing-cloudwatch-logs.

Praktik terbaik

Ikuti praktik CloudFormation terbaik dalam CloudFormation dokumentasi.

Epik

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori kode.	Masukkan perintah berikut untuk mengkloning repositori ke workstation lokal Anda: `git clone https://github.com/aws-samples/sample-macie-for-securing-cloudwatch-logs`	Pengembang aplikasi
(Opsional) Edit CloudFormation template.	1. Buka file main.yaml. 2. Sesuaikan template dengan melakukan salah satu hal berikut: • Anda dapat mengganti nama sumber daya.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Di `parameter` bagian ini, Anda dapat memodifikasi nilai default. • Anda dapat mengubah pola filter langganan . Untuk informasi selengkapnya, lihat Log filter langganan tingkat grup dalam dokumentasi CloudWatch Log. 3. Simpan dan tutup file main.yaml.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 1 — Menyebarkan menggunakan skrip dengan parameter baris perintah.	Masukkan perintah berikut untuk menerapkan solusi dengan menggunakan parameter baris perintah, di mana nilai untuk `enable-macie true` hanya jika Amazon Macie belum diaktifkan:
	`./scripts/test-macie-solution.sh --deploy-stack \`
	`  --stack-name <stack name> \`
	`  --email <email address> \`
	`  --enable-macie <true or false> \`
	`  --region <region> \`
	`  --resource-name <prefix for all resources> \`
	`  --bucket-name <bucket name>`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 2 - Menyebarkan menggunakan skrip dengan variabel lingkungan.	1. Masukkan perintah berikut untuk menentukan variabel lingkungan, di mana nilai untuk `ENABLE_MACIE` `true` hanya jika Amazon Macie belum diaktifkan:
	`export STACK_NAM E=<stack name> export SNS_EMAIL =<email address> export ENABLE_MA CIE=<true or false> export REGION=<r egion> export RESOURCE_ NAME=<prefix for all resources> export BUCKET_NA ME=<bucket name>`
	2. Masukkan perintah berikut untuk memvalida si parameter sebelum penerapan:
	`./scripts/test-mac ie-solution.sh \ --validate-params \ --email <email address> \ --region <region>`
	3. Masukkan perintah berikut untuk menyebarkan solusi:
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`./scripts/test-macie-solution.sh --deploy-stack`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 3 — Menyebarkan menggunakan. AWS CLI	Masukkan perintah berikut untuk menerapkan solusi dengan menggunakan AWS CLI, di mana nilai untuk `true` hanya `EnableMacie` jika Amazon Macie belum diaktifkan: `aws cloudformation create-stack \ --region us-east-1 \ --stack-name macie-for-securing-cloudwatch-logs \ --template-body file://app/main.yml \ --capabilities CAPABILITY_IAM \ --parameters \ ParameterKey=ResourceName,ParameterValue=<prefix for all resources> \ ParameterKey=BucketName,ParameterValue=<bucket name> \ ParameterKey=LogGroupName,ParameterValue=<path for log group> \ ParameterKey=SNSTopicEndpointEmail,ParameterValue=<email address> \ ParameterKey=EnableMacie,ParameterValue=<true or false>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 4 — Menyebarkan melalui. Konsol Manajemen AWS	1. Buka konsol AWS CloudFormation. 2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS untuk membuat tumpukan. 3. Pada halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar) . 4. Pada halaman Buat tumpukan, untuk Prasyarat - Siapkan template, pilih Pilih template yang ada. 5. Di bawah Tentukan templat, pilih Unggah file templat, lalu unggah template main.yaml dari repositori kloning Anda. 6. Pilih Berikutnya. 7. Pada halaman Tentukan detail tumpukan, di kotak Nama tumpukan, masukkan nama tumpukan. 8. Di bagian Parameter , tentukan nilai untuk parameter template berikut. • `ResourceName` : Awalan untuk semua sumber daya	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `BucketName` : Nama unik untuk ember Amazon S3 • `LogGroupName` : Nama grup log untuk CloudWatch Log • `SNSTopicEndpointEmail` : Alamat email untuk notifikasi • `EnableMacie` : Setel ke `true` jika Macie belum diaktifkan • (Opsional)`Region`: Wilayah AWS Tempat Anda ingin menyebarkan tumpukan. • (Opsional)`TemplatePath` : Jalur ke CloudFormation templat 9. Pilih Berikutnya. 10. Untuk Kemampuan, pilih Saya mengakui bahwa template ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam template. 11. Pilih Berikutnya. 12. Pada halaman Tinjau dan buat, tinjau detail tumpukan Anda.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	13. Pilih Kirim untuk meluncurkan tumpukan Anda.	
Pantau status penerapan dan konfirmasikan penerapan.	1. Masukkan perintah berikut untuk memantau status penyebaran: `./scripts/test-macie-solution.sh \ --deployment-status \ --stack-name <stack name>` **Note** Anda juga dapat memantau kemajuan dan status pembuatan tumpukan pada tab Acara untuk tumpukan baru Anda. Untuk informasi selengkapnya, lihat Memantau kemajuan tumpukan. 2. Saat status berubah menjadi CREATE_COMPLETE, tinjau output tumpukan untuk informasi sumber daya.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan langganan Amazon SNS.	Ikuti petunjuk di Konfirmasikan langganan Amazon SNS Anda di dokumentasi Amazon SNS untuk mengonfirmasi langganan Amazon SNS Anda.	Pengembang aplikasi

Pengujian solusi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 1 - Uji dengan pelaporan otomatis.	Jika Anda menggunakan nama stack default, masukkan perintah berikut untuk menguji solusinya:
	`./scripts/test-macie-solution.sh \ --full-test`
	Jika Anda menggunakan nama tumpukan khusus, masukkan perintah berikut untuk menguji solusinya:
	`./scripts/test-macie-solution.sh \ --full-test \ --stack-name <stack name>`
	Jika Anda menggunakan nama tumpukan kustom dan parameter kustom, masukkan
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	perintah berikut untuk menguji solusinya:
	
	`./scripts/test-macie-solution.sh --full-test \`
	`  --stack-name <stack name> \`
	`  --region <region> \`
	`  --log-group <log group path>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 2 - Uji dengan validasi yang ditargetkan.	1. Masukkan perintah berikut untuk menghasilkan data uji dengan informasi sensitif:
	`./scripts/test-mac ie-solution.sh \ --generate-test-da ta \ --stack-name <stack name>`
	Perintah ini melakukan hal berikut:
	• Membuat entri CloudWatch log log yang berisi pola data sensitif realistis, termasuk karyawan IDs, paten IDs, nomor kartu kredit, nomor jaminan sosial, dan alamat email
	• Menghasilkan entri log sensitif dan tidak sensitif untuk pengujian komprehensif
	• Menyediakan pencatata n rinci dari proses pembuatan data pengujian
	2. Masukkan perintah berikut untuk memverifikasi aliran pipa data:
	`./scripts/test-mac ie-solution.sh \`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--verify-pipeline \`
	`--stack-name <stack name>`
	
	Perintah ini melakukan hal berikut:
	• Mengonfirmasi bahwa entri CloudWatch Log dialirkan ke Amazon Data Firehose
	• Memvalidasi bahwa data log dikirim ke bucket Amazon S3 dengan enkripsi yang tepat
	• Memeriksa apakah penyimpanan objek Amazon S3 memiliki struktur awalan yang benar
	• Memverifikasi status enkripsi objek yang disimpan
	• Memantau waktu aliran data dan menyediakan periode tunggu untuk pemrosesan
	3. Masukkan perintah berikut untuk memulai pekerjaan klasifikasi Macie:
	
	`./scripts/test-macie-solution.sh \`
	`--trigger-macie-job \`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>--stack-name
	<stack name></pre>
	Perintah ini melakukan hal berikut:
	• Memicu pekerjaan klasifikasi Macie secara manual melalui fungsi Lambda
	• Memantau status eksekusi pekerjaan dan memberikan umpan balik
	• Memvalidasi ketersediaan layanan Macie sebelum eksekusi
	• Menangani kasus di mana Macie tidak diaktifkan di akun
	• Memberikan hasil eksekusi pekerjaan yang terperinci
	4. Masukkan perintah berikut untuk memvalidasi peringatan dan isolasi data:
	<pre>./scripts/test-macie-solution.sh \
	--verify-alerts \
	--stack-name <stack name></pre>
	Perintah ini melakukan hal berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Mengonfirmasi bahwa EventBridge aturan dikonfigurasi dan aktif dengan benar • Memvalidasi konfigurasi topik Amazon SNS dan status langganan • Memeriksa pengaturan dan izin bucket isolasi data • Memantau pergerakan data sensitif ke bucket isolasi • Memverifikasi fungsiona litas alur kerja peringatan lengkap	

Bersihkan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Opsi 1 - Lakukan pembersihan otomatis.	Jika Anda menggunakan nama stack default, masukkan perintah berikut untuk menghapus tumpukan: `./scripts/cleanup-macie-solution.sh \ --full-cleanup` Jika Anda menggunakan nama tumpukan khusus, masukkan perintah berikut untuk menghapus tumpukan:	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`./scripts/cleanup-macie-solution.sh \ --full-cleanup \ --stack-name <stack name>`
	
	Jika Anda menggunakan nama tumpukan kustom dan parameter kustom, masukkan perintah berikut untuk menghapus tumpukan:
	
	`./scripts/cleanup-macie-solution.sh \ --full-cleanup \ --stack-name <stack name> \ --region <region> \ --disable-macie <true or false>`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Opsi 2 — Lakukan step-by-step pembersihan.	1. Masukkan perintah berikut untuk menghentikan proses aktif:
	`./scripts/cleanup-macie-solution.sh \ --stop-processes \ --stack-name <stack name>`
	Perintah ini melakukan hal berikut:
	• Mematikan EventBridge aturan untuk mencegah eksekusi pekerjaan baru
	• Menghentikan pekerjaan klasifikasi Macie yang sedang berjalan
	• Membatalkan eksekusi pekerjaan Macie yang tertunda
	• Menghapus semua pesan Amazon SNS yang tertunda dalam antrian
	• Menyediakan pembaruan status untuk setiap proses yang dihentikan
	2. Masukkan perintah berikut untuk mengosongkan bucket Amazon S3:
	`./scripts/cleanup-macie-solution.sh \ --empty-buckets \`
		AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`--stack-name <stack name>`
	
	Perintah ini melakukan hal berikut:
	• Menghapus semua objek dari ember untuk CloudWatch Log
	• Menghapus semua objek dari bucket Amazon S3 isolasi data
	• Menghapus semua unggahan multipart yang tidak lengkap
	• Menangani objek berversi jika versi Amazon S3 diaktifkan
	• Menyediakan jumlah objek dan pembaruan kemajuan penghapusan
	3. Masukkan perintah berikut untuk menghapus CloudFormation tumpukan:
	
	`./scripts/cleanup-macie-solution.sh \ --delete-stack \ --stack-name <stack name>`
	
	Perintah ini melakukan hal berikut:	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Memulai proses penghapusan CloudFormation tumpukan • Memantau kemajuan penghapusan dengan pembaruan status waktu nyata • Menangani kegagalan penghapusan dengan pelaporan kesalahan terperinci • Menunggu penghapusan tumpukan lengkap sebelum melanjutkan • Menyediakan riwayat acara tumpukan untuk pemecahan masalah 4. Masukkan perintah berikut untuk membersihkan sumber daya Macie: `./scripts/cleanup-macie-solution.sh \ --cleanup-macie \ --stack-name <stack name>` Tumpukan ini melakukan hal berikut: • Menghapus pengidentifikasi data kustom yang dibuat oleh solusi	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• Membersihkan artefak dan temuan pekerjaan Macie yang tersisa • Menonaktifkan sesi Macie jika diaktifkan oleh tumpukan • Menangani kasus di mana sumber daya Macie dibagikan dengan aplikasi lain • Memberikan status pembersihan terperinci untuk setiap komponen Macie	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Verifikasi membersihkan.	1. Masukkan perintah berikut untuk memverifikasi bahwa tumpukan telah dihapus:
	`aws cloudformation describe-stacks \ --stack-name <stack name> \ --region <region>`
	2. Masukkan perintah berikut untuk memverifikasi bahwa bucket Amazon S3 telah dihapus:
	`aws s3 ls | grep macie`
	3. Masukkan perintah berikut untuk memverifikasi bahwa pengidentifikasi data kustom Macie telah dihapus:
	`aws macie2 list-custom-data-identifiers \ --region <region>`
	4. Masukkan perintah berikut untuk memeriksa sumber daya yang tersisa:
	`./scripts/cleanup-macie-solution.sh \ --verify-cleanup \ --stack-name <stack name>`
		AWS Umum

Pemecahan masalah

Isu	Solusi
CloudFormation status tumpukan menunjukkan CREATE_FAILED.	CloudFormation Template dikonfigurasi untuk mempublikasikan log ke CloudWatch Log. Anda dapat melihat log di Konsol Manajemen AWS sehingga Anda tidak perlu terhubung ke EC2 instans Amazon Anda. Untuk informasi selengkapnya, lihat Melihat CloudFormation log di konsol (posting AWS blog).
CloudFormation `delete-stack` perintah gagal.	Beberapa sumber daya harus kosong sebelum dapat dihapus. Misalnya, Anda harus menghapus semua objek di bucket Amazon S3 atau menghapus semua instance di grup EC2 keamanan Amazon sebelum dapat menghapus bucket atau grup keamanan. Untuk informasi selengkapnya, lihat Hapus tumpukan gagal dalam dokumentasi Amazon S3.
Kesalahan saat mengurai parameter.	Saat Anda menggunakan AWS CLI atau CloudFormation konsol untuk memberikan nilai, tambahkan tanda kutip.

Sumber daya terkait

- Praktik terbaik arsitektur untuk penyimpanan (AWS situs web)
- Filter sintaks pola untuk filter metrik, filter langganan, peristiwa log filter, dan Live Tail (dokumentasi CloudWatch Log)
- Merancang dan menerapkan pencatatan dan pemantauan dengan Amazon CloudWatch (PanduanAWS Preskriptif)
- Pemecahan masalah CloudFormation(dokumentasi)CloudFormation

Kirim peringatan dari AWS Network Firewall ke saluran Slack

Venki Srivatsav dan Aromal Raj Jayarajan, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara menerapkan firewall dengan menggunakan Amazon Web Services (AWS) Network Firewall dengan model penyebaran terdistribusi dan cara menyebarkan peringatan yang dihasilkan oleh AWS Network Firewall ke saluran Slack yang dapat dikonfigurasi.

Standar kepatuhan seperti Payment Card Industry Data Security Standard (PCI DSS) mengharuskan Anda menginstal dan memelihara firewall untuk melindungi data pelanggan. Di AWS Cloud, virtual private cloud (VPC) dianggap sama dengan jaringan fisik dalam konteks persyaratan kepatuhan ini. Anda dapat menggunakan Network Firewall untuk memantau lalu lintas jaringan antara VPCs dan untuk melindungi beban kerja Anda yang berjalan di VPCs diatur oleh standar kepatuhan. Network Firewall memblokir akses atau menghasilkan peringatan ketika mendeteksi akses tidak sah dari orang lain VPCs di akun yang sama. Namun, Network Firewall mendukung sejumlah tujuan untuk mengirimkan peringatan. Tujuan ini mencakup bucket Amazon Simple Storage Service (Amazon S3), grup log Amazon, dan CloudWatch aliran pengiriman Amazon Data Firehose. Setiap tindakan lebih lanjut pada notifikasi ini memerlukan analisis offline dengan menggunakan Amazon Athena atau Amazon Kinesis.

Pola ini menyediakan metode untuk menyebarkan peringatan yang dihasilkan oleh Network Firewall ke saluran Slack yang dapat dikonfigurasi untuk tindakan lebih lanjut dalam waktu dekat. Anda juga dapat memperluas fungsionalitas ke mekanisme peringatan lainnya seperti PagerDuty, Jira, dan email. (Kustomisasi tersebut berada di luar cakupan pola ini.)

Prasyarat dan batasan

Prasyarat

- Saluran Slack (lihat Memulai di pusat bantuan Slack)
- Hak istimewa yang diperlukan untuk mengirim pesan ke saluran
- URL titik akhir Slack dengan token API (pilih aplikasi Anda dan pilih webhook masuk untuk melihat URL-nya; untuk informasi selengkapnya, lihat Membuat Webhook Masuk dalam dokumentasi Slack API)
- Instans pengujian Amazon Elastic Compute Cloud (Amazon EC2) di subnet beban kerja
- Aturan uji di Network Firewall

- Lalu lintas aktual atau simulasi untuk memicu aturan pengujian
- Bucket S3 untuk menyimpan file sumber yang akan digunakan

Batasan

- Saat ini solusi ini hanya mendukung rentang Classless Inter-Domain Routing (CIDR) tunggal sebagai filter untuk sumber dan tujuan. IPs

Arsitektur

Tumpukan teknologi target

- Satu VPC
- Empat subnet (dua untuk firewall dan dua untuk beban kerja)
- gateway internet
- Empat tabel rute dengan aturan
- Bucket S3 digunakan sebagai tujuan peringatan, dikonfigurasi dengan kebijakan bucket dan pengaturan acara untuk menjalankan fungsi Lambda
- Fungsi Lambda dengan peran eksekusi, untuk mengirim pemberitahuan Slack
- Rahasia AWS Secrets Manager untuk menyimpan URL Slack
- Firewall jaringan dengan konfigurasi peringatan
- Saluran kendur

Semua komponen kecuali untuk saluran Slack disediakan oleh CloudFormation template dan fungsi Lambda yang disediakan dengan pola ini (lihat bagian Kode).

