Menerapkan Kerangka Kerja AWS Well-Architected untuk Amazon Neptune Analytics

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Menerapkan Kerangka Kerja AWS Well-Architected untuk Amazon Neptune Analytics

Table of Contents

Pengantar 1
Audiens yang dituju 2
Tujuan 2
Pilar keunggulan operasional 3
Mengotomatiskan penerapan menggunakan pendekatan IAc 3
Desain operasi 4
Buat perubahan yang sering, kecil, dan reversibel 5
Menerapkan observabilitas untuk wawasan yang dapat ditindaklanjuti 5
Belajar dari semua kegagalan operasional 6
Gunakan kemampuan logging untuk memantau aktivitas yang tidak sah atau anomali 6
Pilar keamanan 8
Menerapkan keamanan data 9
Amankan jaringan Anda 9
Menerapkan otentikasi dan otorisasi 10
Pilar keandalan 11
Memahami kuota layanan Neptunus 11
Memahami pola penyebaran Neptunus 12
Kelola dan skala cluster Neptunus 13
Kelola pencadangan dan peristiwa failover 14
Pilar efisiensi performa 15
Memahami pemodelan grafik untuk analitik 15
Optimalkan kueri 17
Optimalkan menulis 18
Grafik ukuran kanan 19
Pilar optimasi biaya 20
Memahami pola penggunaan dan layanan yang dibutuhkan 20
Pilih sumber daya dengan memperhatikan biaya 22
Pilar keberlanjutan 24
Pertimbangkan Wilayah AWS pilihan Anda 24
Optimalkan konsumsi 24
Optimalkan pengembangan perangkat lunak dan pola arsitektur 25
Sumber daya 27
Referensi 27
Posting blog dan video 27

Pelatihan 27
Riwayat dokumen 28
Glosarium 29
..... 29
A 30
B 33
C 35
D 38
E 42
F 44
G 46
H 47
I 48
L 51
M 52
O 57
P 59
Q 62
R 63
D 66
T 70
U 71
V 72
W 72
Z 73
..... lxxv

Menerapkan Kerangka Kerja AWS Well-Architected untuk Amazon Neptune Analytics

Michael Havey, Amazon Web Services ()AWS

Desember 2024 (riwayat dokumen)

Anda dapat membuat solusi berbasis grafik di Amazon Web Services (AWS) dengan menggunakan Amazon Neptunus. Neptunus mencakup Neptunus Analytics, mesin analisis grafik yang dioptimalkan untuk memori yang dapat dengan cepat menganalisis sejumlah besar data grafik untuk mendapatkan wawasan dan menemukan tren. Ini dapat melakukan analitik pada data di cluster database Neptunus yang ada, atau Anda dapat memuat dan menganalisis data dari kumpulan data eksternal. Panduan ini memberikan panduan preskriptif untuk menerapkan prinsip-prinsip AWS Well-Architected Framework saat Anda merencanakan penerapan Neptunus Analytics Anda. Menerapkan AWS Kerangka Well-Architected untuk Amazon Neptunus mencakup topik yang sama untuk database Neptunus.

AWS Well-Architected Framework membantu Anda membangun infrastruktur yang aman, berkinerja tinggi, tangguh, dan efisien untuk berbagai aplikasi dan beban kerja. Ini juga memberikan pendekatan yang konsisten bagi Anda untuk mengevaluasi arsitektur dan menerapkan desain yang dapat diskalakan.

Kerangka AWS Well-Architected dibangun di sekitar enam pilar berikut:

- Keunggulan operasional
- Keamanan
- Keandalan
- Efisiensi kinerja
- Optimalisasi biaya
- Keberlanjutan

Panduan ini memberikan informasi dari pilar desain Well-Architected Framework dan praktik terbaik, dan pertimbangan yang perlu diingat saat Anda menerapkan Neptunus Analytics. AWS

Audiens yang dituju

Panduan ini ditujukan untuk insinyur data, arsitek solusi, dan analis data yang merancang dan menerapkan solusi yang menggunakan grafik. AWS

Tujuan

Panduan ini dapat membantu Anda dan organisasi Anda melakukan hal berikut:

- Pilih dari opsi penerapan yang didukung.
- Ikuti pola desain AWS Well-Architected yang membantu Anda meningkatkan ketahanan dan keamanan.
- Rancang kueri Anda untuk kinerja optimal dan penghematan biaya.
- Pelajari cara menjadi efisien secara operasional saat mengelola grafik Neptunus Analytics Anda dalam produksi.

Pilar keunggulan operasional

Pilar keunggulan operasional dari AWS Well-Architected Framework berfokus pada menjalankan dan memantau sistem, dan terus meningkatkan proses dan prosedur. Ini mencakup kemampuan untuk mendukung pengembangan dan menjalankan beban kerja secara efektif, mendapatkan wawasan tentang operasi mereka, dan terus meningkatkan proses dan prosedur pendukung untuk memberikan nilai bisnis. Anda dapat mengurangi kompleksitas operasional melalui beban kerja penyembuhan diri, yang mendeteksi dan memperbaiki sebagian besar masalah tanpa campur tangan manusia. Anda dapat mencapai tujuan ini dengan mengikuti praktik terbaik yang dijelaskan di bagian ini, dan menggunakan metrik Amazon Neptunus Analytics APIs, serta mekanisme untuk merespons dengan benar ketika beban kerja Anda menyimpang dari perilaku yang diharapkan.