Arsitektur target

Pola ini mengatur firewall jaringan terdesentralisasi dengan integrasi Slack. Arsitektur ini terdiri dari VPC dengan dua Availability Zones. VPC mencakup dua subnet yang dilindungi dan dua subnet firewall dengan titik akhir firewall jaringan. Semua lalu lintas yang masuk dan keluar dari subnet yang dilindungi dapat dipantau dengan membuat kebijakan dan aturan firewall. Firewall jaringan dikonfigurasi untuk menempatkan semua peringatan dalam ember S3. Bucket S3 ini dikonfigurasi untuk memanggil fungsi Lambda saat menerima `put` acara. Fungsi Lambda mengambil URL Slack yang dikonfigurasi dari Secrets Manager dan mengirimkan pesan notifikasi ke ruang kerja Slack.

Untuk informasi selengkapnya tentang arsitektur ini, lihat postingan blog AWS Model penerapan untuk AWS Network Firewall.

Alat

Layanan AWS

- AWS Network Firewall adalah firewall jaringan yang stateful, dikelola, dan layanan deteksi dan pencegahan intrusi untuk AWS Cloud VPCs . Anda dapat menggunakan Network Firewall untuk memfilter lalu lintas di perimeter VPC Anda dan melindungi beban kerja Anda di AWS.
- AWS Secrets Manager adalah layanan untuk penyimpanan dan pengambilan kredensi. Dengan menggunakan Secrets Manager, Anda dapat mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Pola ini menggunakan Secrets Manager untuk menyimpan URL Slack.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web. Pola ini menggunakan Amazon S3 untuk menyimpan CloudFormation template dan skrip Python untuk fungsi Lambda. Ini juga menggunakan bucket S3 sebagai tujuan peringatan firewall jaringan.
- AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Pola ini menggunakan AWS CloudFormation untuk secara otomatis menerapkan arsitektur terdistribusi untuk Firewall Manager.

Kode

Kode untuk pola ini tersedia di GitHub, di repositori Network Firewall Slack Integration. Di `src` folder repositori, Anda akan menemukan:

- Satu set CloudFormation file dalam format YAMAL. Anda menggunakan template ini untuk menyediakan komponen untuk pola ini.
- File sumber Python (`slack-lambda.py`) untuk membuat fungsi Lambda.

- Paket penyebaran arsip.zip (`slack-lambda.py.zip`) untuk mengunggah kode fungsi Lambda Anda.

Untuk menggunakan file-file ini, ikuti instruksi di bagian selanjutnya.

Epik

Siapkan ember S3

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat ember S3.	1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/ 2. Pilih atau buat bucket S3 untuk meng-host kode. Nama bucket S3 unik secara global, dan namespace dibagikan oleh semua akun AWS. Nama bucket S3 tidak dapat menyertakan garis miring terdepan. Kami menyarankan Anda menggunakan awalan untuk mengatur kode untuk pola ini. Untuk informasi selengkapnya, lihat Membuat bucket di dokumentasi Amazon S3.	Pengembang aplikasi, Pemilik aplikasi, administrator Cloud
Unggah CloudFormation template dan kode Lambda.	1. Unduh file-file berikut dari GitHub repositori untuk pola ini:	Pengembang aplikasi, Pemilik aplikasi, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `base.yml` • `igw-ingress-route.yml` • `slack-lambda.py` • `slackLambda.yml` • `decentralized-deployment.yml` • `protected-subnet-route.yml` • `slack-lambda.py.zip` 2. Unggah file ke bucket S3 yang Anda buat.	

Menyebarkan template CloudFormation

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Luncurkan CloudFormation template.	Buka CloudFormation konsol AWS di Wilayah AWS yang sama dengan bucket S3 Anda dan terapkan template. `base.yml` Template ini membuat sumber daya AWS yang diperlukan dan fungsi Lambda untuk peringatan yang akan dikirimkan ke saluran Slack. Untuk informasi selengkapnya tentang penerapan CloudFormation templat, lihat Membuat tumpukan di CloudFormation	Pengembang aplikasi, Pemilik aplikasi, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	konsol AWS dalam CloudFormation dokumentasi.	
Lengkapi parameter dalam template.	Tentukan nama tumpukan dan konfigurasikan nilai parameter. Untuk daftar parameter, deskripsinya, dan nilai default, lihat CloudFormation parameter di bagian Informasi tambahan.	Pengembang aplikasi, Pemilik aplikasi, administrator Cloud
Buat tumpukan.	1. Tinjau detail tumpukan dan perbarui nilai berdasarkan persyaratan lingkungan Anda. 2. Pilih Buat tumpukan untuk menyebarkan template.	Pengembang aplikasi, Pemilik aplikasi, administrator Cloud

Verifikasi solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Uji penyebaran.	Gunakan CloudFormation konsol AWS atau AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) untuk memverifikasi bahwa sumber daya yang tercantum di bagian tumpukan teknologi Target telah dibuat. Jika CloudFormation template gagal diterapkan dengan sukses, periksa nilai	Pengembang aplikasi, Pemilik aplikasi, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	yang Anda berikan untuk `pAvailabilityZone2` parameter `pAvailabi lityZone1` dan. Ini harus sesuai untuk Wilayah AWS tempat Anda menerapka n solusinya. Untuk daftar Availability Zone untuk setiap Wilayah, lihat Wilayah dan Zona di EC2 dokumentasi Amazon.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Fungsionalitas uji.	1. Buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/.
	
	2. Buat EC2 instance di salah satu subnet yang dilindungi. Pilih Amazon Linux 2 AMI (HVM) untuk digunakan sebagai server HTTPS. Untuk petunjuknya, lihat Meluncurkan instance di EC2 dokumentasi Amazon.
	
	Note
	Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs.
	
	3. Gunakan data pengguna berikut untuk menginstal server web pada EC2 instance:
	
	`#!/bin/bash`
	`yum install httpd -y`
	`systemctl start httpd`
	`systemctl stop firewalld`
	`cd /var/www/html`
		Pengembang aplikasi, Pemilik aplikasi, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	`echo "Hello!! this is a NFW alert test page, 200 OK" > index.html`
	
	4. Buat aturan firewall jaringan berikut:
	
	Aturan tanpa kewargane garaan:
	
	`Source: 0.0.0.0/0`
	`Destination 10.0.3.65 /32 (private IP of the EC2 instance)`
	`Action: Forward`
	
	Aturan stateful:
	
	`Protocol: HTTP`
	`Source ip/port: Any / Any`
	`Destination ip/port: Any /Any`
	
	5. Dapatkan IP publik dari server web yang Anda buat di langkah 3.
	
	6. Akses IP publik di browser. Anda akan melihat pesan berikut di browser:
	
	`Hello!! this is a NFW alert test page, 200 OK`
	
	Anda juga akan mendapatk an notifikasi di saluran Slack.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Pemberitahuan mungkin tertunda, tergantung pada ukuran pesan. Untuk tujuan pengujian, pertimbangkan untuk menyediakan filter CIDR yang tidak terlalu sempit (misalnya, nilai CIDR dengan/32 akan dianggap terlalu sempit, dan /8 akan terlalu luas). Untuk informasi selengkapnya, lihat bagian Memfilter perilaku di Informasi tambahan.	

Sumber daya terkait

- Model penerapan untuk AWS Network Firewall (postingan blog AWS)
- Kebijakan AWS Network Firewall (dokumentasi AWS)
- Integrasi Slack Firewall Jaringan (GitHub repositori)
- Buat ruang kerja Slack (pusat bantuan Slack)

Informasi tambahan

CloudFormation parameter

Parameter	Deskripsi	Nilai default atau sampel
`pVpcName`	Nama VPC yang akan dibuat.	Inspeksi
`pVpcCidr`	Rentang CIDR untuk VPC untuk dibuat.	10.0.0.0/16
`pVpcInstanceTenancy`	Bagaimana EC2 instance didistribusikan di seluruh	default

	perangkat keras fisik. Pilihannya adalah `default` (sewa bersama) atau `dedicated` (sewa tunggal).	
`pAvailabilityZone1`	Availability Zone pertama untuk infrastruktur.	us-timur-2a
`pAvailabilityZone2`	Availability Zone kedua untuk infrastruktur.	us-timur-2b
`pNetworkFirewallSubnet1Cidr`	Rentang CIDR untuk subnet firewall pertama (minimum /28).	10.0.1.0/24
`pNetworkFirewallSubnet2Cidr`	Rentang CIDR untuk subnet firewall kedua (minimum /28).	10.0.2.0/24
`pProtectedSubnet1Cidr`	Rentang CIDR untuk subnet pertama yang dilindungi (beban kerja).	10.0.3.0/24
`pProtectedSubnet2Cidr`	Rentang CIDR untuk subnet kedua yang dilindungi (beban kerja).	10.0.4.0/24
`pS3BucketName`	Nama bucket S3 yang ada tempat Anda mengunggah kode sumber Lambda.	kita-w2- yourname-lambda-functions
`pS3KeyPrefix`	Awalan bucket S3 tempat Anda mengunggah kode sumber Lambda.	uji aod
`pAWSSecretName4Slack`	Nama rahasia yang menyimpan URL Slack.	SlackEnpoint-Cfn
`pSlackChannelName`	Nama saluran Slack yang Anda buat.	pemberitahuan somename

`pSlackUserName`	Nama pengguna Slack.	Pengguna Slack
`pSecretKey`	Ini bisa menjadi kunci apa saja. Kami menyarankan Anda menggunakan default.	WebHooURL
`pWebHookUrl`	Nilai URL Slack.	https://hooks.slack.com/services/T??? 9T?? /A031885 JRM7 /9D4Y
`pAlertS3Bucket`	Nama bucket S3 yang akan digunakan sebagai tujuan peringatan firewall jaringan. Ember ini akan dibuat untuk Anda.	kita-w2- yourname-security-aod-alerts
`pSecretTagName`	Nama tag untuk rahasianya.	AppName
`pSecretTagValue`	Nilai tag untuk nama tag yang ditentukan.	LambdaSlackIntegration
`pdestCidr`	Filter untuk rentang CIDR tujuan. Untuk informasi selengkapnya, lihat bagian selanjutnya, Filter perilaku.	10.0.0.0/16
`pdestCondition`	Bendera untuk menunjukkan apakah akan mengecualikan atau menyertakan kecocokan tujuan. Untuk informasi lebih lanjut, lihat bagian selanjutn ya. Nilai yang valid adalah `include` dan `exclude`.	memasukkan
`psrcCidr`	Filter untuk rentang CIDR sumber untuk diperingatkan. Untuk informasi lebih lanjut, lihat bagian selanjutnya.	118.2.0.0/16

`psrcCondition`	Bendera untuk mengecualikan atau menyertakan kecocokan sumber. Untuk informasi lebih lanjut, lihat bagian selanjutnya.	memasukkan

Perilaku filter

Jika Anda belum mengonfigurasi filter apa pun di AWS Lambda, semua peringatan yang dihasilkan akan dikirim ke saluran Slack Anda. Sumber dan tujuan peringatan IPs yang dihasilkan dicocokkan dengan rentang CIDR yang Anda konfigurasikan saat Anda menerapkan templat. CloudFormation Jika kecocokan ditemukan, kondisi diterapkan. Jika sumber atau tujuan berada dalam rentang CIDR yang dikonfigurasi dan setidaknya salah satunya dikonfigurasi dengan kondisi`include`, peringatan akan dihasilkan. Tabel berikut memberikan contoh nilai CIDR, kondisi, dan hasil.

	CIDR yang dikonfigurasi	Peringatan IP	Dikonfigurasi	Pemberitahuan
Sumber	10.0.0.0/16	10.0.0.25	memasukkan	Ya
Destinasi	100.0.0.0/16	202.0.0.13	memasukkan	

	CIDR yang dikonfigurasi	Peringatan IP	Dikonfigurasi	Pemberitahuan
Sumber	10.0.0.0/16	10.0.0.25	mengecualikan	Tidak
Destinasi	100.0.0.0/16	202.0.0.13	memasukkan	

	CIDR yang dikonfigurasi	Peringatan IP	Dikonfigurasi	Pemberitahuan
Sumber	10.0.0.0/16	10.0.0.25	memasukkan	Ya
Destinasi	100.0.0.0/16	100.0.0.13	memasukkan	

	CIDR yang dikonfigurasi	Peringatan IP	Dikonfigurasi	Pemberitahuan
Sumber	10.0.0.0/16	90.0.0.25	memasukkan	Ya
Destinasi	Null	202.0.0.13	memasukkan	

	CIDR yang dikonfigurasi	Peringatan IP	Dikonfigurasi	Pemberitahuan
Sumber	10.0.0.0/16	90.0.0.25	memasukkan	Tidak
Destinasi	100.0.0.0/16	202.0.0.13	memasukkan	

Kirim atribut khusus ke Amazon Cognito dan masukkan ke dalam token

Carlos Alessandro Ribeiro dan Mauricio Mendoza, Amazon Web Services

Ringkasan

Mengirim atribut khusus ke proses autentikasi Amazon Cognito dapat memberikan konteks tambahan ke aplikasi, mengaktifkan kontrol akses yang lebih terperinci, dan mempermudah pengelolaan profil pengguna dan persyaratan otentikasi. Fitur-fitur ini berguna dalam berbagai aplikasi dan skenario, dan mereka dapat membantu Anda meningkatkan keamanan dan fungsionalitas aplikasi secara keseluruhan.

Pola ini menunjukkan cara mengirim atribut khusus ke proses otentikasi Amazon Cognito saat aplikasi perlu memberikan konteks tambahan ke token akses atau token identitas (ID). Anda menggunakan Node.js sebagai aplikasi backend. Aplikasi ini mengautentikasi pengguna dari kumpulan pengguna Amazon Cognito dan meneruskan atribut khusus yang diperlukan untuk pembuatan token. Anda dapat menggunakan AWS Lambda pemicu untuk Amazon Cognito untuk menyesuaikan proses otentikasi Anda tanpa penyesuaian kode utama atau upaya yang signifikan.

> **Important**
>
> Kode dan sampel dalam pola ini tidak direkomendasikan untuk beban kerja produksi karena ditujukan untuk tujuan demonstrasi saja. Untuk beban kerja produksi, konfigurasi tambahan diperlukan di sisi klien. Gunakan pola ini sebagai referensi untuk pilot atau proof-of-concept tujuan saja.

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS
- Izin untuk membuat dan mengelola kumpulan dan fungsi pengguna Amazon Cognito AWS Lambda
- AWS Command Line Interface (AWS CLI), diinstal dan dikonfigurasi
- Lingkungan pengembangan terintegrasi (IDE) yang mendukung Node.js
- Node.js versi 18 atau yang lebih baru, diinstal

- npm versi 8 atau yang lebih baru, diinstal
- TypeScript, dipasang

Batasan

- Pola ini tidak berlaku untuk integrasi aplikasi melalui alur otentikasi Client Credentials.
- Pemicu pembuatan pra-token hanya dapat menambah atau mengubah beberapa atribut token akses dan token identitas. Untuk informasi selengkapnya, lihat Pemicu Lambda pra pembuatan token di dokumentasi Amazon Cognito.

Arsitektur

Arsitektur target

Diagram berikut menunjukkan arsitektur target untuk pola ini. Ini juga menunjukkan bagaimana aplikasi Node.js dapat bekerja dengan backend untuk memperbarui database. Namun, pembaruan basis data backend berada di luar cakupan pola ini.

Diagram menunjukkan alur kerja berikut:

1. Aplikasi Node.js mengeluarkan token akses dengan atribut khusus ke kumpulan pengguna Amazon Cognito.
2. Kumpulan pengguna Amazon Cognito memulai fungsi Lambda generasi pra-token, yang menyesuaikan akses dan token ID.
3. Aplikasi Node.js melakukan panggilan API melalui Amazon API Gateway.

> **Note**
>
> Komponen arsitektur lain yang ditunjukkan dalam arsitektur ini misalnya saja dan berada di luar cakupan pola ini.

Otomatisasi dan skala

Anda dapat mengotomatiskan penyediaan kumpulan pengguna Amazon Cognito, AWS Lambda fungsi, instance database, dan sumber daya lainnya dengan menggunakan, HashiCorp Terraform

AWS CloudFormation, atau alat infrastruktur AWS Cloud Development Kit (AWS CDK)sebagai kode (IAc) yang didukung. Jika Anda ingin menskalakan penerapan Anda, gunakan pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD), yang membantu mencegah kesalahan yang terkait dengan penerapan manual.