Diskusi pilar keunggulan operasional ini berfokus pada bidang-bidang utama berikut:

- Infrastruktur sebagai kode (IAc)
- Manajemen perubahan
- Strategi ketahanan
- Manajemen insiden
- Pelaporan audit untuk kepatuhan
- Pencatatan log dan pemantauan

Mengotomatiskan penerapan menggunakan pendekatan IAc

Praktik terbaik untuk mengotomatisasi penyebaran di Neptunus dengan menggunakan IAc meliputi:

- Terapkan IAc untuk menyebarkan grafik Neptunus Analytics dan sumber daya terkait. Untuk konfigurasi lingkungan yang konsisten, gunakan dukungan untuk Neptunus Analytics yang disediakan AWS CloudFormationoleh untuk menyediakan grafik dan titik akhir pribadi.
- Gunakan CloudFormation untuk menyediakan instans notebook Neptunus di Amazon AI. SageMaker Anda dapat menggunakan buku catatan untuk menanyakan dan memvisualisasikan data dalam grafik Neptunus Analytics.
- Saat Anda membuat grafik Neptune Analytics dari sumber yang ada, seperti cluster database Neptune atau snapshot, atau file data yang dipentaskan di Amazon Simple Storage Service (Amazon S3), pantau tugas impor massal.

- Mengotomatiskan prosedur operasional Neptunus Analytics, seperti mengubah ukuran grafik, menghapus dan memotret grafik, memulihkan grafik dari snapshot, dan mengatur ulang dan memuat ulang grafik. Gunakan Neptune Analytics API, yang tersedia melalui AWS CLI() atau AWS Command Line Interface SDK s.
- Nilai waktu aktif yang diperlukan dari grafik Anda. Analytics sering bersifat sementara; grafik hanya diperlukan untuk waktu yang Anda butuhkan untuk menjalankan algoritme. Jika ini masalahnya, gunakan AWS CLI atau SDKs untuk snapshot dan hapus grafik saat tidak lagi diperlukan. Anda kemudian dapat mengembalikannya dari snapshot nanti, jika perlu.
- Simpan string koneksi secara eksternal dari klien Anda. Anda dapat menyimpan string koneksi di AWS Secrets Manager, Amazon DynamoDB, atau lokasi mana pun di mana mereka dapat diubah secara dinamis.
- Gunakan tag untuk menambahkan metadata ke sumber daya Neptunus Analytics Anda, dan lacak penggunaan berdasarkan tag. Tag membantu mengatur sumber daya Anda. Misalnya, Anda dapat menerapkan tag umum ke sumber daya di lingkungan atau aplikasi tertentu. Anda juga dapat menggunakan tag untuk menganalisis penagihan penggunaan sumber daya; untuk informasi selengkapnya, lihat Mengatur dan melacak biaya menggunakan tag alokasi AWS biaya di Panduan Pengguna AWS Penagihan. Selain itu, Anda dapat menggunakan kondisi dalam kebijakan AWS Identity and Access Management (IAM) untuk mengontrol akses ke AWS sumber daya berdasarkan tag yang digunakan pada sumber daya tersebut. Anda dapat melakukan ini dengan menggunakan kunci kondisi `aws:ResourceTag/tag-key` global. Untuk informasi selengkapnya, lihat Mengontrol akses ke AWS sumber daya di Panduan Pengguna IAM.

Desain operasi

Mengadopsi pendekatan untuk meningkatkan cara Anda mengoperasikan grafik Neptunus Analytics:

- Pertahankan grafik Neptunus Analytics terpisah untuk pengembangan, pengujian, dan penggunaan produksi. Grafik ini mungkin memiliki kumpulan data, pengguna, dan kontrol operasional yang berbeda.
- Pertahankan grafik Neptunus Analytics terpisah untuk penggunaan yang berbeda. Misalnya, jika dua kelompok pengguna analitis memerlukan grafik terpisah dengan garis waktu, model, kinerja dan ketersediaan yang berbeda SLAs, dan pola penggunaan, pertahankan grafik terpisah untuk setiap grup.
- Mempersiapkan pengguna dan staf operasional untuk pembaruan pemeliharaan Neptunus Analytics.

Buat perubahan yang sering, kecil, dan reversibel

Rekomendasi berikut berfokus pada perubahan kecil dan reversibel yang dapat Anda lakukan untuk meminimalkan kompleksitas dan mengurangi kemungkinan gangguan beban kerja:

- Simpan templat dan skrip IAC dalam layanan kontrol sumber seperti GitHub atau. GitLab

> **Important**
>
> Jangan menyimpan AWS kredensil dalam kontrol sumber.

- Memerlukan penyebaran IAC untuk menggunakan layanan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) seperti atau. AWS CodeDeployAWS CodeBuild Kompilasi, uji, dan terapkan kode di lingkungan Neptunus Analytics non-produksi sebelum mempromosikannya ke grafik produksi.

Menerapkan observabilitas untuk wawasan yang dapat ditindaklanjuti

Dapatkan pemahaman yang komprehensif tentang perilaku beban kerja, kinerja, keandalan, biaya, dan kesehatan. Rekomendasi berikut membantu Anda mendapatkan tingkat pemahaman tersebut di Neptunus Analytics:

- Pantau CloudWatch metrik Amazon untuk Neptunus Analytics. Dari metrik ini, Anda dapat menentukan ukuran grafik (jumlah node, tepi, dan vektor, ditambah ukuran byte total), pemanfaatan CPU, dan permintaan kueri dan tingkat kesalahan.
- Buat CloudWatch dasbor dan alarm untuk metrik utama seperti`NumQueuedRequestsPerSec`,,, `NumOpenCypherRequestsPerSec` `GraphStorageUsagePercent``GraphSizeBytes`, dan `CPUUtilization` serta respons klien Neptunus yang ditemukan di log aplikasi Anda.
- Tetapkan notifikasi untuk memantau kesehatan grafik Neptunus Analytics seperti ketika ukuran grafik, tingkat permintaan, atau pemanfaatan CPU melebihi ambang batas Anda. Misalnya, jika `GraphStorageUsagePercent` telah naik menjadi 90 persen pada grafik yang ingin Anda tumbuhkan secara signifikan, putuskan apakah akan meningkatkan kapasitas Unit Kapasitas Neptunus (m-NCU) yang dioptimalkan untuk memori. Jika m-NCU saat ini 128, meningkatkannya menjadi 256 akan mengurangi penyimpanan sekitar 45 persen. Jika `NumQueuedRequestsPerSec` seringkali lebih besar dari nol, pertimbangkan untuk meningkatkan

kapasitas m-NCU untuk menyediakan lebih banyak kapasitas komputasi. Atau, Anda dapat mengurangi konkurensi sisi klien.

Belajar dari semua kegagalan operasional

Infrastruktur penyembuhan diri adalah upaya jangka panjang yang berkembang dalam iterasi karena masalah langka terjadi atau respons tidak seefektif yang diinginkan. Mengadopsi praktik-praktik berikut mendorong fokus ke arah tujuan itu:

- Mendorong peningkatan dengan belajar dari semua kegagalan.
- Bagikan apa yang dipelajari di seluruh tim dan organisasi. Jika beberapa tim dalam organisasi Anda menggunakan Neptunus, buat ruang obrolan umum atau grup pengguna untuk berbagi pembelajaran dan praktik terbaik.

Gunakan kemampuan logging untuk memantau aktivitas yang tidak sah atau anomali

Gunakan logging untuk mengamati kinerja anomali dan pola aktivitas. Pertimbangkan praktik terbaik berikut:

- Neptune Analytics mendukung pencatatan tindakan bidang kontrol dengan menggunakan. AWS CloudTrail Untuk informasi selengkapnya, lihat Mencatat panggilan API Neptune Analytics menggunakan. AWS CloudTrail Melalui kemampuan ini, Anda dapat melacak pembuatan, pembaruan, dan penghapusan sumber daya Neptunus Analytics. Untuk pemantauan dan peringatan yang kuat, Anda juga dapat mengintegrasikan CloudTrail peristiwa dengan Amazon CloudWatch Logs. Untuk meningkatkan analisis aktivitas layanan Neptunus Analytics dan mengidentifikasi perubahan aktivitas untuk Akun AWS sebuah, Anda dapat melakukan CloudTrail kueri log dengan menggunakan Amazon Athena. Misalnya, Anda dapat menggunakan kueri untuk mengidentifikasi tren dan mengisolasi aktivitas lebih lanjut berdasarkan atribut seperti alamat IP sumber atau pengguna.
- Anda juga dapat menggunakan CloudTrail untuk mengaktifkan pencatatan aktivitas bidang data Neptunus Analytics seperti eksekusi kueri. Anda dapat melihat kueri mana yang sedang dijalankan, frekuensinya, dan sumbernya. Secara default, CloudTrail tidak mencatat peristiwa data. Biaya tambahan berlaku untuk peristiwa data. Untuk informasi selengkapnya, lihat harga AWS CloudTrail.

- Anda juga dapat mencatat panggilan aplikasi Anda ke Neptunus Analytics baik di bidang kontrol atau bidang data. Misalnya, jika Anda menggunakan AWS SDK untuk Python (Boto3)untuk membuat kueri, Anda dapat mengaktifkan logging tingkat debug untuk mendapatkan jejak kueri ke konsol atau file. Ini berguna selama pengembangan. Kami juga menyarankan Anda menangkap dan mencatat pengecualian dari aplikasi Anda.

Pilar keamanan

Keamanan cloud adalah prioritas tertinggi di AWS. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan. Keamanan adalah tanggung jawab bersama antara Anda dan AWS. Model tanggung jawab bersama menggambarkan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang berjalan Layanan AWS di dalamnya AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas AWS keamanan sebagai bagian dari program AWS kepatuhan. Untuk mempelajari tentang program kepatuhan yang berlaku untuk Neptunus, AWS lihat Layanan dalam Lingkup berdasarkan Program Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh Layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. Untuk informasi selengkapnya tentang privasi data, lihat Privasi Data FAQs. Untuk informasi tentang perlindungan data di Eropa, lihat Model Tanggung Jawab AWS Bersama dan posting blog GDPR.