Alat

Layanan AWS

- Amazon API Gateway membantu Anda membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs dalam skala apa pun.
- Amazon Cognito menyediakan otentikasi, otorisasi, dan manajemen pengguna untuk aplikasi web dan seluler.
- Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang cepat dan dapat diskalakan yang membantu Anda menjalankan, menghentikan, dan mengelola kontainer di klaster.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.
- AWS SDK untuk JavaScriptmenyediakan JavaScript API untuk Layanan AWS. Anda dapat menggunakannya untuk membangun pustaka atau aplikasi untuk Node.js atau browser.

Alat-alat lainnya

- Node.js adalah lingkungan JavaScript runtime berbasis peristiwa yang dirancang untuk membangun aplikasi jaringan yang dapat diskalakan.
- npm adalah registri perangkat lunak yang berjalan di lingkungan Node.js dan digunakan untuk berbagi atau meminjam paket dan mengelola penyebaran paket pribadi.

Praktik terbaik

Kami menyarankan Anda menerapkan praktik terbaik berikut:

- Rahasia dan data sensitif — Jangan menyimpan rahasia atau data sensitif dalam aplikasi. Gunakan sistem eksternal tempat aplikasi dapat menarik data dari, seperti AWS AppConfigAWS Secrets Manager, atau AWS Systems Manager Parameter Store.

- Penerapan standar — Gunakan CI/CD pipeline untuk menyebarkan aplikasi Anda. Anda dapat menggunakan layanan seperti AWS CodeBilddan AWS CodePipeline.
- Token kedaluwarsa — Tetapkan tanggal kedaluwarsa singkat untuk token akses.
- Gunakan koneksi aman — Semua komunikasi antara aplikasi klien dan backend harus dienkripsi dengan menggunakan SSL/TLS. Gunakan AWS Certificate Manager (ACM) untuk membuat dan mengelola SSL/TLS sertifikat, dan gunakan Amazon CloudFront atau Elastic Load Balancing untuk SSL/TLS menangani penghentian.
- Validasi input pengguna — Pastikan semua input pengguna divalidasi untuk mencegah serangan injeksi dan kerentanan keamanan lainnya. Gunakan pustaka validasi input dan layanan seperti Amazon API Gateway dan AWS WAFuntuk mencegah vektor serangan umum.
- Gunakan peran IAM — Gunakan peran AWS Identity and Access Management (IAM) untuk mengontrol akses ke AWS sumber daya dan memastikan bahwa hanya pengguna yang berwenang yang memiliki akses. Ikuti prinsip hak istimewa paling sedikit dan pastikan bahwa setiap pengguna hanya memiliki izin yang diperlukan untuk melakukan peran mereka.
- Gunakan kebijakan kata sandi — Konfigurasikan kebijakan kata sandi yang memenuhi persyaratan keamanan Anda, seperti panjang minimum, kompleksitas, dan kedaluwarsa. Gunakan Secrets Manager atau AWS Systems Manager Parameter Store untuk menyimpan dan mengelola kata sandi dengan aman.
- Aktifkan otentikasi multi-faktor (MFA) — Aktifkan MFA bagi semua pengguna untuk menyediakan lapisan keamanan tambahan dan mengurangi risiko akses yang tidak sah. Gunakan AWS IAM Identity Centeratau Amazon Cognito untuk mengaktifkan MFA dan metode otentikasi lainnya.
- Simpan informasi sensitif dengan aman — Simpan informasi sensitif, seperti kata sandi dan token akses, dengan aman menggunakan AWS Key Management Service (AWS KMS) atau layanan enkripsi lainnya.
- Gunakan metode otentikasi yang kuat — Untuk meningkatkan keamanan proses otentikasi, gunakan metode otentikasi yang kuat, seperti otentikasi biometrik atau otentikasi multi-faktor.
- Memantau aktivitas yang mencurigakan — Gunakan AWS CloudTraildan alat pemantauan lainnya untuk memantau aktivitas mencurigakan dan potensi ancaman keamanan. Siapkan peringatan otomatis untuk aktivitas yang tidak biasa, dan gunakan Amazon GuardDuty atau AWS Security Hub CSPMuntuk mendeteksi potensi ancaman.
- Tinjau dan perbarui kebijakan keamanan secara berkala — Tinjau dan perbarui kebijakan dan prosedur keamanan Anda secara teratur untuk memastikan bahwa kebijakan tersebut memenuhi persyaratan keamanan dan praktik terbaik Anda yang berubah. Gunakan AWS Config untuk melacak dan mengaudit perubahan pada kebijakan dan prosedur keamanan Anda.

- Pendaftaran otomatis — Jangan aktifkan pendaftaran otomatis ke kumpulan pengguna Amazon Cognito. Untuk informasi selengkapnya, lihat Mengurangi risiko penipuan pendaftaran pengguna dan pemompaan SMS dengan kumpulan pengguna Amazon CognitoAWS (posting blog).

Untuk praktik terbaik tambahan, lihat Praktik terbaik keamanan untuk kumpulan pengguna Amazon Cognito di dokumentasi Amazon Cognito.

Epik

Siapkan sumber AWS daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Membuat pengguna.	1. Di terminal CLI, masukkan perintah berikut untuk membuat kumpulan pengguna Amazon Cognito:
	`aws cognito-idp create-user-pool \ --pool-name <MyUserPool>`
	2. Masukkan perintah berikut untuk membuat ID klien dan rahasia yang akan digunakan dengan SDK untuk JavaScript:
	`aws cognito-idp create-user-pool-client \ --user-pool-id <UserPoolID> \ --client-name <MyNewClient> \ --no-generate-secret \ --explicit-auth-flows "USER_PAS`
		Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre>SWORD_AUTH"
	"ADMIN_NO_SRP_AUTH"</pre>
	Untuk informasi selengkap nya dan petunjuk tentang cara menyiapkan kumpulan pengguna di Konsol Manajemen AWS, lihat Memulai kumpulan pengguna dan Menambahkan lebih banyak fitur dan opsi keamanan ke kumpulan pengguna Anda.
	Tip
	Untuk mengurang i biaya, gunakan paket Essentials atau paket Lite untuk menguji pola ini. Untuk informasi selengkap nya, lihat harga Amazon Cognito.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan pengguna ke kumpulan pengguna.	Masukkan perintah berikut untuk membuat satu pengguna di kumpulan pengguna Amazon Cognito: `aws cognito-idp sign-up \ --client-id <ClientID> \ --username <jane@example.com> \ --password <PASSWORD> \ --user-attributes Name="email",Value="<jane@example.com>" Name="name",Value="<Jane>"`	Pengembang aplikasi, AWS DevOps

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan klien aplikasi ke kumpulan pengguna.	1. Arahkan ke konsol Amazon Cognito. 2. Pilih kumpulan Pengguna. 3. Pilih kumpulan pengguna yang Anda buat sebelumnya. 4. Di bawah Tentukan aplikasi Anda, pilih jenis Aplikasi yang paling sesuai dengan skenario aplikasi yang ingin Anda buat layanan otentikasi dan otorisasi. 5. Dalam Nama aplikasi Anda, masukkan nama deskriptif atau lanjutkan dengan nama default. 6. Di bawah Tambahkan URL pengembalian, masukkan jalur pengalihan ke aplikasi Anda setelah pengguna menyelesaikan otentikasi. 7. Pilih Buat aplikasi Anda. Anda dapat mengatur opsi lanjutan setelah membuat klien aplikasi.	Administrator sistem AWS, administrator AWS, AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat pemicu Lambda untuk pembuatan pra-token.	1. Buka halaman Fungsi di konsol Lambda. 2. Pilih Buat fungsi. 3. Pilih Penulis dari awal. 4. Di panel Informasi dasar, untuk nama Fungsi, masukkan `myPreTokenGenerationLambdaFunction` . 5. Untuk Runtime, pilih Node.js 22. 6. Biarkan arsitektur disetel ke x86_64. 7. Pilih Buat fungsi. 8. Pada tab Kode, di bawah Sumber Kode, masukkan contoh fungsi Lambda dari bagian Sumber daya tambahan dari pola ini. Fungsi Lambda ini menyuntikkan atribut kustom ke token ID dan token akses.	AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan alur kerja kumpulan pengguna.	1. Arahkan ke konsol Amazon Cognito. 2. Pilih kumpulan Pengguna. 3. Pilih kumpulan pengguna yang Anda buat sebelumnya. 4. Di panel navigasi kiri, di bawah Otentikasi, pilih Ekstensi. 5. Pilih Tambahkan pemicu Lambda. 6. Tambahkan atau edit pemicu pembuatan Pra token. 7. Untuk tipe Trigger, pilih Authentication. 8. Di bawah Otentikasi, pilih Pre token generation trigger. 9. Untuk fungsi Tetapkan Lambda, pilih. `myPreTokenGenerationLambdaFunction` 10. Pilih Tambahkan pemicu Lambda. 11. Pilih versi acara Trigger dari fitur Dasar+kustomisasi token akses untuk identitas pengguna atau Fitur dasar +kustomisasi token akses untuk identitas pengguna dan mesin. Pengaturan ini	AWS DevOps, Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	memperbarui parameter permintaan yang dikirimkan Amazon Cognito ke fungsi Anda untuk menyertakan bidang untuk kustomisasi token akses. Untuk informasi selengkapnya, lihat Menyesuaikan alur kerja kumpulan pengguna dengan pemicu Lambda di dokumentasi Amazon Cognito.	

Buat aplikasi Node.js

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Buat aplikasi Node.js.	1. Di terminal, masukkan perintah berikut:
	
	`mkdir my-app`
	`cd my-app`
	`npm init -y`
	`npm install --save-dev typescript`
	`npm i --save-dev @types/node`
	`npm i amazon-cognito-identity-js`
	`npx tsc --init`
	`mkdir src`
	`touch src/index.ts`
	
	2. Ubah `package.json` file untuk menyertakan skrip TypeScript kompilasi:
		Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`"scripts": { "start": "node src/index.js", "build": "tsc" },`	
Menerapkan logika otentikasi.	1. Buka file `index.ts`. 2. Tempel sampel TypeScript yang disediakan di bagian Sumber daya tambahan dari pola ini. 3. Simpan dan tutup file `index.ts`. **Note** Anda dapat membuat TypeScript file Anda sendiri atau memodifikasi sampel yang disediakan sesuai kebutuhan untuk kasus penggunaan Anda.	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfigurasikan variabel lingkungan dan file konfigurasi.	Di terminal, masukkan perintah berikut untuk membuat variabel lingkungan: `export USERNAME= "<COGNITO_USER_NAME>" export PASSWORD= "<COGNITO_USER_PASSWORD>" export USER_POOL_ID="<COGNITO_USER_ID>" export CLIENT_ID ="<COGNITO_CLIENT_ID>"` **Important** Jangan membuat hardcode rahasia atau mengekspos kredensialmu.	Pengembang aplikasi
Jalankan aplikasi.	Masukkan perintah berikut untuk menjalankan aplikasi dan konfirmasikan bahwa itu berfungsi: `npm run build npm start`	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Konfirmasikan bahwa atribut khusus disuntikkan ke token.	Gunakan fitur debugging untuk IDE Anda untuk melihat akses dan token ID. Konfirmasikan bahwa atribut kustom telah ditambahkan. Untuk contoh token, lihat bagian Informasi tambahan dari pola ini.	Pengembang aplikasi

Pemecahan masalah

Isu	Solusi
ID klien tidak valid saat mencoba mengauten tikasi pengguna	Kesalahan ini biasanya terjadi ketika Anda menggunakan ID klien dengan rahasia klien yang dihasilkan. Anda harus membuat ID klien tanpa rahasia yang melekat padanya. Untuk informasi selengkapnya, lihat Pengaturan khusus aplikasi dengan klien aplikasi.

Sumber daya terkait

- Menyesuaikan alur kerja kumpulan pengguna dengan pemicu Lambda (dokumentasi Amazon Cognito)
- Pemicu Lambda generasi pra token (dokumentasi Amazon Cognito)
- CognitoIdentityProviderClient(AWS SDK untuk JavaScript dokumentasi)
- cognito-idp (dokumentasi)AWS CLI

Informasi tambahan

Contoh TypeScript berkas

Contoh kode berikut adalah TypeScript file yang memanggil proses otentikasi menggunakan AWS SDK untuk mengirim atribut kustom ke Amazon Cognito:

```
import * as AmazonCognitoIdentity from "amazon-cognito-identity-js";

const userPoolId: string = process.env.USER_POOL_ID ?? '';
const clientId: string = process.env.CLIENT_ID ?? '';

const poolData = {
  UserPoolId: userPoolId,
  ClientId: clientId
};
const userPool = new AmazonCognitoIdentity.CognitoUserPool(poolData);

export const loginWithCognitoSDK = function (userName: string, password: string) {
  const authenticationDetails = new AmazonCognitoIdentity.AuthenticationDetails({
    Username: userName,
    Password: password,
    ClientMetadata: {
        customGroup: "MyCustomGroup",
        customApplicationData: "Custom data from a custom application"
    }
  });
  const userData = {
    Username: userName,
    Pool: userPool
  };

  const cognitoUser = new AmazonCognitoIdentity.CognitoUser(userData);

  // Authenticate the user using the authenticationDetails object
  cognitoUser.authenticateUser(authenticationDetails, {
    onSuccess: function (result: any) {},
    onFailure: function (err: any) {},
  });
}
loginWithCognitoSDK(process.env.USERNAME ?? '', process.env.PASSWORD ?? '');
```

Sampel menggunakan `AuthenticationDetails` model dari SDK JavaScript untuk memberikan nama pengguna, kata sandi, dan file. `ClientMetadada` Setelah otentikasi di Amazon Cognito, metadata klien dapat diambil dari token akses dan ID.

Contoh fungsi Lambda

Contoh kode berikut adalah fungsi Lambda yang ditautkan ke Token Pra-Generasi dari Amazon Cognito. Ini membantu Anda menyesuaikan token akses dan token ID yang digunakan Amazon

Cognito. Token dilewatkan melalui integrasi antara arsitektur Anda. Contoh ini mencakup atribut klaim kustom yang dipanggil `customApplicationData` dan nama grup kustom yang disebut`MyCustomGroup`:

```
export const handler = async(event, context, callback) => {
    event.response = {
        claimsOverrideDetails: {
            claimsToAddOrOverride: { customApplicationData:
 event.request.clientMetadata.customApplicationData },
            groupOverrideDetails: { groupsToOverride:
 [event.request.clientMetadata.customGroup] }
        }
    };
    callback(null, event);
};
```

Contoh token akses

Anda dapat memecahkan kode token akses untuk memvisualisasikan atribut kustom yang ditambahkan. Berikut ini adalah contoh token akses:

```
{
  "sub": "6daf331f-4451-48b4-abde-774579299204",
  "cognito:groups": [
    "MyCustomGroup"
  ],
  "iss": "https://cognito-idp.<REGION>.amazonaws.com/<USERPOOL_ID>",
  "client_id": "<YOUR_CLIENT_ID>",
  "origin_jti": "acff7e91-09f9-4fde-8eec-38b0f8c47cdc",
  "event_id": "c5113a9c-1f01-435b-9b73-a5cd3e88514e",
  "token_use": "access",
  "scope": "aws.cognito.signin.user.admin",
  "auth_time": 1677979246,
  "exp": 1677982846,
  "iat": 1677979246,
  "jti": "5c9c2708-a871-4428-bd9b-18ad261bea90",
  "username": "<USER_NAME>"
}
```

Contoh ID token

Anda dapat memecahkan kode token akses untuk memvisualisasikan atribut kustom yang ditambahkan. Berikut ini adalah contoh token akses:

```
{
  "sub": "6daf331f-4451-48b4-abde-774579299204",
  "cognito:groups": [
    "MyCustomGroup"
  ],
  "iss": "https://cognito-idp.<REGION>.amazonaws.com/<USERPOOL_ID>",
  "cognito:username": "<USER_NAME>",
  "origin_jti": "acff7e91-09f9-4fde-8eec-38b0f8c47cdc",
  "customApplicationData": "Custom data from a custom application",
  "aud": "<YOUR_CLIENT_ID>",
  "event_id": "c5113a9c-1f01-435b-9b73-a5cd3e88514e",
  "token_use": "id",
  "auth_time": 1677979246,
  "exp": 1677982846,
  "iat": 1677979246,
  "jti": "f7ca006b-f25b-44d2-a7a4-6e6423f4201f",
  "email": "<USER_EMAIL>"
}
```

Sederhanakan manajemen sertifikat pribadi dengan menggunakan AWS Private CA dan AWS RAM

Everett Hinckley dan Vivek Goyal, Amazon Web Services

Ringkasan

Anda dapat menggunakan AWS Private Certificate Authority (AWS Private CA) untuk menerbitkan sertifikat pribadi untuk mengautentikasi sumber daya internal dan menandatangani kode komputer. Pola ini menyediakan CloudFormation template AWS untuk penerapan cepat hirarki CA multi-level dan pengalaman penyediaan yang konsisten. Secara opsional, Anda dapat menggunakan AWS Resource Access Manager (AWS RAM) untuk berbagi CA secara aman dalam organisasi atau unit organisasi (OUs) Anda di AWS Organizations, dan memusatkan CA saat menggunakan AWS RAM untuk mengelola izin. Tidak perlu CA pribadi di setiap akun, jadi pendekatan ini menghemat uang Anda. Selain itu, Anda dapat menggunakan Amazon Simple Storage Service (Amazon S3) untuk menyimpan daftar pencabutan sertifikat (CRL) dan mengakses log.