Pilar keamanan dari AWS Well-Architected Framework membantu Anda memahami cara menerapkan model tanggung jawab bersama saat Anda menggunakan Neptunus Analytics. Topik berikut menjelaskan cara mengonfigurasi Neptunus Analytics untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan yang lain Layanan AWS yang membantu Anda memantau dan mengamankan sumber daya Neptunus Analytics Anda. Pilar keamanan mencakup area fokus utama berikut:

- Keamanan data
- Keamanan jaringan
- Autentikasi dan otorisasi

Menerapkan keamanan data

Kebocoran data dan pelanggaran menempatkan pelanggan Anda pada risiko dan dapat menyebabkan dampak negatif yang besar pada perusahaan Anda. Praktik terbaik berikut membantu melindungi data pelanggan Anda dari paparan yang tidak disengaja dan berbahaya:

- Nama grafik, tag, peran IAM, dan metadata lainnya tidak boleh berisi informasi rahasia atau sensitif, karena data tersebut mungkin muncul di log penagihan atau diagnostik.
- URIs atau tautan ke server eksternal yang disimpan sebagai data di Neptunus tidak boleh berisi informasi kredensi untuk memvalidasi permintaan.
- Grafik Neptunus Analytics dienkripsi saat istirahat. Anda dapat menggunakan kunci default atau kunci AWS Key Management Service (AWS KMS) yang Anda pilih untuk mengenkripsi grafik. Anda juga dapat mengenkripsi snapshot dan data yang diekspor ke Amazon S3 selama impor massal. Anda dapat menghapus enkripsi saat impor selesai.
- Saat Anda menggunakan bahasa OpenCypher, praktikkan validasi input dan teknik parameterisasi yang tepat untuk mencegah injeksi SQL dan bentuk serangan lainnya. Hindari membuat kueri yang menggunakan penggabungan string dengan input yang disediakan pengguna. Gunakan kueri berparameter atau pernyataan yang disiapkan untuk meneruskan parameter input dengan aman ke database grafik. Untuk informasi selengkapnya, lihat Contoh kueri parameter OpenCypher dalam dokumentasi Neptunus.

Amankan jaringan Anda

Anda dapat mengaktifkan grafik Neptunus Analytics untuk konektivitas publik sehingga dapat dijangkau dari luar virtual private cloud (VPC). Konektivitas ini dinonaktifkan secara default. Grafik membutuhkan otentikasi IAM. Penelepon harus mendapatkan identitas dan memiliki izin untuk menggunakan grafik. Misalnya, untuk menjalankan kueri OpenCypher, penelepon harus membaca, menulis, atau menghapus izin pada grafik tertentu.

Anda juga dapat membuat titik akhir pribadi untuk grafik untuk mengakses grafik dari dalam VPC. Saat Anda membuat titik akhir, Anda menentukan VPC, subnet, dan grup keamanan untuk membatasi akses untuk memanggil grafik.

Untuk melindungi data Anda dalam perjalanan, Neptune Analytics memberlakukan koneksi SSL melalui HTTPS ke grafik. Untuk informasi selengkapnya, lihat Perlindungan data di Neptunus Analytics dalam dokumentasi Neptunus Analytics.

Menerapkan otentikasi dan otorisasi

Panggilan ke grafik Neptunus Analytics memerlukan otentikasi IAM. Penelepon harus mendapatkan identitas dan memiliki izin yang cukup untuk melakukan tindakan pada grafik. Untuk deskripsi tindakan API dan izin yang diperlukan, lihat dokumentasi Neptune Analytics API. Anda dapat menerapkan pemeriksaan kondisi untuk membatasi akses berdasarkan tag.

Autentikasi IAM menggunakan protokol AWS Signature Version 4 (SiGv4). Untuk menyederhanakan penggunaan dari aplikasi Anda, kami sarankan Anda menggunakan AWS SDK. Misalnya, dengan Python, gunakan klien Boto3 untuk Grafik Neptunus, yang mengabstraksi SigV4.

Saat Anda memuat data ke dalam grafik, pemuatan batch menggunakan kredenal IAM pemanggil. Penelepon harus memiliki izin untuk mengunduh data dari Amazon S3 dengan hubungan kepercayaan yang diatur sehingga Neptunus Analytics dapat mengambil peran untuk memuat data ke dalam grafik dari file Amazon S3.

Impor massal dapat dilakukan baik selama pembuatan grafik (oleh tim infrastruktur) atau pada grafik kosong yang ada (oleh tim rekayasa data yang memiliki izin untuk memulai tugas impor). Dalam kedua kasus tersebut, Neptune Analytics mengasumsikan peran IAM yang diberikan penelepon sebagai input. Peran ini memberikan izin untuk membaca dan mencantumkan konten folder Amazon S3 tempat data input dipentaskan.

Pilar keandalan

Pilar keandalan Well-Architected Framework mencakup kemampuan beban kerja untuk menjalankan fungsi yang dimaksudkan dengan benar dan konsisten ketika diharapkan. AWS Ini termasuk kemampuan untuk mengoperasikan dan menguji beban kerja melalui seluruh siklus hidupnya.

Beban kerja yang andal dimulai dengan desain perangkat lunak dan infrastruktur yang diputuskan sejak awal. Pilihan arsitektur Anda memengaruhi perilaku beban kerja Anda di semua pilar Well-Architected. Untuk keandalan, ada pola khusus yang harus Anda ikuti, seperti yang dibahas di bagian ini.

Pilar keandalan berfokus pada bidang-bidang utama berikut:

- Arsitektur beban kerja, termasuk kuota layanan dan pola penerapan
- Manajemen perubahan
- Manajemen kegagalan

Memahami kuota layanan Neptunus

AWS Akun Anda 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, tetapi tidak semua, kuota.

Untuk menemukan kuota untuk Neptune Analytics, buka konsol Service Quotas. Di panel navigasi, pilih Layanan AWS, lalu pilih Amazon Neptune Analytics. Perhatikan kuota pada jumlah grafik dan snapshot, memori maksimum yang disediakan untuk grafik, dan tingkat permintaan API.

Jika memori maksimum yang disediakan tidak cukup untuk kumpulan data Anda, nilai tipe node dan edge mana yang penting untuk penggunaan analitis yang Anda maksudkan. Muat subset data sehingga analitik dimungkinkan dalam kapasitas yang disediakan yang diizinkan. Banyak beban kerja analitik, terutama yang menjalankan algoritme grafik, hanya membutuhkan topologi dengan seperangkat properti terbatas, bukan grafik transaksional lengkap. (Untuk diskusi tentang perbedaan antara beban kerja transaksional dan analitis, lihat bagian Pilar efisiensi kinerja.)

Jika jumlah maksimum grafik tidak cukup untuk tujuan penggunaan Anda:

- Pertimbangkan untuk menggabungkan grafik yang memiliki kegunaan serupa.

- Nilai berapa banyak grafik yang harus dijalankan pada waktu tertentu. Jika Anda memiliki kasus penggunaan analitik singkat, snapshot dan hapus grafik saat tidak lagi diperlukan. Ini mengurangi jumlah grafik terhadap kuota.
- Pertimbangkan penyediaan grafik secara berbeda. Akun AWS

Memahami pola penyebaran Neptunus

Pahami poin keputusan berikut saat Anda berencana menerapkan grafik Neptunus Analytics:

- Pembibitan: Putuskan apakah akan membuat grafik kosong atau memuat data ke dalamnya pada waktu pembuatan dengan data dari Amazon S3, kluster basis data Neptunus yang ada, atau snapshot database Neptunus yang ada.

  Rekomendasi: Jika sumbernya adalah cluster atau snapshot Neptunus, Anda harus memuat datanya pada waktu pembuatan grafik. Jika sumbernya adalah Amazon S3, muat data pada waktu pembuatan jika upaya pemuatannya signifikan dan paling baik dilakukan sebagai aktivitas penyediaan infrastruktur. Jika Anda lebih suka memuat data sebagai rekayasa data atau aktivitas aplikasi, buat grafik kosong dan muat data dari Amazon S3 nanti.
- Kapasitas: Perkirakan kapasitas penyediaan yang diperlukan untuk grafik, mengingat ukuran data dan penggunaan aplikasi yang diharapkan.

  Rekomendasi: Pada waktu pembuatan, tentukan memori maksimum yang disediakan untuk membatasi ukuran grafik. Pengaturan ini wajib. Anda dapat mengubah kapasitas nanti jika perlu.
- Ketersediaan dan toleransi kesalahan: Putuskan apakah replika diperlukan untuk ketersediaan. Replika bertindak sebagai siaga hangat untuk pemulihan jika terjadi kegagalan grafik. Grafik dengan replika pulih lebih cepat daripada grafik tanpa replika. Juga pertimbangkan berapa lama grafik diperlukan, apakah itu hanya untuk analitik singkat, dan, jika demikian, kapan akan dihapus.

  Rekomendasi: Tentukan persyaratan ketersediaan—seperti berapa lama grafik tidak tersedia dan kapan dapat dihapus—sebelum Anda membuat grafik.
- Jaringan dan keamanan: Tentukan apakah Anda memerlukan konektivitas publik, konektivitas pribadi, atau keduanya, dan apakah Anda ingin mengenkripsi data Anda.

  Rekomendasi: Pahami persyaratan organisasi—seperti apakah konektivitas publik diizinkan dan di mana aplikasi klien grafik akan dikerahkan—sebelum Anda membuat grafik.

- Pencadangan dan pemulihan: Tentukan apakah snapshot harus dibuat, dan, jika demikian, kapan atau dalam kondisi apa. Pertimbangkan apakah organisasi Anda memiliki persyaratan pemulihan bencana (DR).

  Rekomendasi: Membuat snapshot adalah aktivitas manual. Putuskan kapan harus membuat snapshot dan pertimbangkan persyaratan DR Anda sebelum Anda membuat grafik.

Kelola dan skala cluster Neptunus

Grafik Neptunus Analytics terdiri dari satu instance yang dioptimalkan untuk memori. Kapasitas (m-NCU) dari instance diatur pada waktu pembuatan. Instance dapat diskalakan secara vertikal dengan meningkatkan kapasitas yang disediakan melalui tindakan administratif; kapasitas yang disediakan juga dapat dikurangi. Replika adalah target failover pasif, sehingga tidak meningkatkan skala grafik. Dalam hal ini, replika grafik berbeda dari replika baca database Neptunus, yang merupakan instance aktif dalam cluster Neptunus yang dapat memproses operasi baca dari aplikasi.

Replika dikenakan biaya. Replika diberi harga pada tingkat m-NCu dari grafik. Misalnya, jika grafik disediakan untuk 128 m-NCU dan memiliki replika tunggal, biayanya dua kali lipat dari grafik ekivalen yang tidak memiliki replika.