Implementasi ini menyediakan fitur dan manfaat berikut:

- Memusatkan dan menyederhanakan pengelolaan hierarki CA pribadi dengan menggunakan AWS Private CA.
- Mengekspor sertifikat dan kunci ke perangkat yang dikelola pelanggan di AWS dan di lokasi.
- Menggunakan CloudFormation template AWS untuk penerapan yang cepat dan pengalaman penyediaan yang konsisten.
- Membuat CA root pribadi bersama dengan hierarki CA bawahan 1, 2, 3, atau 4.
- Secara opsional, menggunakan AWS RAM untuk berbagi CA bawahan entitas akhir dengan akun lain di tingkat organisasi atau OU.
- Menghemat uang dengan menghilangkan kebutuhan akan CA pribadi di setiap akun dengan menggunakan AWS RAM.
- Membuat bucket S3 opsional untuk CRL.
- Membuat bucket S3 opsional untuk log akses CRL.

Prasyarat dan batasan

Prasyarat

Jika Anda ingin membagikan CA dalam struktur AWS Organizations, identifikasi atau siapkan hal berikut:

- Akun keamanan untuk membuat hierarki CA dan berbagi.
- OU atau akun terpisah untuk pengujian.
- Berbagi diaktifkan dalam akun manajemen AWS Organizations. Untuk informasi selengkapnya, lihat Mengaktifkan berbagi sumber daya dalam AWS Organizations dalam dokumentasi AWS RAM.

Batasan

- CAs adalah sumber daya regional. Semua CAs berada dalam satu akun AWS dan dalam satu Wilayah AWS.
- Sertifikat dan kunci yang dibuat pengguna tidak didukung. Untuk kasus penggunaan ini, kami menyarankan Anda menyesuaikan solusi ini untuk menggunakan CA root eksternal.
- Bucket CRL publik tidak didukung. Kami menyarankan Anda menjaga kerahasiaan CRL. Jika akses internet ke CRL diperlukan, lihat bagian tentang menggunakan Amazon CloudFront untuk melayani CRLs dalam Mengaktifkan fitur S3 Block Public Access (BPA) dalam dokumentasi AWS Private CA.
- Pola ini menerapkan pendekatan Single-region. Jika Anda memerlukan otoritas sertifikat Multi-wilayah, Anda dapat menerapkan bawahan di Wilayah AWS kedua atau di lokasi. Kompleksitas itu berada di luar cakupan pola ini, karena implementasinya bergantung pada kasus penggunaan spesifik Anda, volume beban kerja, dependensi, dan persyaratan.

Arsitektur

Tumpukan teknologi target

- AWS Private CA
- RAM AWS
- Amazon S3
- AWS Organizations
- AWS CloudFormation

Arsitektur target

Pola ini menyediakan dua opsi untuk berbagi ke AWS Organizations:

Opsi 1 – Buat bagian di tingkat organisasi. Semua akun dalam organisasi dapat menerbitkan sertifikat pribadi dengan menggunakan CA bersama, seperti yang ditunjukkan pada diagram berikut.

Opsi 2 – Buat bagian di tingkat unit organisasi (OU). Hanya akun di OU yang ditentukan yang dapat mengeluarkan sertifikat pribadi dengan menggunakan CA bersama. Misalnya, dalam diagram berikut, jika pembagian dibuat pada tingkat Sandbox OU, Pengembang 1 dan Pengembang 2 dapat menerbitkan sertifikat pribadi dengan menggunakan CA bersama.

Alat

Layanan AWS

- AWS Private CA — AWS Private Certificate Authority (AWS Private CA) adalah layanan CA pribadi yang dihosting untuk menerbitkan dan mencabut sertifikat digital pribadi. Ini membantu Anda membuat hierarki CA pribadi, termasuk root dan bawahan CAs, tanpa biaya investasi dan pemeliharaan pengoperasian CA lokal.
- AWS RAM — AWS Resource Access Manager (AWS RAM) membantu Anda berbagi sumber daya dengan aman di seluruh akun AWS dan di dalam organisasi Anda atau OUs di AWS Organizations. Untuk mengurangi overhead operasional di lingkungan multi-akun, Anda dapat membuat sumber daya dan menggunakan AWS RAM untuk membagikan sumber daya tersebut ke seluruh akun.
- AWS Organizations — AWS Organizations adalah layanan manajemen akun yang memungkinkan Anda mengkonsolidasikan beberapa akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon S3 - Amazon Simple Storage Service (Amazon S3) Simple Storage Service adalah layanan penyimpanan objek. Anda dapat menggunakan Amazon S3 untuk menyimpan dan mengambil data sebanyak apa pun kapan pun, dari mana pun di web. Pola ini menggunakan Amazon S3 untuk menyimpan daftar pencabutan sertifikat (CRL) dan log akses.
- AWS CloudFormation — AWS CloudFormation membantu Anda memodelkan dan menyiapkan sumber daya AWS Anda, menyediakannya dengan cepat dan konsisten, serta mengelolanya sepanjang siklus hidupnya. Anda dapat menggunakan template untuk mendeskripsikan sumber daya Anda dan dependensinya, dan meluncurkan dan mengonfigurasinya bersama-sama sebagai tumpukan, alih-alih mengelola sumber daya secara individual. Pola ini menggunakan AWS CloudFormation untuk secara otomatis menerapkan hierarki CA multi-level.

Kode

Kode sumber untuk pola ini tersedia di GitHub, di repositori hierarki AWS Private CA. Repositori meliputi:

- CloudFormation Template AWSACMPCA-RootCASubCA.yaml. Anda dapat menggunakan template ini untuk menerapkan hierarki CA untuk implementasi ini.
- Uji file untuk kasus penggunaan seperti meminta, mengekspor, mendeskripsikan, dan menghapus sertifikat.

Untuk menggunakan file-file ini, ikuti instruksi di bagian Epics.

Epik

Arsitek hierarki CA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kumpulkan informasi subjek sertifikat.	Kumpulkan informasi subjek sertifikat tentang pemilik sertifikat: nama organisasi, unit organisasi, negara, negara bagian, lokalitas, dan nama umum.	Arsitek cloud, Arsitek keamanan, insinyur PKI
Kumpulkan informasi opsional tentang AWS Organizations.	Jika CA akan menjadi bagian dari struktur AWS Organizations dan Anda ingin membagikan hierarki CA di dalam struktur tersebut, kumpulkan nomor akun manajemen, ID organisasi, dan secara opsional ID OU (jika Anda ingin berbagi hierarki CA hanya dengan OU tertentu). Juga, tentukan akun AWS Organizations atau OUs,	Arsitek cloud, Arsitek keamanan, insinyur PKI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	jika ada, yang ingin Anda bagikan CA.	
Desain hierarki CA.	Tentukan akun mana yang akan menampung root dan bawahan. CAs Tentukan berapa banyak level bawahan yang dibutuhka n hierarki antara root dan sertifikat entitas akhir. Untuk informasi selengkapnya, lihat Merancang hierarki CA dalam dokumentasi AWS Private CA.	Arsitek cloud, Arsitek keamanan, insinyur PKI
Tentukan konvensi penamaan dan penandaan untuk hierarki CA.	Tentukan nama untuk sumber daya AWS: root CA dan setiap CA bawahan. Tentukan tag mana yang harus ditetapkan untuk setiap CA.	Arsitek cloud, Arsitek keamanan, insinyur PKI
Tentukan enkripsi dan algoritma penandatanganan yang diperlukan.	Tentukan yang berikut: • Persyaratan algoritme enkripsi organisasi Anda untuk kunci publik yang digunakan CA Anda saat mengeluarkan sertifika t. Defaultnya adalah RSA_2048. • Algoritma kunci yang digunakan CA Anda untuk penandatanganan sertifikat. Defaultnya adalah SHA256 WITHRSA.	Arsitek cloud, Arsitek keamanan, insinyur PKI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tentukan persyaratan pencabutan sertifikat untuk hierarki CA.	Jika kemampuan pencabutan sertifikat diperlukan, buat konvensi penamaan untuk bucket S3 yang berisi daftar pencabutan sertifikat (CRL).	Arsitek cloud, Arsitek keamanan, insinyur PKI
Tentukan persyaratan logging untuk hierarki CA.	Jika kemampuan pencatatan akses diperlukan, buat konvensi penamaan untuk bucket S3 yang berisi log akses.	Arsitek cloud, Arsitek keamanan, insinyur PKI
Tentukan periode kedaluwarsa sertifikat.	Tentukan tanggal kedaluwarsa untuk sertifikat root (defaultnya adalah 10 tahun), sertifikat entitas akhir (defaultnya adalah 13 bulan), dan sertifikat CA bawahan (defaultnya adalah 3 tahun). Sertifikat CA bawahan harus kedaluwarsa lebih awal dari sertifikat CA pada tingkat yang lebih tinggi dalam hierarki. Untuk informasi selengkapnya, lihat Mengelola siklus hidup CA pribadi dalam dokumentasi AWS Private CA.	Arsitek cloud, Arsitek keamanan, insinyur PKI

Menyebarkan hierarki CA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Prasyarat lengkap.	Selesaikan langkah-langkah di bagian Prasyarat dari pola ini.	Administrator cloud, insinyur keamanan, insinyur PKI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat peran CA untuk berbagai persona.	1. Tentukan jenis peran AWS Identity and Access Management (IAM) atau pengguna di AWS IAM Identity Center (penerus AWS Single Sign-On) yang diperlukan untuk mengelola berbagai tingkatan hierarki CA, seperti Root, Subordinate, dan. CAAdmin CAAdmin CertificateConsumer 2. Tentukan perincian kebijakan yang diperlukan untuk memisahkan tugas. 3. Buat peran IAM atau pengguna yang diperlukan di Pusat Identitas IAM di akun tempat hierarki CA berada.	Administrator cloud, insinyur keamanan, insinyur PKI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan CloudFormation tumpukan.	1. Dari GitHub repositori untuk pola ini, unduh AWSPCA-RootCASubCA template. yaml. 2. Menerapkan template dari CloudFormation konsol AWS atau dari AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat Bekerja dengan tumpukan dalam CloudFormation dokumentasi. 3. Lengkapi parameter dalam template, termasuk nama organisasi, nama OU, algoritma kunci, algoritma penandatanganan, dan opsi lainnya.	Administrator cloud, insinyur keamanan, insinyur PKI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Arsitek solusi untuk memperbarui sertifikat yang digunakan oleh sumber daya yang dikelola pengguna.	Sumber daya layanan AWS terintegrasi, seperti Elastic Load Balancing, memperbarui sertifikat secara otomatis sebelum kedaluwar sa. Namun, sumber daya yang dikelola pengguna, seperti server web yang berjalan di instans Amazon Elastic Compute Cloud (Amazon EC2), memerlukan mekanisme lain. 1. Tentukan sumber daya yang dikelola pengguna mana yang memerlukan sertifikat entitas akhir dari CA pribadi. 2. Rencanakan proses untuk diberi tahu tentang berakhirnya sumber daya dan sertifikat yang dikelola pengguna. Untuk contoh , lihat yang berikut ini: • Menggunakan aturan terkelola AWS Config • Menggunakan Amazon CloudWatch dan Amazon EventBridge 3. Tulis skrip khusus untuk memperbarui sertifikat pada sumber daya yang dikelola pengguna dan menginteg rasikannya dengan layanan	Administrator cloud, insinyur keamanan, insinyur PKI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	AWS untuk mengotoma tiskan pembaruan. Untuk informasi selengkapnya tentang AWS Services terintegrasi, lihat Layanan yang terintegrasi dengan AWS Certificate Manager dalam dokumentasi ACM.	

Validasi dan dokumentasikan hierarki CA

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Validasi berbagi AWS RAM opsional.	Jika hierarki CA dibagikan dengan akun lain di AWS Organizations, masuk ke salah satu akun tersebut dari AWS Management Console, navigasikan ke konsol AWS Private CA, dan konfirmas ikan bahwa CA yang baru dibuat dibagikan ke akun ini. Hanya CA tingkat terendah dalam hierarki yang akan terlihat, karena itu adalah CA yang menghasilkan sertifika t entitas akhir. Ulangi untuk pengambilan sampel akun yang dibagikan CA.	Administrator cloud, insinyur keamanan, insinyur PKI
Validasi hierarki CA dengan pengujian siklus hidup sertifika t.	Dalam GitHub repositori untuk pola ini, cari tes siklus hidup. Jalankan pengujian dari AWS CLI untuk meminta sertifikat, mengekspor sertifika	Administrator cloud, insinyur keamanan, insinyur PKI

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	t, menjelaskan sertifikat, dan menghapus sertifikat.	
Impor rantai sertifikat ke toko kepercayaan.	Agar browser dan aplikasi lain mempercayai sertifika t, penerbit sertifikat harus dimasukkan dalam toko kepercayaan browser, yang merupakan daftar tepercaya CAs. Tambahkan rantai sertifikat untuk hierarki CA baru ke toko kepercayaan browser dan aplikasi Anda. Konfirmasikan bahwa sertifikat entitas akhir dipercaya.	Administrator cloud, insinyur keamanan, insinyur PKI
Buat runbook untuk mendokumentasikan hierarki CA.	Buat dokumen runbook untuk menjelaskan arsitektur hierarki CA, struktur akun yang dapat meminta sertifikat entitas akhir, proses pembuatan, dan tugas manajemen dasar seperti menerbitkan sertifikat entitas akhir (kecuali jika Anda ingin mengizinkan layanan mandiri menurut akun anak), penggunaan, dan pelacakan.	Administrator cloud, insinyur keamanan, insinyur PKI

Sumber daya terkait

- Merancang hierarki CA (dokumentasi AWS Private CA)
- Membuat CA pribadi (dokumentasi AWS Private CA)
- Cara menggunakan AWS RAM untuk membagikan akun silang AWS Private CA Anda (postingan blog AWS)

- Praktik terbaik AWS Private CA (postingan blog AWS)
- Aktifkan berbagi sumber daya dalam AWS Organizations (dokumentasi AWS RAM)
- Mengelola siklus hidup CA pribadi (dokumentasi AWS Private CA)
- acm-certificate-expiration-check untuk AWS Config (dokumentasi AWS Config)
- AWS Certificate Manager sekarang menyediakan pemantauan kedaluwarsa sertifikat melalui Amazon CloudWatch (pengumuman AWS)
- Layanan terintegrasi dengan AWS Certificate Manager (dokumentasi ACM)

Informasi tambahan

Saat Anda mengekspor sertifikat, gunakan frasa sandi yang kuat secara kriptografis dan selaras dengan strategi pencegahan kehilangan data organisasi Anda.

Merampingkan manajemen EC2 kepatuhan Amazon dengan agen Amazon Bedrock dan AWS Config

Anand Bukkapatnam Tirumala, Amazon Web Services

Ringkasan

Pola ini menjelaskan cara mengintegrasikan Amazon Bedrock dengan AWS Config aturan untuk memfasilitasi manajemen kepatuhan untuk instans Amazon Elastic Compute Cloud EC2 (Amazon). Pendekatan ini menggunakan kemampuan AI generatif canggih untuk memberikan rekomendasi yang disesuaikan yang selaras dengan AWS Well-Architected Framework, untuk memastikan pemilihan jenis instans yang optimal dan efisiensi sistem. Fitur utama dari pola ini meliputi:

- Pemantauan kepatuhan otomatis: AWS Config aturan terus menilai EC2 instans terhadap kriteria yang telah ditentukan untuk jenis instans yang diinginkan.
- Rekomendasi berbasis AI: Model AI generatif di Amazon Bedrock menganalisis pola infrastruktur. Model-model ini memberikan saran cerdas untuk perbaikan berdasarkan praktik terbaik yang diuraikan dalam Kerangka AWS Well-Architected.
- Remediasi: Grup tindakan Amazon Bedrock memungkinkan langkah-langkah remediasi otomatis untuk mengatasi instans yang tidak sesuai dengan cepat dan meminimalkan potensi kinerja atau inefisiensi biaya.
- Skalabilitas dan kemampuan beradaptasi: Solusi ini dirancang untuk disesuaikan dengan infrastruktur Anda dan beradaptasi dengan kebutuhan arsitektur cloud Anda yang terus berkembang.
- Rekomendasi keamanan yang ditingkatkan: Kepatuhan terhadap prinsip-prinsip AWS Well-Architected berkontribusi pada peningkatan postur keamanan dan kinerja sistem.

Anda dapat menggunakan pola ini sebagai cetak biru untuk menyebarkan infrastruktur berbasis AI generatif Anda sendiri ke beberapa lingkungan dengan perubahan minimal, menggunakan praktik seperlunya. DevOps

Prasyarat dan batasan

Prasyarat

- Aktif Akun AWS.