Dalam analitik, ada dua alasan utama untuk meningkatkan skala:

- Untuk menyediakan lebih banyak memori dan CPU untuk kueri analitik dan algoritme, karena kueri individual mahal, algoritma grafik yang dijalankan secara inheren kompleks dan membutuhkan lebih banyak sumber daya mengingat inputnya, atau tingkat permintaan bersamaan tinggi. Jika kueri semacam itu mengalami out-of-memory kesalahan, penskalaan adalah solusi yang masuk akal.
- Untuk mendukung ukuran grafik yang lebih besar dari yang Anda rencanakan. Misalnya, jika kapasitas yang disediakan saat ini adalah 128 m-NCU untuk mendukung 60 GB data sumber dan Anda memerlukan tambahan 40 GB data sumber, peningkatan menjadi 256 m-NCU dijamin.

Pantau CloudWatch metrik untuk Neptunus Analytics, `NumQueuedRequestsPerSec` seperti,,,`NumOpenCypherRequestsPerSec`,, `CPUUtilization` dan `GraphStorageUsagePercentGraphSizeBytes`, untuk menentukan apakah penskalaan diperlukan. Anda dapat memperbarui konfigurasi grafik melalui konsol, AWS CLI, atau SDKs. (Untuk contoh dan praktik terbaik, lihat bagian pilar keunggulan operasional.)

Kelola pencadangan dan peristiwa failover

Gunakan replika untuk memastikan bahwa grafik tetap tersedia jika terjadi kegagalan. Grafik menggunakan persistensi berbasis log untuk melakukan perubahan di seluruh Availability Zone dalam file. Wilayah AWS Replika bertindak sebagai siaga hangat dan memiliki akses ke data ini. Jika ada kegagalan, grafik melanjutkan operasi pada replika. Aplikasi terus menggunakan titik akhir yang sama untuk terhubung ke grafik. Permintaan dalam penerbangan selama kegagalan menghasilkan kesalahan dengan pengecualian layanan yang tidak tersedia. Pertimbangkan untuk menggunakan coba lagi dengan pola backoff dalam kode aplikasi untuk menangkap kesalahan, dan coba lagi setelah interval singkat. Permintaan baru yang dibuat selama failover diantrian dan mungkin mengalami latensi yang lebih lama.

Jika tidak ada replika yang dikonfigurasi dan grafik gagal, Neptunus Analytics pulih dari penyimpanan yang tahan lama, tetapi pemulihan membutuhkan waktu lebih lama karena Neptunus harus menginisialisasi ulang sumber daya.

Buat snapshot dari grafik. (Neptune Analytics tidak mengambil snapshot otomatis.) Jika grafik dimodifikasi secara teratur setelah pembuatan, sering-seringlah mengambil snapshot untuk menangkap statusnya saat ini. Hapus snapshot lama jika pemulihan ke titik waktu sebelumnya tidak diperlukan.

Anda dapat berbagi snapshot dengan akun lain dan di seberang Wilayah AWS. Jika Anda memiliki persyaratan DR, pertimbangkan apakah memulihkan grafik di Wilayah yang berbeda dari snapshot memenuhi persyaratan tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO) Anda.

Pilar efisiensi performa

Pilar efisiensi kinerja dari AWS Well-Architected Framework berfokus pada cara mengoptimalkan kinerja sambil menelan atau menanyakan data. Optimalisasi kinerja adalah proses tambahan dan berkelanjutan dari hal-hal berikut:

- Mengonfirmasi persyaratan bisnis
- Mengukur kinerja beban kerja
- Mengidentifikasi komponen yang berkinerja buruk
- Menyetel komponen untuk memenuhi kebutuhan bisnis Anda

Pilar efisiensi kinerja menyediakan pedoman khusus kasus penggunaan yang dapat membantu Anda mengidentifikasi model data grafik dan bahasa kueri yang tepat untuk digunakan. Ini juga mencakup praktik terbaik untuk diikuti saat memasukkan data ke dalam, dan mengkonsumsi data dari, Neptunus Analytics.

Pilar efisiensi kinerja berfokus pada bidang-bidang utama berikut:

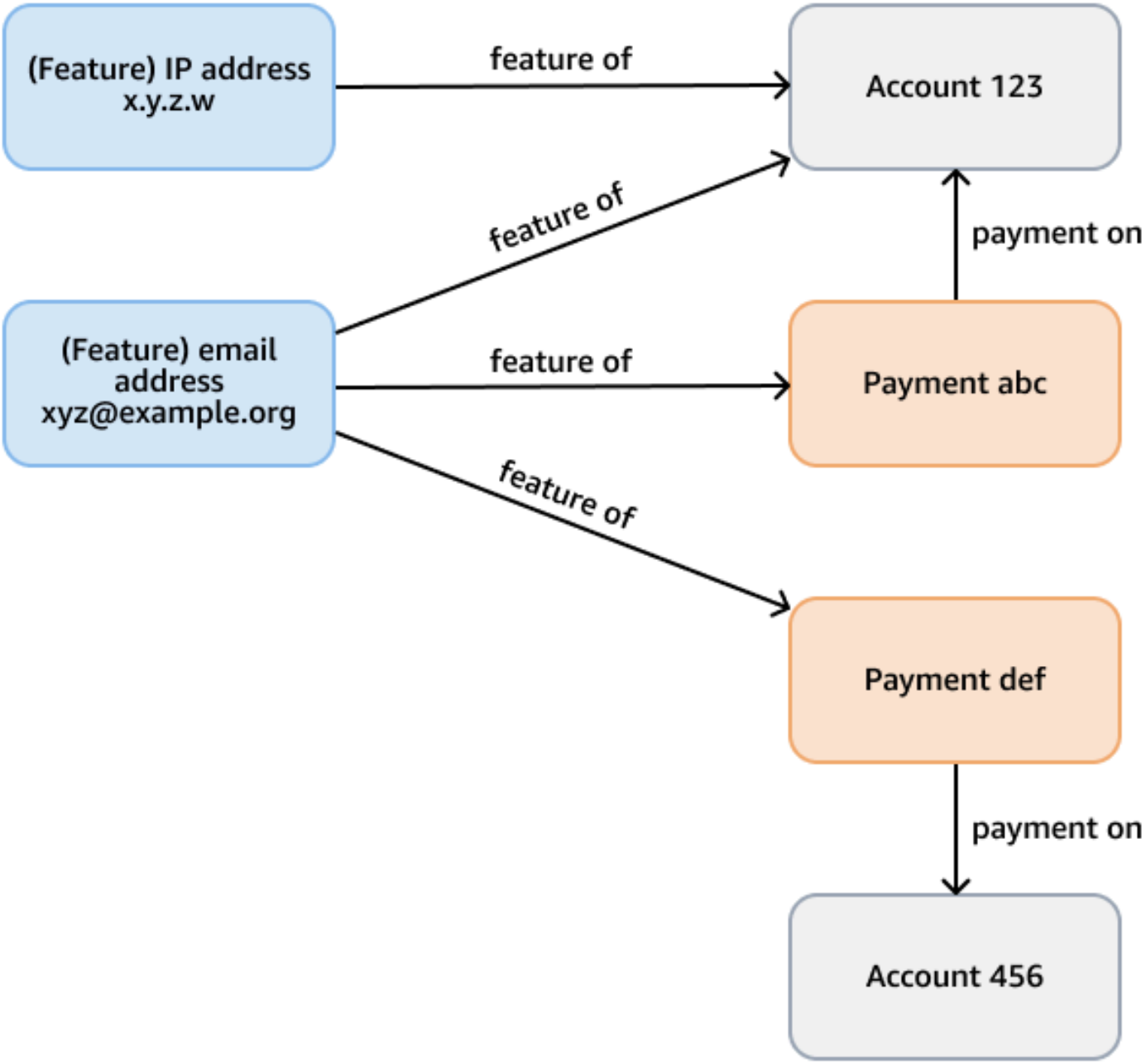
- Pemodelan grafik
- Pengoptimalan kueri
- Grafik ukuran kanan
- Tulis optimasi

Memahami pemodelan grafik untuk analitik

Panduan Menerapkan Kerangka AWS Well-Architected untuk Amazon Neptunus membahas pemodelan grafik untuk efisiensi kinerja. Keputusan pemodelan yang memengaruhi kinerja termasuk memilih node dan tepi mana yang diperlukan IDs, label dan propertinya, arah tepi, apakah label harus generik atau spesifik, dan umumnya seberapa efisien mesin kueri dapat menavigasi grafik untuk memproses kueri umum.

Pertimbangan ini juga berlaku untuk Neptunus Analytics; namun, penting untuk membedakan antara pola penggunaan transaksional dan analitis. Model grafik yang efisien untuk kueri dalam database transaksional seperti database Neptunus mungkin perlu dibentuk ulang untuk analitik.

Misalnya, pertimbangkan grafik penipuan dalam database Neptunus yang tujuannya adalah untuk memeriksa pola penipuan dalam pembayaran kartu kredit. Grafik ini mungkin memiliki node yang mewakili akun, pembayaran, dan fitur (seperti alamat email, alamat IP, nomor telepon) dari akun dan pembayaran. Grafik terhubung ini mendukung kueri seperti melintasi jalur panjang variabel yang dimulai dari pembayaran tertentu dan membutuhkan beberapa lompatan untuk menemukan fitur dan akun terkait. Gambar berikut menunjukkan grafik seperti itu.



Persyaratan analitis mungkin lebih spesifik, seperti menemukan komunitas akun yang ditautkan oleh fitur. Anda dapat menggunakan algoritma komponen yang terhubung lemah (WCC) untuk

tujuan ini. Untuk menjalankannya melawan model pada contoh sebelumnya tidak efisien, karena perlu melintasi beberapa jenis node dan tepi yang berbeda. Model dalam diagram berikutnya lebih efisien. Ini menghubungkan `account` node dengan `shares feature` tepi jika akun itu sendiri—atau pembayaran dari akun—berbagi fitur. Misalnya, `Account 123` memiliki fitur email`xyz@example.org`, dan `Account 456` menggunakan email yang sama untuk pembayaran (`Payment def`).

Kompleksitas komputasi WCC adalah`O(|E|logD)`, di mana `|E|` jumlah tepi dalam grafik, dan `D` merupakan diameter (panjang jalur terpanjang) yang menghubungkan node. Karena model transaksional menghilangkan node dan tepi yang tidak penting, ia mengoptimalkan jumlah tepi dan diameter, dan mengurangi kompleksitas algoritma WCC.