- Peran AWS Identity and Access Management (IAM) dengan izin untuk membuat dan mengelola sumber daya di bucket Amazon Simple Storage Service (Amazon S3),, fungsi, Amazon Bedrock AWS Config AWS Lambda , IAM, Amazon Logs, dan Amazon. CloudWatch EC2
- EC2 Instance untuk menandai sebagai tidak sesuai. Jangan gunakan `t2.small` tipe untuk contoh ini.
- Amazon Titan Text Embeddings V2 dan model Anthropic Claude 3 Haiku diaktifkan di. Akun AWS Untuk mengaktifkan akses model untuk Wilayah AWS tempat Anda menerapkan solusi, lihat Menambahkan atau menghapus akses ke model dasar Amazon Bedrock dalam dokumentasi Amazon Bedrock.
- Terraform, diinstal dan dikonfigurasi.
- AWS Command Line Interface (AWS CLI) v2 diinstal dan dikonfigurasi di lingkungan penerapan.
- Tinjauan lengkap tentang kebijakan AI Bertanggung Jawab Amazon.

Keterbatasan

- Beberapa Layanan AWS tidak tersedia di semua Wilayah AWS. Untuk ketersediaan Wilayah, lihat Layanan AWS berdasarkan Wilayah. Untuk titik akhir tertentu, lihat Titik akhir dan kuota layanan, dan pilih tautan untuk layanan.
- Solusi ini telah diuji dengan menggunakan model Amazon Titan Text Embeddings V2 dan Claude 3 Haiku. Jika Anda lebih suka menggunakan model lain, Anda dapat menyesuaikan kode Terraform, yang diparameterisasi untuk perubahan yang mudah.
- Solusi ini tidak menyertakan fitur riwayat obrolan, dan obrolan tidak disimpan.

Arsitektur

Diagram berikut menunjukkan alur kerja dan komponen arsitektur untuk pola ini.

Alur kerja terdiri dari langkah-langkah ini:

1. Pengguna berinteraksi dengan model melalui konsol obrolan Amazon Bedrock. Pengguna mengajukan pertanyaan seperti:
   - `What can you help me with?`
   - `List non-complaint resources`
   - `Suggest security best practices`

2. Jika model sudah dilatih sebelumnya, ia merespons petunjuk langsung dari pengetahuannya yang ada. Jika tidak, prompt masuk ke grup tindakan Amazon Bedrock.
3. Grup tindakan mencapai titik akhir VPC dengan menggunakan komunikasi layanan yang AWS PrivateLinkaman.
4. Permintaan mencapai fungsi Lambda melalui titik akhir VPC untuk layanan Amazon Bedrock.
5. Fungsi Lambda adalah mesin eksekusi utama. Berdasarkan permintaan, fungsi memanggil API untuk melakukan tindakan pada file Layanan AWS. Hal ini juga menangani operasi routing dan eksekusi.
6. Fungsi Lambda memanggil AWS Config untuk menentukan sumber daya non-keluhan (EC2 instance yang tidak sesuai yang Anda buat sebagai prasyarat).
7. AWS Config menandai sumber daya non-keluhan. Pola ini menerapkan AWS Config desired-instance-typeaturan untuk menemukan ukuran EC2 instans yang ideal.
8. AWS Config meminta pengguna untuk menjeda atau memulihkan instance, dan mengambil tindakan yang sesuai pada instance. EC2 Amazon Bedrock memahami muatan pengembalian ini.
9. Pengguna menerima respons di konsol obrolan Amazon Bedrock.

Otomatisasi dan skala

Solusi ini menggunakan Terraform sebagai alat infrastruktur sebagai kode (IAc) untuk memungkinkan penerapan yang mudah ke Akun AWS dan berfungsi sebagai utilitas mandiri di beberapa akun. Pendekatan ini menyederhanakan manajemen dan meningkatkan konsistensi dalam penerapan.

Alat

Layanan AWS

- AWS Configmemungkinkan Anda untuk menilai, mengaudit, dan mengevaluasi konfigurasi AWS sumber daya Anda untuk kepatuhan dan pengaturan yang diinginkan.
- Amazon Bedrock adalah layanan AI yang dikelola sepenuhnya yang menyediakan akses ke banyak model fondasi berkinerja tinggi melalui API terpadu.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Lambdaadalah layanan komputasi yang membantu Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Ini menjalankan kode Anda hanya bila diperlukan dan skala secara otomatis, jadi Anda hanya membayar untuk waktu komputasi yang Anda gunakan.

Alat-alat lainnya

- Git adalah sistem kontrol versi terdistribusi open source.
- Terraform adalah alat infrastruktur sebagai kode (IAc) HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal.

Repositori kode

Kode untuk pola ini tersedia di repositori GitHub sample-awsconfig-bedrock-compliance-manager.

Praktik terbaik

- Ikuti prinsip hak istimewa terkecil dan berikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Memberikan hak istimewa terkecil dan praktik terbaik Keamanan serta kasus penggunaan dalam dokumentasi IAM.
- Pantau log eksekusi Lambda secara teratur. Untuk informasi selengkapnya, lihat Memantau, men-debug, dan pemecahan masalah fungsi Lambda dan Praktik terbaik untuk bekerja dengan AWS Lambda fungsi dalam dokumentasi Lambda.

Epik

Terapkan solusinya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Untuk mengkloning repositor i untuk pola ini, gunakan perintah berikut: `git clone "git@gith ub.com:aws-samples /sample-awsconfig- bedrock-compliance- manager.git"`	AWS DevOps, Membangun pemimpin, DevOps insinyur, Administrator Cloud
Edit variabel lingkungan.	Di direktori root repositor i kloning pada mesin lokal Anda, edit file.	Administrator sistem AWS, AWS DevOps, DevOps insinyur, administrator AWS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	`terraform.tfvars` Tinjau placeholder yang ditandai dengan`[XXXXX]`, dan edit berdasarkan lingkungan Anda.	

Buat infrastruktur.	1. Untuk membuat infrastru ktur untuk solusi ini, jalankan perintah berikut:
	`terraform init`
	`terraform plan`
	2. Tinjau rencana eksekusi, dan kemudian jalankan perintah:
	`terraform apply -- auto-approve`
		AWS DevOps, DevOps insinyur, administrator sistem AWS, administrator Cloud

Gunakan agen

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Mengobrol dengan agen.	Menerapkan solusi di penerapan langkah sebelumny `asecurity-bot-agent` , yang merupakan agen Amazon Bedrock dengan konsol obrolan. Untuk menggunakan agen: 1. Masuk ke Konsol Manajemen AWS dengan identitas IAM yang memiliki izin untuk menggunakan	AWS DevOps, DevOps insinyur, administrator sistem AWS, administrator Cloud

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	konsol Amazon Bedrock. Kemudian, buka konsol Amazon Bedrock.
	2. Di panel navigasi, pilih Build, Agents. Pilih security-bot-agent.
	3. Mulailah berbicara dengan agen. Contoh:
	`List non-complaint EC2 instances`
	dan:
	`Replace these non-complaint EC2 instances with compliant instances`	

Pembersihan sumber daya

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hapus infrastruktur dan sumber daya.	Ketika Anda telah menyelesaikan pekerjaan Anda dengan solusi ini, Anda dapat menghapus infrastruktur yang dibuat oleh pola ini dengan menjalankan perintah: `terraform destroy --auto-approve`	AWS DevOps, DevOps insinyur, administrator sistem AWS, administrator Cloud

Pemecahan masalah

Isu	Solusi
Masalah perilaku agen	Untuk informasi pemecahan masalah, lihat Menguji dan memecahkan masalah perilaku agen di dokumentasi Amazon Bedrock.
AWS Lambda masalah jaringan	Untuk informasi selengkapnya, lihat Memecahkan masalah jaringan di Lambda dalam dokumentasi Lambda.
Izin IAM	Untuk informasi selengkapnya, lihat Memecahkan Masalah IAM di dokumentasi IAM.

Sumber daya terkait

- Agen Amazon Bedrock
- Gunakan grup tindakan untuk menentukan tindakan yang akan dilakukan agen Anda (dokumentasi Amazon Bedrock)
- desired-instance-type aturan (AWS Config dokumentasi)
- Cara AWS Config kerja (AWS Config dokumentasi)

Perbarui kredensyal AWS CLI dari AWS IAM Identity Center dengan menggunakan PowerShell

Cad Miles dan Andy Bowen, Amazon Web Services

Ringkasan

Jika Anda ingin menggunakan kredensil AWS IAM Identity Center (penerus AWS Single Sign-On) dengan AWS Command Line Interface (AWS CLI), AWS, atau AWS Cloud Development Kit (SDKsAWS CDK), Anda biasanya harus menyalin dan menempelkan kredensional dari konsol IAM Identity Center ke antarmuka baris perintah. Proses ini dapat memakan waktu yang cukup lama dan harus diulang untuk setiap akun yang memerlukan akses.

Salah satu solusi umum adalah dengan menggunakan perintah AWS CLI`aws sso configure`. Perintah ini menambahkan profil yang diaktifkan IAM Identity Center ke AWS CLI atau AWS SDK Anda. Namun, kelemahan dari solusi ini adalah Anda harus menjalankan perintah `aws sso login` untuk setiap profil AWS CLI atau akun yang telah Anda konfigurasikan dengan cara ini.

Sebagai solusi alternatif, pola ini menjelaskan cara menggunakan profil bernama AWS CLI dan AWS Tools untuk menyimpan dan menyegarkan kredensil PowerShell untuk beberapa akun dari satu instans Pusat Identitas IAM secara bersamaan. Skrip ini juga menyimpan data sesi Pusat Identitas IAM di memori untuk menyegarkan kredensyal tanpa masuk ke Pusat Identitas IAM lagi.

Prasyarat dan batasan

Prasyarat

- PowerShell, diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Menginstal PowerShell (dokumentasi Microsoft).
- AWS Tools untuk PowerShell, diinstal, dan dikonfigurasi. Untuk alasan kinerja, kami sangat menyarankan Anda menginstal AWS Tools versi termodulasi untuk PowerShell, yang disebut. `AWS.Tools` Setiap layanan AWS didukung oleh modul kecil individualnya sendiri. Dalam PowerShell prompt, masukkan perintah berikut untuk menginstal modul yang diperlukan untuk pola ini:`AWS.Tools.Installer`,`SSO`, dan`SSOIDC`.

```
Install-Module AWS.Tools.Installer
Install-AWSToolsModule SSO, SSOOIDC
```

Untuk informasi selengkapnya, lihat Menginstal AWS.Tools di Windows atau Menginstal AWS.Tools di Linux atau macOS.

- AWS CLI atau AWS SDK harus dikonfigurasi sebelumnya dengan kredensyal kerja dengan melakukan salah satu hal berikut:
  - Gunakan perintah AWS CLI`aws configure`. Untuk informasi selengkapnya, lihat Konfigurasi cepat (dokumentasi AWS CLI).
  - Konfigurasikan AWS CLI atau AWS CDK untuk mendapatkan akses sementara melalui peran IAM. Untuk informasi selengkapnya, lihat Mendapatkan kredensyal peran IAM untuk akses CLI (dokumentasi Pusat Identitas IAM).

Batasan

- Skrip ini tidak dapat digunakan dalam pipeline atau solusi yang sepenuhnya otomatis. Ketika Anda menyebarkan skrip ini, Anda harus secara manual mengotorisasi akses dari IAM Identity Center. Script kemudian berlanjut secara otomatis.

Versi produk

- Untuk semua sistem operasi, disarankan agar Anda menggunakan PowerShell versi 7.0 atau yang lebih baru.

Arsitektur

Anda dapat menggunakan skrip dalam pola ini untuk menyegarkan beberapa kredensional Pusat Identitas IAM secara bersamaan, dan Anda dapat membuat file kredensi untuk digunakan dengan AWS CLI, AWS, atau SDKs AWS CDK.

Alat

Layanan AWS

- AWS Command Line Interface (AWS CLI) adalah alat sumber terbuka yang membantu Anda berinteraksi dengan layanan AWS melalui perintah di shell baris perintah Anda.
- AWS IAM Identity Center membantu Anda mengelola akses masuk tunggal (SSO) secara terpusat ke semua akun AWS dan aplikasi cloud Anda.

- AWS Tools for PowerShell adalah sekumpulan PowerShell modul yang membantu Anda menjalankan skrip pada sumber daya AWS Anda dari baris PowerShell perintah.

Alat-alat lainnya

- PowerShelladalah program manajemen otomatisasi dan konfigurasi Microsoft yang berjalan di Windows, Linux, dan macOS.

Praktik terbaik

Simpan satu salinan skrip ini untuk setiap instance IAM Identity Center. Menggunakan satu skrip untuk beberapa instance tidak didukung.

Epik

Jalankan skrip SSO

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Sesuaikan skrip SSO.	1. Salin skrip SSO di bagian Informasi tambahan. 2. Di `Param` bagian ini, untuk lingkungan AWS Anda, tentukan nilai untuk variabel berikut: • `DefaultRoleName` — Peran IAM atau izin ditetapkan untuk digunakan secara default. • `Region`— Wilayah AWS tempat Pusat Identitas IAM digunakan. Untuk daftar lengkap Wilayah dan kodenya, lihat Titik akhir Regional.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `StartUrl`— URL yang digunakan untuk mengakses halaman login IAM Identity Center Anda. Gunakan format yang sama dengan nilai contoh dalam skrip. • `EnvironmentName` — Nama pendek untuk mereferensikan salinan skrip ini, yang akan digunakan saat Anda menjalankan beberapa salinan skrip dalam sesi yang sama. 3. Di bawah baris 10, yang berbunyi`# Add your Account Information` , edit nilai-nilai berikut dalam tabel hash untuk mencerminkan lingkungan Anda: • `Profile`— Nama profil AWS CLI untuk menyimpan kredensyal sementara. • `AccountId` — ID akun AWS tempat Anda mengambil kredensialnya. • `RoleName`— Nama peran IAM Identity Center atau set izin yang ingin	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Anda gunakan. Anda dapat membiarkan ini `$DefaultRoleName` seolah-olah Anda ingin menggunakan peran yang sama yang Anda tentukan di `Param` bagian ini. Setiap baris dalam tabel hash harus diakhiri dengan koma kecuali yang terakhir.	

Jalankan skrip SSO.	Disarankan agar Anda menjalankan skrip kustom Anda di PowerShell shell dengan perintah berikut.
	
	`./Set-AwsCliSsoCredentials.ps1`
	
	Atau, Anda dapat menjalankan skrip dari shell lain dengan memasukkan perintah berikut.
	
	`pwsh Set-AwsCliSsoCredentials.ps1`
		Administrator awan

Pemecahan masalah

Isu	Solusi
`No Access` kesalahan	Peran IAM yang Anda gunakan tidak memiliki izin untuk mengakses peran atau set izin yang Anda tetapkan dalam parameter. `RoleName`

Isu	Solusi
	Perbarui izin untuk peran yang Anda gunakan, atau tentukan peran atau izin yang berbeda yang ditetapkan dalam skrip.

Sumber daya terkait

- Di mana pengaturan konfigurasi disimpan? (Dokumentasi AWS CLI)
- Mengonfigurasi AWS CLI untuk menggunakan AWS IAM Identity Center (dokumentasi AWS CLI)
- Menggunakan profil bernama (dokumentasi AWS CLI)

Informasi tambahan

Skrip SSO

Dalam skrip berikut, ganti placeholder dalam kurung sudut (**<>**) dengan informasi Anda sendiri dan lepaskan tanda kurung sudut.

```
Set-AwsCliSsoCredentials.ps1
Param(
    $DefaultRoleName = '<AWSAdministratorAccess>',
    $Region          = '<us-west-2>',
    $StartUrl        = "<https://d-12345abcde.awsapps.com/start/>",
    $EnvironmentName = "<CompanyName>"
)
Try {$SsoAwsAccounts = (Get-Variable -name "$($EnvironmentName)SsoAwsAccounts" -Scope
 Global -ErrorAction 'SilentlyContinue').Value.Clone()}
Catch {$SsoAwsAccounts = $False}
if (-not $SsoAwsAccounts) { $SsoAwsAccounts = @(
# Add your account information in the list of hash tables below, expand as necessary,
 and do not forget the commas
    @{Profile = "<Account1>"      ; AccountId = "<012345678901 >"; RoleName =
 $DefaultRoleName },
    @{Profile = "<Account2>"      ; AccountId = "<123456789012>"; RoleName =
 "<AWSReadOnlyAccess>" }
)}
$ErrorActionPreference = "Stop"
if (-not (Test-Path ~\.aws))      { New-Item ~\.aws -type Directory }
if (-not (Test-Path ~\.aws\credentials)) { New-Item ~\.aws\credentials -type File }
```

```
$CredentialFile = Resolve-Path ~\.aws\credentials
$PsuedoCreds    = @{AccessKey =
 'AKAEXAMPLE123ACCESS';SecretKey='PsuedoS3cret4cceSSKey123PsuedoS3cretKey'} # Pseudo
 Creds, do not edit.
Try {$SSOTokenExpire = (Get-Variable -Scope Global -Name
 "$($EnvironmentName)SSOTokenExpire" -ErrorAction 'SilentlyContinue').Value} Catch
 {$SSOTokenExpire = $False}
Try {$SSOToken       = (Get-Variable -Scope Global -Name "$($EnvironmentName)SSOToken"
 -ErrorAction 'SilentlyContinue').Value }      Catch {$SSOToken       = $False}
if ( $SSOTokenExpire -lt (Get-Date) ) {
    $SSOToken = $Null
    $Client   = Register-SSOOIDCClient -ClientName cli-sso-client -ClientType public -
Region $Region @PsuedoCreds
    $Device   = $Client | Start-SSOOIDCDeviceAuthorization -StartUrl $StartUrl -Region
 $Region @PsuedoCreds
    Write-Host "A Browser window should open. Please login there and click ALLOW." -
NoNewline
    Start-Process $Device.VerificationUriComplete
    While (-Not $SSOToken){
        Try {$SSOToken = $Client | New-SSOOIDCToken -DeviceCode $Device.DeviceCode -
GrantType "urn:ietf:params:oauth:grant-type:device_code" -Region $Region @PsuedoCreds}
        Catch {If ($_.Exception.Message -notlike "*AuthorizationPendingException*")
{Write-Error $_.Exception} ; Start-Sleep 1}
    }
    $SSOTokenExpire = (Get-Date).AddSeconds($SSOToken.ExpiresIn)
    Set-Variable -Name "$($EnvironmentName)SSOToken" -Value $SSOToken -Scope Global
    Set-Variable -Name "$($EnvironmentName)SSOTokenExpire" -Value $SSOTokenExpire -
Scope Global
}
$CredsTime     = $SSOTokenExpire - (Get-Date)
$CredsTimeText = ('{0:D2}:{1:D2}:{2:D2} left on SSO Token' -f $CredsTime.Hours,
 $CredsTime.Minutes, $CredsTime.Seconds).TrimStart("0 :")
for ($i = 0; $i -lt $SsoAwsAccounts.Count; $i++) {
    if (([DateTimeOffset]::FromUnixTimeSeconds($SsoAwsAccounts[$i].CredsExpiration /
 1000)).DateTime -lt (Get-Date).ToUniversalTime()) {
        Write-host "`r
    `rRegistering Profile $($SsoAwsAccounts[$i].Profile)" -NoNewline
        $TempCreds = $SSOToken | Get-SSORoleCredential -AccountId
 $SsoAwsAccounts[$i].AccountId -RoleName $SsoAwsAccounts[$i].RoleName -Region $Region
 @PsuedoCreds
        [PSCustomObject]@{AccessKey = $TempCreds.AccessKeyId; SecretKey =
 $TempCreds.SecretAccessKey; SessionToken = $TempCreds.SessionToken
        } | Set-AWSCredential -StoreAs $SsoAwsAccounts[$i].Profile -ProfileLocation
 $CredentialFile
```

```
        $SsoAwsAccounts[$i].CredsExpiration = $TempCreds.Expiration
    }
}
Set-Variable -name "$($EnvironmentName)SsoAwsAccounts" -Value $SsoAwsAccounts.Clone() -
Scope Global
Write-Host "`r$($SsoAwsAccounts.Profile) Profiles registered, $CredsTimeText"
```

Gunakan Network Firewall untuk menangkap nama domain DNS dari Indikasi Nama Server untuk lalu lintas keluar

Kirankumar Chandrashekar, Amazon Web Services

Ringkasan

Pola ini menunjukkan cara menggunakan AWS Network Firewall untuk mengumpulkan nama domain DNS yang disediakan oleh Server Name Indication (SNI) di header HTTPS lalu lintas jaringan keluar Anda. Network Firewall adalah layanan terkelola yang memudahkan penerapan perlindungan jaringan penting untuk Amazon Virtual Private Cloud (Amazon VPC), termasuk kemampuan untuk mengamankan lalu lintas keluar dengan firewall yang memblokir paket yang gagal memenuhi persyaratan keamanan tertentu. Mengamankan lalu lintas keluar ke nama domain DNS tertentu disebut penyaringan jalan keluar, yang merupakan praktik pemantauan dan berpotensi membatasi aliran informasi keluar dari satu jaringan ke jaringan lainnya.

Setelah Anda menangkap data SNI yang melewati Network Firewall, Anda dapat menggunakan Amazon CloudWatch Logs dan AWS Lambda untuk mempublikasikan data ke topik Amazon Simple Notification Service (Amazon SNS) yang menghasilkan notifikasi email. Pemberitahuan email mencakup nama server dan informasi SNI terkait lainnya. Selain itu, Anda dapat menggunakan output dari pola ini untuk mengizinkan atau membatasi lalu lintas keluar berdasarkan nama domain di SNI dengan menggunakan aturan firewall. Untuk informasi selengkapnya, lihat Bekerja dengan grup aturan stateful di AWS Network Firewall dalam dokumentasi Network Firewall.

Prasyarat dan batasan

Prasyarat

- Akun AWS yang aktif.
- AWS Command Line Interface (AWS CLI) versi 2, diinstal dan dikonfigurasi di Linux, macOS, atau Windows.
- Network Firewall, diatur dan dikonfigurasi di Amazon VPC dan digunakan untuk memeriksa lalu lintas keluar. Anda dapat mengonfigurasi Network Firewall untuk menggunakan salah satu konfigurasi VPC berikut:
  - Arsitektur zona tunggal sederhana dengan gateway internet
  - Arsitektur multi zona dengan gateway internet

- Arsitektur dengan gateway internet dan gateway NAT

Arsitektur

Diagram berikut menunjukkan cara menggunakan Network Firewall untuk mengumpulkan data SNI dari lalu lintas jaringan keluar, dan kemudian mempublikasikan data tersebut ke topik SNS dengan menggunakan Log CloudWatch dan Lambda.

Diagram menunjukkan alur kerja berikut:

1. Network Firewall mengumpulkan nama domain dari data SNI di header HTTPS lalu lintas jaringan keluar Anda.
2. CloudWatch Log memonitor data SNI dan memanggil fungsi Lambda setiap kali lalu lintas jaringan keluar melewati Network Firewall.
3. Fungsi Lambda membaca data SNI yang diambil oleh CloudWatch Log dan kemudian menerbitkan data tersebut ke topik SNS.
4. Topik SNS mengirimi Anda notifikasi email yang menyertakan data SNI.

Otomatisasi dan skala

- Anda dapat menggunakan AWS CloudFormation untuk membuat pola ini dengan menggunakan infrastruktur sebagai kode.

Tumpukan teknologi

- CloudWatch Log Amazon
- Amazon SNS
- Amazon VPC
- AWS Lambda
- AWS Network Firewall

Alat

Layanan AWS

- CloudWatch Log Amazon — Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan mengakses file log Anda dari instans Amazon Elastic Compute Cloud (Amazon EC2) CloudTrail, AWS, Amazon Route 53, dan sumber lainnya.
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan (juga dikenal sebagai produsen dan konsumen).
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) menyediakan bagian AWS Cloud yang terisolasi secara logis tempat Anda dapat meluncurkan sumber daya AWS di jaringan virtual yang telah Anda tentukan. Jaringan virtual ini sangat mirip dengan jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaatnya yaitu menggunakan infrastruktur AWS yang dapat diskalakan.
- AWS Lambda — AWS Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server.
- AWS Network Firewall — AWS Network Firewall adalah layanan terkelola yang memudahkan penerapan perlindungan jaringan penting untuk semua Amazon Anda. VPCs

Epik

Buat grup CloudWatch log untuk Network Firewall

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat grup CloudWatch log.	1. Masuk ke AWS Management Console dan buka CloudWatch konsol. 2. Pada panel navigasi, pilih Grup log. 3. Pilih Actions (Tindakan), lalu pilih Create log group (Buat grup log). 4. Masukkan nama untuk grup log, lalu pilih Create log group (Buat grup log).	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	Untuk informasi selengkap nya, lihat Bekerja dengan grup log dan aliran log dalam CloudWatch dokumentasi.	

Buat topik dan langganan SNS

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Membuat sebuah topik SNS.	Untuk membuat topik SNS, ikuti petunjuk dalam dokumentasi Amazon SNS.	Administrator awan
Berlangganan titik akhir ke topik SNS.	Untuk berlangganan alamat email sebagai titik akhir ke topik SNS yang Anda buat, ikuti petunjuk dalam dokumentasi Amazon SNS. Untuk Protokol, pilih Email/ Email-JSON. **Note** Anda juga dapat memilih titik akhir yang berbeda berdasarkan kebutuhan Anda.	Administrator awan

Mengatur login di Network Firewall

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Aktifkan pencatatan firewall.	1. Masuk ke AWS Management Console dan buka konsol Amazon VPC. 2. Di panel navigasi, di bawah NETWORK FIREWALL, pilih Firewall. 3. Di bagian Firewall, pilih firewall tempat Anda ingin menangkap nama server dari SNI untuk lalu lintas keluar. 4. Pilih tab Detail Firewall, lalu pilih Edit di bagian Logging. 5. Untuk jenis Log, pilih Alert. Untuk tujuan Log untuk peringatan, pilih grup CloudWatch log. 6. Untuk grup CloudWatch log, cari dan pilih grup log yang Anda buat sebelumnya, lalu pilih Simpan. Untuk informasi selengkapnya tentang menggunakan CloudWatch Log sebagai tujuan log untuk Network Firewall, lihat CloudWatch Log Amazon dalam dokumentasi Network Firewall.	Administrator awan

Siapkan aturan stateful di Network Firewall

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat aturan stateful.	1. Masuk ke AWS Management Console dan buka konsol Amazon VPC. 2. Di panel navigasi, di bawah NETWORK FIREWALL, pilih Network Firewall Rule Groups. 3. Pilih Buat grup aturan Firewall Jaringan. 4. Pada halaman grup aturan Buat Network Firewall, untuk jenis grup Aturan, pilih Grup aturan stateful. Catatan: Untuk informasi selengkapnya, lihat Bekerja dengan grup aturan stateful di AWS Network Firewall. 5. Di bagian grup aturan stateful, masukkan nama dan deskripsi untuk grup aturan. 6. Untuk Kapasitas, atur kapasitas maksimum yang ingin Anda izinkan untuk grup aturan stateful (hingga maksimum 30.000). Catatan: Anda tidak dapat mengubah setelan ini setelah membuat grup aturan. Untuk informasi tentang cara menghitung kapasitas, lihat Menyetel	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	kapasitas grup aturan di AWS Network Firewall. Untuk informasi tentang pengaturan maksimum, lihat kuota AWS Network Firewall. 7. Untuk opsi grup aturan Stateful, pilih 5-tuple. 8. Di bagian urutan aturan Stateful, pilih Default. 9. Di bagian variabel Aturan, pertahankan nilai default. 10. Di bagian Tambah aturan, pilih TLS untuk Protokol. Untuk Sumber, pilih Any. Untuk port Sumber, pilih Port apa saja. Untuk Tujuan, pilih Any. Untuk port Tujuan, pilih Port apa saja. Untuk arah Lalu Lintas, pilih Maju. Untuk Tindakan, pilih Alert. Pilih Tambahkan aturan. 11. Pilih Buat grup aturan stateful.	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kaitkan aturan stateful ke Network Firewall.	1. Masuk ke AWS Management Console dan buka konsol Amazon VPC. 2. Di panel navigasi, di bawah NETWORK FIREWALL, pilih Firewall. 3. Pilih firewall tempat Anda ingin menangkap nama server dari SNI untuk lalu lintas keluar. 4. Di bagian Grup aturan stateful, pilih Tindakan, lalu pilih Tambahkan grup aturan stateful yang tidak dikelola. 5. Pada halaman Tambahkan grup aturan stateful yang tidak dikelola, pilih grup aturan stateful yang Anda buat sebelumnya, lalu pilih Tambahkan grup aturan stateful.	Administrator awan

Buat fungsi Lambda untuk membaca log

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Buat kode untuk fungsi Lambda.	Dalam lingkungan pengembangan terintegrasi (IDE) yang dapat membaca peristiwa CloudWatch Log dari Network Firewall untuk lalu lintas keluar, tempelkan kode	Pengembang aplikasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	Python 3 berikut dan <SNS-topic-ARN> ganti dengan nilai Anda:
	
	`import json`
	`import gzip`
	`import base64`
	`import boto3`
	`sns_client = boto3.client('sns')`
	`def lambda_handler(event, context):`
	`  decoded_event = json.loads(gzip.decompress(base64.b64decode(event['awslogs']['data'])))`
	`  body = '''`
	`  {filtermatch}`
	`  '''.format(`
	`  loggroup=decoded_event['logGroup'],`
	`  logstream=decoded_event['logStream'],`
	`  filtermatch=decoded_event['logEvents'][0]['message'],`
	`  )`
	`  print(body)`
	`  filterMatch = json.loads(body)`
	`  data = []`
	`  if 'http' in filterMatch['event']:`
	`  data.append(filterMatch['event']['http']['hostname'])`	

Tugas	Deskripsi	Keterampilan yang dibutuhkan

	<pre> elif 'tls' in
	 filterMatch['event']:
	        data.appe
	nd(filterMatch['ev
	ent']['tls']['sni'])
	    result = 'Domain
	 accessed ' + 1*' ' +
	 (data[0]) + 1*' ' 'via
	 AWS Network Firewall
	 ' + 1*' ' + (filterMa
	tch['firewall_name'])
	    print(result)
	    message = {'ServerN
	ame': result}
	    send_to_sns =
	 sns_client.publish(
	        TargetArn=<SNS-
	topic-ARN>,
	 #Replace with the SNS
	 topic ARN
	        Message=j
	son.dumps({'default':
	 json.dumps(message),
	
	    'sms': json.dump
	s(message),
	
	   'email': json.dump
	s(message)}),
	        Subject='Server
	 Name passed through the
	 Network Firewall',
	        MessageSt
	ructure='json'
	    )</pre> Contoh kode ini mem-parsing konten CloudWatch Log dan menangkap nama server yang	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	disediakan oleh SNI di header HTTPS.	
Buat fungsi Lambda.	Untuk membuat fungsi Lambda, ikuti instruksi dalam dokumentasi Lambda dan pilih Python 3.9 untuk Runtime.	Administrator awan
Tambahkan kode ke fungsi Lambda.	Untuk menambahkan kode Python Anda ke fungsi Lambda yang Anda buat sebelumnya, ikuti petunjuk dalam dokumentasi Lambda.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan CloudWatch Log sebagai pemicu ke fungsi Lambda.	1. Masuk ke AWS Management Console dan buka konsol Lambda. 2. Di panel navigasi, pilih Fungsi, lalu pilih fungsi yang Anda buat sebelumnya. 3. Di bagian Ikhtisar fungsi, pilih Tambahkan pemicu. 4. Pada halaman Tambah pemicu, di bagian konfigurasi Pemicu, pilih CloudWatch Log, lalu pilih Tambah. 5. Untuk grup Log, pilih grup CloudWatch log yang Anda buat sebelumnya. 6. Untuk nama Filter, masukkan nama untuk filter Anda. 7. Pilih Tambahkan. 8. Pada tab Konfigurasi halaman fungsi Anda, di bagian Pemicu, pilih pemicu yang baru saja Anda tambahkan, lalu pilih Aktifkan. Untuk informasi selengkapnya, lihat Menggunakan Lambda dengan CloudWatch Log di dokumentasi Lambda.	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Tambahkan izin publikasi SNS.	Tambahkan izin SNS:Publish ke peran eksekusi Lambda, sehingga Lambda dapat melakukan panggilan API untuk mempublikasikan pesan ke SNS. 1. Temukan peran eksekusi fungsi Lambda yang Anda buat sebelumnya. 2. Tambahkan kebijakan berikut ke peran AWS Identity and Access Management (IAM) Anda: `{ "Version": "2012-10-17", "Statement": [ { "Sid": "AllowSNSPublish", "Effect": "Allow", "Action": [ "sns:GetTopicAttributes", "sns:Subscribe", "sns:Unsubscribe", "sns:Publish" ], "Resource": "*" }`	Administrator awan

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	<pre>] }</pre>	

Uji fungsionalitas notifikasi SNS Anda

Tugas	Deskripsi	Keterampilan yang dibutuhkan

Kirim lalu lintas melalui Network Firewall.	1. Kirim atau tunggu lalu lintas HTTPS melewati Network Firewall.
	2. Periksa email notifikasi SNS yang Anda terima dari AWS saat lalu lintas melewati Network Firewall. Email tersebut mencakup detail SNI untuk lalu lintas keluar. Misalnya, email yang dihasilkan dari kode Lambda di atas akan memiliki konten berikut jika nama domain yang diakses https://aws.amazon.comdan protokol berlangganan adalah EMAIL-JSON:
	<pre>{
	    "Type": "Notification",
	    "MessageId": "<messageID>",
	    "TopicArn": "arn:aws:sns:us-west-2:123456789:testSNSTopic",</pre>
		Insinyur uji

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"Subject": "Server Name passed through the Network Firewall", "Message": "{\"ServerName\": \"Domain 'aws.amazon.com' accessed via AWS Network Firewall 'AWS-Network-Firewall-Multi-AZ-firewall\"}", "Timestamp": "2022-03-22T04:10:04.217Z", "SignatureVersion": "1", "Signature": "<Signature>", "SigningCertURL": "<SigningCertUrl>", "UnsubscribeURL": "<UnsubscribeURL>" } Kemudian, periksa log peringatan Network Firewall di Amazon CloudWatch dengan mengikuti instruksi dalam CloudWatch dokumentasi Amazon. Log peringatan menunjukkan output berikut: { "firewall_name": "AWS-Network-Firewall-Multi-AZ-firewall", "availability_zone": "us-east-2b",	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"event_timestamp": "<event timestamp>", "event": { "timestamp": "2021-03-22T04:10:04.214222+0000", "flow_id": <flow ID>, "event_type": "alert", "src_ip": "10.1.3.76", "src_port": 22761, "dest_ip": "99.86.59.73", "dest_port": 443, "proto": "TCP", "alert": { "action": "allowed", "signature_id": 2, "rev": 0, "signature": "", "category": "", "severity": 3 }, "tls": { "subject": "CN=aws.amazon.com", "issuerdn": "C=US, O=Amazon, OU=Server CA 1B, CN=Amazon", "serial": "<serial number>",	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	"fingerpr int": "<fingerprint ID>", "sni": "aws.amazon.com", "version": "TLS 1.2", "notbefor e": "2020-09-30T00:00: 00", "notafter ": "2021-09-23T12:00: 00", "ja3": {}, "ja3s": {} }, "app_proto": "tls" } }	

Gunakan Terraform untuk mengaktifkan Amazon secara otomatis GuardDuty untuk organisasi

Aarthi Kannan, Amazon Web Services

Ringkasan

Amazon GuardDuty terus memantau akun Amazon Web Services (AWS) Anda dan menggunakan intelijen ancaman untuk mengidentifikasi aktivitas tak terduga dan berpotensi berbahaya dalam lingkungan AWS Anda. Mengaktifkan secara manual GuardDuty untuk beberapa akun atau organisasi, di beberapa Wilayah AWS, atau melalui AWS Management Console dapat menjadi rumit. Anda dapat mengotomatiskan proses dengan menggunakan alat infrastruktur sebagai kode (IAc), seperti Terraform, yang dapat menyediakan dan mengelola multi-akun, layanan Multi-wilayah, dan sumber daya di cloud.

AWS merekomendasikan penggunaan AWS Organizations untuk menyiapkan dan mengelola beberapa akun GuardDuty. Pola ini mematuhi rekomendasi itu. Salah satu manfaat dari pendekatan ini adalah bahwa, ketika akun baru dibuat atau ditambahkan ke organisasi, GuardDuty akan diaktifkan secara otomatis di akun ini untuk semua Wilayah yang didukung, tanpa perlu intervensi manual.

Pola ini menunjukkan cara menggunakan HashiCorp Terraform untuk mengaktifkan Amazon GuardDuty untuk tiga atau lebih akun Amazon Web Services (AWS) dalam suatu organisasi. Kode sampel yang disediakan dengan pola ini melakukan hal berikut:

- Mengaktifkan GuardDuty semua akun AWS yang merupakan anggota organisasi target saat ini di AWS Organizations
- Mengaktifkan fitur Auto-Enable GuardDuty, yang secara otomatis mengaktifkan GuardDuty akun apa pun yang ditambahkan ke organisasi target di masa mendatang
- Memungkinkan Anda memilih Wilayah tempat Anda ingin mengaktifkan GuardDuty
- Menggunakan akun keamanan organisasi sebagai administrator yang GuardDuty didelegasikan
- Membuat bucket Amazon Simple Storage Service (Amazon S3) di akun logging dan GuardDuty mengonfigurasi untuk mempublikasikan temuan agregat dari semua akun di bucket ini
- Menetapkan kebijakan siklus hidup yang mentransisikan temuan dari bucket S3 ke penyimpanan Amazon S3 Glacier Flexible Retrieval setelah 365 hari, secara default

Anda dapat menjalankan kode sampel ini secara manual, atau Anda dapat mengintegrasikannya ke dalam pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) Anda.

Target audiens

Pola ini direkomendasikan untuk pengguna yang memiliki pengalaman dengan Terraform, Python, dan GuardDuty AWS Organizations.

Prasyarat dan batasan

Prasyarat

- Akun AWS aktif.
- Organisasi disiapkan di AWS Organizations, dan berisi setidaknya tiga akun berikut:
  - Akun manajemen — Ini adalah akun tempat Anda menerapkan kode Terraform, baik mandiri maupun sebagai bagian dari pipeline. CI/CD Status Terraform juga disimpan di akun ini.
  - Akun keamanan — Akun ini digunakan sebagai administrator yang GuardDuty didelegasikan. Untuk informasi selengkapnya, lihat Pertimbangan penting untuk administrator yang GuardDuty didelegasikan (dokumentasi)GuardDuty .
  - Akun logging — Akun ini berisi bucket S3 tempat GuardDuty mempublikasikan temuan agregat dari semua akun anggota.

  Untuk informasi selengkapnya tentang cara menyiapkan organisasi dengan konfigurasi yang diperlukan, lihat Membuat struktur akun (AWS Well-Architected Labs).
- Bucket Amazon S3 dan tabel Amazon DynamoDB yang berfungsi sebagai backend jarak jauh untuk menyimpan status Terraform di akun manajemen. Untuk informasi lebih lanjut tentang penggunaan backend jarak jauh untuk status Terraform, lihat Backend S3 (dokumentasi Terraform). Untuk contoh kode yang mengatur manajemen status jarak jauh dengan backend S3, lihat remote-state-s3-backend (Terraform Registry). Perhatikan persyaratan berikut:
  - Bucket S3 dan tabel DynamoDB harus berada di Region yang sama.
  - Saat membuat tabel DynamoDB, kunci partisi **`LockID`** harus (case-sensitive), dan tipe kunci partisi harus String. Semua pengaturan tabel lainnya harus pada nilai default mereka. Untuk informasi selengkapnya, lihat Tentang kunci utama dan Membuat tabel (dokumentasi DynamoDB).
- Bucket S3 yang akan digunakan untuk menyimpan log akses untuk bucket S3 yang GuardDuty akan mempublikasikan temuan. Untuk informasi selengkapnya, lihat Mengaktifkan pencatatan akses server Amazon S3 (dokumentasi Amazon S3). Jika Anda menerapkan ke zona landing zone

AWS Control Tower, Anda dapat menggunakan kembali bucket S3 di akun arsip log untuk tujuan ini.

- Terraform versi 0.14.6 atau yang lebih baru diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Memulai — AWS (dokumentasi Terraform).
- Python versi 3.9.6 atau yang lebih baru diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Rilis sumber (situs web Python).
- AWS SDK untuk Python (Boto3) diinstal. Untuk informasi selengkapnya, lihat Instalasi (dokumentasi Boto3).
- jq diinstal dan dikonfigurasi. Untuk informasi selengkapnya, lihat Unduh jq (dokumentasi jq).

Batasan

- Pola ini mendukung sistem operasi macOS dan Amazon Linux 2. Pola ini belum diuji untuk digunakan dalam sistem operasi Windows.

> **Note**
>
> Amazon Linux 2 mendekati akhir dukungan. Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs.

- GuardDuty harus belum diaktifkan di salah satu akun, di salah satu Wilayah target.
- Solusi IAc dalam pola ini tidak menerapkan prasyarat.
- Pola ini dirancang untuk landing zone AWS yang mematuhi praktik terbaik berikut:
  - Landing zone dibuat dengan menggunakan AWS Control Tower.
  - Akun AWS terpisah digunakan untuk keamanan dan pencatatan.

Versi produk

- Terraform versi 0.14.6 atau yang lebih baru. Kode sampel telah diuji untuk versi 1.2.8.
- Python versi 3.9.6 atau yang lebih baru.

Arsitektur

Bagian ini memberikan gambaran tingkat tinggi dari solusi ini dan arsitektur yang ditetapkan oleh kode sampel. Diagram berikut menunjukkan sumber daya yang digunakan di berbagai akun di organisasi, dalam satu Wilayah AWS.

1. Terraform membuat peran GuardDutyTerraformOrgRoleAWS Identity and Access Management (IAM) di akun keamanan dan akun logging.
2. Terraform membuat bucket S3 di Wilayah AWS default di akun logging. Bucket ini digunakan sebagai tujuan penerbitan untuk mengumpulkan semua GuardDuty temuan di semua Wilayah dan dari semua akun di organisasi. Terraform juga membuat kunci AWS Key Management Service (AWS KMS) di akun keamanan yang digunakan untuk mengenkripsi temuan di bucket S3 dan mengonfigurasi pengarsipan otomatis temuan dari bucket S3 ke dalam penyimpanan S3 Glacier Flexible Retrieval.
3. Dari akun manajemen, Terraform menetapkan akun keamanan sebagai administrator yang didelegasikan untuk. GuardDuty Ini berarti bahwa akun keamanan sekarang mengelola GuardDuty layanan untuk semua akun anggota, termasuk akun manajemen. Akun anggota individu tidak dapat menangguhkan atau menonaktifkan GuardDuty sendiri.
4. Terraform membuat GuardDuty detektor di akun keamanan, untuk administrator yang GuardDuty didelegasikan.
5. Jika belum diaktifkan, Terraform mengaktifkan perlindungan S3 di. GuardDuty Untuk informasi selengkapnya, lihat perlindungan Amazon S3 di Amazon GuardDuty (GuardDuty dokumentasi).
6. Terraform mendaftarkan semua akun anggota aktif saat ini di organisasi sebagai anggota. GuardDuty
7. Terraform mengonfigurasi administrator yang GuardDuty didelegasikan untuk mempublikasikan temuan agregat dari semua akun anggota ke bucket S3 di akun logging.
8. Terraform mengulangi langkah 3 hingga 7 untuk setiap Wilayah AWS yang Anda pilih.

Otomatisasi dan skala

Kode sampel yang disediakan dimodulasi sehingga Anda dapat mengintegrasikannya ke dalam CI/CD pipeline untuk penerapan otomatis.

Alat

Layanan AWS

- Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya yang menyediakan kinerja yang cepat, dapat diprediksi, dan terukur.
- Amazon GuardDuty adalah layanan pemantauan keamanan berkelanjutan yang menganalisis dan memproses log untuk mengidentifikasi aktivitas tak terduga dan berpotensi tidak sah di lingkungan AWS Anda.
- AWS Identity and Access Management (IAM) membantu Anda mengelola akses ke sumber daya AWS dengan aman dengan mengontrol siapa yang diautentikasi dan diberi wewenang untuk menggunakannya.
- AWS Key Management Service (AWS KMS) membantu Anda membuat dan mengontrol kunci kriptografi untuk melindungi data Anda.
- AWS Organizations adalah layanan manajemen akun yang membantu Anda mengkonsolidasikan beberapa akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat.
- Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek berbasis cloud yang membantu Anda menyimpan, melindungi, dan mengambil sejumlah data.
- AWS SDK for Python (Boto3) adalah kit pengembangan perangkat lunak yang membantu Anda mengintegrasikan aplikasi, pustaka, atau skrip Python Anda dengan layanan AWS.

Alat dan layanan lainnya

- HashiCorp Terraform adalah aplikasi antarmuka baris perintah yang membantu Anda menggunakan kode untuk menyediakan dan mengelola infrastruktur dan sumber daya cloud.
- Python adalah bahasa pemrograman tujuan umum.
- jq adalah prosesor baris perintah yang membantu Anda bekerja dengan file JSON.

Repositori kode

Kode untuk pola ini tersedia di GitHub, di organizations-with-terraform repositori amazon-guardduty-for-aws-.

Epik

Aktifkan GuardDuty dalam organisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Kloning repositori.	Dalam shell Bash, jalankan perintah berikut. Di Kloning repositori di bagian Informasi tambahan, Anda dapat menyalin perintah lengkap yang berisi URL repositor i. GitHub Ini mengkloning amazon-guardduty-for-aws-organizations-with-terraform repositori dari. GitHub `git clone <github-r epository-url>`	DevOps insinyur
Edit file konfigurasi Terraform.	1. Dalam `root` folder repositori kloning, replikasi file configuration.json .sample dengan menjalank an perintah berikut. `cp configuration.json .sample configura tion.json` 2. Edit file configuration.json baru, dan tentukan nilai untuk masing-masing variabel berikut: • `managemen t_acc_id` — ID Akun akun manajemen.	DevOps insinyur, AWS Umum, Terraform, Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	• `delegated_admin_acc_id` — ID Akun akun keamanan. • `logging_acc_id` — ID Akun dari akun logging. • `target_regions` — Daftar Wilayah AWS yang dipisahkan koma di mana Anda ingin mengaktifkan. GuardDuty • `organization_id` AWS Organizations ID dari organisasi tempat Anda mengaktifkan GuardDuty. • `default_region` — Wilayah tempat status Terraform Anda disimpan di akun manajemen . Ini adalah Wilayah yang sama tempat Anda menerapkan bucket S3 dan tabel DynamoDB untuk backend Terraform. • `role_to_assume_for_role_creation` — Nama yang ingin Anda tetapkan ke peran IAM baru di akun keamanan dan pencatatan. Anda membuat peran baru ini di cerita selanjutnya. Terraform	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	mengasumsikan peran ini untuk membuat peran `GuardDutyTerraformOrgRole` IAM dalam akun keamanan dan pencatatan. • `finding_publishing_frequency` — Frekuensi di mana GuardDuty menerbitkan temuan ke bucket S3. • `guardduty_findings_bucket_region` — Wilayah Pilihan tempat Anda ingin membuat bucket S3 untuk temuan yang dipublikasikan. • `logging_acc_s3_bucket_name` — Nama yang disukai untuk ember S3 untuk temuan yang dipublikasikan. • `security_acc_kms_key_alias` — Alias AWS KMS untuk kunci yang digunakan untuk mengenkripsi temuan. GuardDuty • `s3_access_log_bucket_name` — Nama bucket S3 yang sudah ada sebelumnya tempat Anda ingin mengumpulkan log	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akses untuk bucket S3 yang digunakan untuk temuan. GuardDuty Bucket ini harus berada di Wilayah AWS yang sama dengan bucket GuardDuty temuan. • `tfm_state_backend_s3_bucket` — Nama bucket S3 yang sudah ada sebelumnya untuk menyimpan status backend jarak jauh Terraform. • `tfm_state_backend_dynamodb_table` — Nama tabel DynamoDB yang sudah ada sebelumnya untuk mengunci status Terraform. 3. Simpan dan tutup file konfigurasi .	

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Hasilkan CloudFormation template untuk peran IAM baru.	Pola ini mencakup solusi IAc untuk membuat dua CloudFormation template. Template ini membuat dua peran IAM yang digunakan Terraform selama proses penyiapan. Template ini mematuhi praktik terbaik keamanan izin hak istimewa paling sedikit. 1. Di shell Bash, di `root` folder repositori, navigasikan ke. `cfn-templates/` Folder ini berisi file CloudFormation template dengan rintisan. 2. Jalankan perintah berikut. Ini menggantikan stub dengan nilai yang Anda berikan dalam file configuration.json. `bash scripts/replace_config_stubs.sh` 3. Konfirmasikan bahwa CloudFormation templat berikut telah dibuat di `cfn-templates/` folder: • management-account-role.yaml — File ini berisi definisi peran dan izin terkait untuk peran IAM di	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	akun manajemen, yang memiliki izin minimum yang diperlukan untuk menyelesaikan pola ini. • role-to-assume-for-role-creation.yaml — File ini berisi definisi peran dan izin terkait untuk peran IAM dalam akun keamanan dan logging. Terraform mengasumsikan peran ini untuk membuat peran dalam akun GuardDuty TerraformOrgRoleini.	
Buat peran IAM.	Mengikuti petunjuk dalam Membuat tumpukan (CloudFormation dokumentasi), lakukan hal berikut: 1. Terapkan tumpukan role-to-assume-for-role-creation.yaml di akun keamanan dan pencatatan. 2. Terapkan management-account-roletumpukan.yaml di akun manajemen . Saat Anda berhasil membuat tumpukan dan melihat status `CREATE_COMPLETE` tumpukan, dalam output, catat Amazon Resource Name (ARN) dari peran baru ini.	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Asumsikan peran IAM dalam akun manajemen.	Sebagai praktik terbaik keamanan, kami sarankan Anda mengambil peran management-account-roleIAM baru sebelum melanjutkan. Di AWS Command Line Interface (AWS CLI), masukkan perintah di Asumsikan peran IAM akun manajemen di bagian Informasi Tambahan.	DevOps insinyur, AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip pengaturan.	Di `root` folder repositori, jalankan perintah berikut untuk memulai skrip pengaturan. `bash scripts/full-setup.sh` Skrip full-setup.sh melakukan tindakan berikut: • Mengekspor semua nilai konfigurasi sebagai variabel lingkungan • Menghasilkan file kode backend.tf dan terraform .tfvars untuk setiap modul Terraform • Mengaktifkan akses tepercaya untuk GuardDuty dalam organisasi melalui AWS CLI. • Mengimpor negara organisasi ke negara bagian Terraform • Membuat bucket S3 untuk mempublikasikan temuan di akun logging • Membuat kunci AWS KMS untuk mengenkripsi temuan di akun keamanan • GuardDuty Aktifkan di seluruh organisasi, di semua Wilayah yang dipilih,	DevOps insinyur, Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	seperti yang dijelaskan di bagian Arsitektur	

(Opsional) Nonaktifkan GuardDuty di organisasi

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Jalankan skrip pembersihan.	Jika Anda menggunakan pola ini GuardDuty untuk mengaktifkan organisasi dan ingin menonaktifkan GuardDuty , di `root` folder repositori, jalankan perintah berikut untuk memulai skrip cleanup-gd.sh. `bash scripts/cleanup-gd.sh` Skrip ini dinonaktifkan GuardDuty di organisasi target, menghapus sumber daya apa pun yang diterapkan, dan mengembalikan organisasi ke status sebelumnya sebelum menggunakan Terraform untuk mengaktifkan. GuardDuty **Note** Skrip ini tidak menghapus file status Terraform atau mengunci file	DevOps insinyur, AWS Umum, Terraform, Python

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	dari backend lokal dan jarak jauh. Jika Anda perlu melakukan nya, Anda harus melakukan tindakan ini secara manual. Selain itu, skrip ini tidak menghapus organisasi yang diimpor atau akun yang dikelola olehnya. Akses tepercaya untuk GuardDuty tidak dinonaktifkan sebagai bagian dari skrip pembersihan.	
Hapus peran IAM.	Hapus tumpukan yang dibuat dengan templat role-to-assume-for-role-creation.yaml dan .yaml. management-account-role CloudFormation Untuk informasi selengkapnya, lihat Menghapus tumpukan (CloudFormation dokumentasi).	DevOps insinyur, AWS Umum

Sumber daya terkait

Dokumentasi AWS

- Mengelola beberapa akun (GuardDuty dokumentasi)
- Memberikan hak istimewa paling sedikit (dokumentasi IAM)

Pemasaran AWS

- Amazon GuardDuty
- AWS Organizations

Sumber daya lainnya

- Terraform
- Dokumentasi CLI Terraform

Informasi tambahan

Kloning repositori

Jalankan perintah berikut untuk mengkloning GitHub repositori.

```
git clone https://github.com/aws-samples/amazon-guardduty-for-aws-organizations-with-terraform
```

Asumsikan peran IAM akun manajemen

Untuk mengasumsikan peran IAM dalam akun manajemen, jalankan perintah berikut. Ganti `<IAM role ARN>` dengan ARN dari peran IAM.

```
export ROLE_CREDENTIALS=$(aws sts assume-role --role-arn <IAM role ARN> --role-session-name AWSCLI-Session --output json)
export AWS_ACCESS_KEY_ID=$(echo $ROLE_CREDENTIALS | jq .Credentials.AccessKeyId | sed 's/"//g')
export AWS_SECRET_ACCESS_KEY=$(echo $ROLE_CREDENTIALS | jq .Credentials.SecretAccessKey | sed 's/"//g')
export AWS_SESSION_TOKEN=$(echo $ROLE_CREDENTIALS | jq .Credentials.SessionToken | sed 's/"//g')
```

Verifikasi praktik terbaik operasional untuk PCI DSS 4.0 dengan menggunakan AWS Config

Tala Qraitem dan Alex Goff, Amazon Web Services

Ringkasan

Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS) menguraikan protokol teknis dan operasional penting untuk membantu melindungi data pembayaran. PCI DSS dikembangkan untuk mendorong dan meningkatkan keamanan data untuk akun kartu pembayaran. Ini juga memfasilitasi adopsi global langkah-langkah keamanan yang konsisten. Meskipun dirancang khusus untuk lingkungan dengan data akun kartu pembayaran, Anda dapat menggunakan PCI DSS untuk membantu melindungi dari ancaman dan mengamankan elemen lain dalam ekosistem pembayaran.

PCI DSS versi 4.0 dirilis untuk memenuhi persyaratan yang berkembang, memberikan klarifikasi atau panduan tambahan, dan meningkatkan struktur dan format standar. Untuk informasi selengkapnya tentang perubahan, lihat Ringkasan perubahan dari PCI DSS versi 3.2.1 ke 4.0.

Paket AWS Config kesesuaian adalah kumpulan AWS Config aturan dan tindakan remediasi yang membantu Anda membuat pemeriksaan tata kelola keamanan, operasional, atau pengoptimalan biaya. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal dalam Akun AWS dan Wilayah AWS, atau Anda dapat menerapkan di seluruh organisasi di. AWS Organizations

Paket kesesuaian untuk PCI DSS versi 4.0 ditambah dan dibangun di atas paket kesesuaian untuk versi 3.2.1. Aturan dalam peta paket kesesuaian dengan aturan dalam standar. Untuk informasi selengkapnya, lihat pemetaan yang disediakan di bagian Lampiran. Anda dapat memilih di antara dua versi paket kesesuaian ini: satu yang menyertakan jenis sumber daya global dan satu yang mengecualikan mereka.

> **Important**
>
> Paket kesesuaian tidak dirancang untuk sepenuhnya memastikan kepatuhan terhadap tata kelola atau standar kepatuhan tertentu. Anda bertanggung jawab untuk membuat penilaian sendiri tentang apakah penggunaan memenuhi persyaratan hukum dan peraturan yang berlaku.

Prasyarat dan batasan

Prasyarat

- Memiliki yang aktif Akun AWS.
- Mengatur AWS Config.
- Memenuhi prasyarat untuk paket kesesuaian.
- Menyebarkan paket kesesuaian PCI DSS versi 3.2.1.
- Memiliki izin untuk mengakses AWS Config dan mengelola paket kesesuaian. Untuk kebijakan contoh, lihat bagian Informasi tambahan dari pola ini.

Batasan

- Anda Akun AWS memiliki kuota default, sebelumnya disebut sebagai batas, untuk masing-masing. Layanan AWS Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah. Anda dapat meminta kenaikan untuk beberapa kuota, tetapi tidak semua kuota dapat ditingkatkan. Pastikan Anda terbiasa dengan batas AWS Config layanan, termasuk batasan untuk paket kesesuaian akun tunggal dan paket kesesuaian organisasi.
- Versi paket kesesuaian ini yang mencakup tipe sumber daya global ditujukan untuk penerapan hanya di Wilayah. `us-east-1`
- Versi paket kesesuaian ini yang mengecualikan jenis sumber daya global ditujukan untuk penerapan hanya di Wilayah berikut:
  - `ap-east-1`
  - `ap-south-1`
  - `ap-northeast-2`
  - `ap-southeast-1`
  - `ap-southeast-2`
  - `ap-northeast-1`
  - `ca-central-1`
  - `eu-central-1`
  - `eu-west-1`
  - `eu-west-2`
  - `eu-west-3`

- `eu-north-1`
- `sa-east-1`
- `us-east-2`
- `us-west-1`
- `us-west-2`

Alat

Layanan AWS

- AWS Configmemberikan tampilan rinci tentang sumber daya di Anda Akun AWS dan bagaimana mereka dikonfigurasi. Ini membantu Anda mengidentifikasi bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasinya telah berubah dari waktu ke waktu.
- AWS Systems Managermembantu Anda mengelola aplikasi dan infrastruktur yang berjalan di AWS Cloud. Ini menyederhanakan aplikasi dan manajemen sumber daya, mempersingkat waktu untuk mendeteksi dan menyelesaikan masalah operasional, dan membantu Anda mengelola AWS sumber daya Anda dengan aman dalam skala besar.

Repositori kode

Paket kesesuaian terletak di repositori paket AWS Config kesesuaian. GitHub Repositori ini berisi template berikut yang terkait dengan PCI DSS versi 4.0:

- Praktik Terbaik-Operasional-untuk-PCI-DSS-v4.0- .yaml including-global-resourcetypes
- Praktik Terbaik-Operasional-untuk-PCI-DSS-v4.0- .yaml excluding-global-resourcetypes

Epik

Menyebarkan dan mengelola paket kesesuaian

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Unduh paket kesesuaian.	Jika Anda menerapkan paket kesesuaian di `us-east-1` Wilayah, unduh template	DevOps insinyur

Tugas	Deskripsi	Keterampilan yang dibutuhkan
	operational-best-practices-for-PCI-DSS-v4.0 - .yaml. including-global-resourcetypes Jika Anda menerapkan paket kesesuaian di Wilayah yang berbeda, unduh template operational-best-practices-for-PCI-DSS-v4.0 - .yaml. excluding-global-resourcety pes	
(Opsional) Ubah paket kesesuaian.	Anda dapat memodifikasi template paket kesesuaia n untuk kebutuhan unik organisasi Anda. Misalnya, Anda dapat membuat tindakan remediasi khusus. Untuk informasi selengkapnya tentang cara membuat dan memodifikasi template, lihat Membuat template untuk paket kesesuaian kustom dalam dokumentasi. AWS Config	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
Menyebarkan paket kesesuaian.	Jika Anda menerapkan di target Akun AWS atau Wilayah AWS, ikuti petunjuk dalam Menerapkan paket kesesuaian dalam dokumentasi. AWS Config Anda dapat menggunakan Konsol Manajemen AWS atau AWS Command Line Interface (AWS CLI). Jika Anda menerapkan paket kesesuaian di seluruh organisasi AWS Organizations, ikuti petunjuk di Deploy AWS Config conformance pack menggunakan Quick Setup dalam dokumentasi. AWS Systems Manager	AWS Umum
(Opsional) Edit paket kesesuaian.	Jika Anda ingin mengedit paket kesesuaian, ikuti petunjuk dalam Mengedit paket kesesuaian dalam dokumentasi. AWS Config Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI.	AWS Umum

Tugas	Deskripsi	Keterampilan yang dibutuhkan
(Opsional) Hapus paket kesesuaian.	Jika Anda ingin menghapus paket kesesuaian, ikuti petunjuk di Menghapus paket kesesuaian dalam dokumentasi. AWS Config Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI.	AWS Umum

Sumber daya terkait

AWS sumber daya

- Paket kesesuaian untuk AWS Config (dokumentasi)AWS Config
- Menyebarkan paket AWS Config kesesuaian menggunakan Quick Setup (dokumentasi Systems Manager)
- Kepatuhan PCI DSS pada AWS(situs web)AWS
- PCI DSS versi 4.0 aktif AWS(Panduan Kepatuhan)

Sumber daya PCI DSS

- PCI DSS versi 4.0 Resource Hub
- Perpustakaan Dokumen Dewan Standar Keamanan PCI
- Ringkasan perubahan dari PCI DSS versi 3.2.1 ke 4.0

Informasi tambahan

Berikut ini adalah contoh kebijakan AWS Identity and Access Management (IAM) yang memungkinkan pengguna mengakses AWS Config dan mengelola paket kesesuaian:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "config:PutConfigRule",
                "config:PutConformancePack",
                "config:DeleteConfigRule",
                "config:DeleteRemediationConfiguration",
                "config:DeleteConformancePack",
                "config:PutRemediationConfigurations",
                "config:BatchGetAggregateResourceConfig",
                "config:BatchGetResourceConfig",
                "config:Get*",
                "config:Describe*",
                "config:Deliver*",
                "config:List*",
                "config:Select*"
            ],
            "Resource": "*"
        }
    ]
}
```

Lampiran

Untuk mengakses konten tambahan yang terkait dengan dokumen ini, unzip file berikut: attachment.zip

Lebih banyak pola

- Akses host bastion menggunakan Session Manager dan Amazon EC2 Instance Connect
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS Fargate, PrivateLink AWS, dan Network Load Balancer
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS PrivateLink dan Network Load Balancer
- Akses aplikasi kontainer secara pribadi di Amazon EKS menggunakan AWS PrivateLink dan Network Load Balancer
- Izinkan EC2 instance menulis akses ke bucket S3 di akun AMS
- Kaitkan AWS CodeCommit repositori di satu dengan Akun AWS Amazon SageMaker AI Studio Classic di akun lain
- Mengautentikasi pengguna aplikasi React yang ada dengan menggunakan Amazon Cognito dan AWS Amplify UI
- Otomatis menambahkan atau memperbarui entri registri Windows menggunakan AWS Systems Manager
- Otomatiskan penegakan enkripsi di AWS Glue menggunakan templat AWS CloudFormation
- Secara otomatis melampirkan kebijakan terkelola AWS untuk Systems Manager ke profil EC2 instans menggunakan Cloud Custodian dan AWS CDK
- Secara otomatis mengenkripsi volume Amazon EBS yang ada dan baru
- Blokir akses publik ke Amazon RDS dengan menggunakan Cloud Custodian
- Memusatkan resolusi DNS dengan menggunakan Microsoft Active AWS Managed Microsoft AD Directory dan lokal
- Menerapkan pemindaian Checkov kustom terpusat untuk menegakkan kebijakan sebelum menerapkan infrastruktur AWS
- Periksa EC2 instance untuk tag wajib saat peluncuran
- Mengonfigurasikan akses lintas akun ke Amazon DynamoDB
- Konfigurasikan enkripsi HTTPS untuk Oracle JD Edwards EnterpriseOne di Oracle WebLogic dengan menggunakan Application Load Balancer
- Konfigurasikan otentikasi TLS timbal balik untuk aplikasi yang berjalan di Amazon EKS
- Konfigurasikan otentikasi Windows untuk Amazon RDS untuk Microsoft SQL Server menggunakan AWS Managed Microsoft AD
- Connect dengan menggunakan terowongan SSH di pgAdmin

- Buat aplikasi React dengan menggunakan AWS Amplify dan tambahkan autentikasi dengan Amazon Cognito
- Buat laporan temuan Network Access Analyzer untuk akses internet masuk dalam beberapa Akun AWS
- Kustomisasi CloudWatch peringatan Amazon untuk AWS Network Firewall
- Terapkan ChatOps solusi untuk mengelola hasil pemindaian SAST dengan menggunakan Amazon Q Developer dalam tindakan kustom aplikasi obrolan dan CloudFormation
- Menyebarkan validasi keamanan pengkodean real-time dengan menggunakan server MCP dengan Kiro dan asisten pengkodean lainnya
- Dokumentasikan desain AWS landing zone
- Aktifkan koneksi terenkripsi untuk instans PostgreSQL DB di Amazon RDS
- Enkripsi Amazon RDS yang ada untuk instans DB Amazon RDS for PostgreSQL
- Menerapkan penandaan otomatis database Amazon RDS saat diluncurkan
- Menegakkan penandaan kluster EMR Amazon saat diluncurkan
- Pastikan pencatatan EMR Amazon ke Amazon S3 diaktifkan saat peluncuran
- Buat CloudFormation template AWS yang berisi aturan terkelola AWS Config menggunakan Troposfer
- Dapatkan notifikasi Amazon SNS saat status kunci kunci AWS KMS berubah
- Membantu menegakkan penandaan DynamoDB
- Identifikasi dan beri tahu kapan sumber daya Amazon Data Firehose tidak dienkripsi dengan kunci AWS KMS
- Menerapkan otentikasi berbasis Microsoft Entra ID dalam aplikasi mainframe modern AWS Blu Age
- Menerapkan otentikasi SAMP 2.0 untuk Amazon WorkSpaces dengan menggunakan Auth0 dan AWS Managed Microsoft AD
- Menerapkan SHA1 hashing untuk data PII saat bermigrasi dari SQL Server ke PostgreSQL
- Tingkatkan kinerja operasional dengan mengaktifkan Amazon DevOps Guru di beberapa Wilayah AWS, akun, dan OUs dengan AWS CDK
- Menyerap dan memigrasikan instans EC2 Windows ke akun AWS Managed Services
- Migrasikan Amazon RDS for Oracle ke Amazon RDS untuk PostgreSQL dalam mode SSL dengan menggunakan AWS DMS
- Migrasi Tumpukan ELK ke Elastic Cloud di AWS
- Migrasikan beban kerja F5 BIG-IP ke F5 BIG-IP VE di AWS Cloud

- Pantau Amazon Aurora untuk instance tanpa enkripsi
- Menyediakan peran IAM dengan hak istimewa paling sedikit dengan menerapkan solusi mesin penjual otomatis peran
- Mengamankan dan merampingkan akses pengguna dalam database federasi Db2 di AWS dengan menggunakan konteks tepercaya
- Kirim AWS WAF log ke Splunk dengan menggunakan AWS Firewall Manager dan Amazon Data Firehose
- Sajikan konten statis dalam bucket Amazon S3 melalui VPC dengan menggunakan Amazon CloudFront
- Siapkan end-to-end enkripsi untuk aplikasi di Amazon EKS menggunakan cert-manager dan Let's Encrypt
- Gunakan pengguna IDs dalam kebijakan IAM untuk kontrol akses dan otomatisasi
- Verifikasi bahwa penyeimbang beban ELB memerlukan penghentian TLS
- Lihat log dan metrik AWS Network Firewall dengan menggunakan Splunk
- Visualisasikan laporan kredensi IAM untuk semua akun AWS menggunakan Amazon Quick Sight

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.