Saat Anda menggunakan Neptune Analytics, kerjakan kembali dari algoritme dan kueri analitis yang diperlukan. Jika perlu, bentuk kembali model untuk mengoptimalkan kueri ini. Anda dapat membentuk kembali model sebelum memuat data ke dalam grafik, atau menulis kueri yang mengubah data yang ada dalam grafik.

Optimalkan kueri

Ikuti rekomendasi ini untuk mengoptimalkan kueri Neptune Analytics:

- Gunakan kueri berparameter dan cache rencana kueri, yang diaktifkan secara default. Saat Anda menggunakan cache paket, mesin menyiapkan kueri untuk digunakan nanti — asalkan kueri selesai dalam 100 milidetik atau kurang — yang menghemat waktu pada pemanggilan berikutnya.
- Untuk kueri yang lambat, jalankan rencana penjelasan untuk menemukan kemacetan dan lakukan perbaikan yang sesuai.
- Jika Anda menggunakan pencarian kesamaan vektor, putuskan apakah penyematan yang lebih kecil menghasilkan hasil kesamaan yang akurat. Anda dapat membuat, menyimpan, dan mencari embeddings yang lebih kecil dengan lebih efisien.

- Ikuti praktik terbaik yang terdokumentasi untuk menggunakan OpenCypher di Neptune Analytics. Misalnya, gunakan peta yang diratakan dalam klausa UNWIND dan tentukan label tepi jika memungkinkan.
- Saat Anda menggunakan algoritma grafik, pahami input dan output algoritme, kompleksitas komputasinya, dan secara luas cara kerjanya.
  - Sebelum Anda memanggil algoritma grafik, gunakan `MATCH` klausa untuk meminimalkan set node input. Misalnya, untuk membatasi node untuk melakukan breadth-first search (BFS) dari, ikuti contoh yang disediakan dalam dokumentasi Neptunus Analytics.
  - Filter pada label simpul dan tepi jika memungkinkan. Misalnya, BFS memiliki parameter input untuk memfilter traversal ke label node tertentu (`vertexLabel`) atau label tepi tertentu (`edgeLabels`).
  - Gunakan parameter pembatas seperti `maxDepth` untuk membatasi hasil.
  - Eksperimen dengan `concurrency` parameter. Cobalah dengan nilai 0, yang menggunakan semua utas algoritma yang tersedia untuk memparalelkan pemrosesan. Bandingkan dengan eksekusi single-threaded dengan menyetel parameter ke 1. Algoritma dapat menyelesaikan lebih cepat dalam satu utas, terutama pada input yang lebih kecil seperti pencarian breadth-first dangkal di mana paralelisme tidak menawarkan pengurangan waktu eksekusi yang terukur dan mungkin menimbulkan overhead.
  - Pilih di antara jenis algoritma yang serupa. Misalnya, Bellman-Ford dan delta-stepping keduanya merupakan algoritma jalur terpendek sumber tunggal. Saat menguji dengan kumpulan data Anda sendiri, coba kedua algoritme dan bandingkan hasilnya. Delta-stepping seringkali lebih cepat daripada Bellman-Ford karena kompleksitas komputasi yang lebih rendah. Namun, kinerja tergantung pada dataset dan parameter input, terutama `delta` parameter.

Optimalkan menulis

Ikuti praktik berikut untuk mengoptimalkan operasi penulisan di Neptunus Analytics:

- Carilah cara paling efisien untuk memuat data ke dalam grafik. Saat Anda memuat dari data di Amazon S3, gunakan impor massal jika data berukuran lebih besar dari 50 GB. Untuk data yang lebih kecil, gunakan pemuatan batch. Jika Anda mendapatkan out-of-memory kesalahan saat menjalankan pemuatan batch, pertimbangkan untuk meningkatkan nilai m-NCU atau membagi beban menjadi beberapa permintaan. Salah satu cara untuk mencapai ini adalah dengan membagi file di beberapa awalan di bucket S3. Dalam hal ini, panggil pemuatan batch secara terpisah untuk setiap awalan.

- Gunakan impor massal atau pemuat batch untuk mengisi kumpulan data grafik awal. Gunakan operasi transaksional OpenCypher buat, perbarui, dan hapus hanya untuk perubahan kecil.
- Gunakan impor massal atau pemuat batch dengan konkurensi 1 (ulir tunggal) untuk menyerap penyematan ke dalam grafik. Cobalah untuk memuat embeddings di depan dengan menggunakan salah satu metode ini.
- Nilai dimensi penyematan vektor yang diperlukan untuk pencarian kesamaan yang akurat dalam algoritma pencarian kesamaan vektor. Gunakan dimensi yang lebih kecil jika memungkinkan. Ini menghasilkan kecepatan beban yang lebih cepat untuk penyematan.
- Gunakan algoritma mutasi untuk mengingat hasil algoritmik jika diperlukan. Misalnya, algoritma sentralitas mutasi derajat menemukan derajat setiap node input dan menulis nilai itu sebagai properti node. Jika koneksi di sekitar node tersebut tidak kemudian berubah, properti memegang hasil yang benar. Tidak perlu menjalankan algoritma lagi.
- Gunakan tindakan administratif reset grafik untuk menghapus semua node, tepi, dan embeddings jika Anda perlu memulai dari awal. Menjatuhkan semua node, tepi, dan penyematan dengan menggunakan kueri OpenCypher tidak layak jika grafik Anda besar. Kueri drop tunggal pada kumpulan data besar dapat habis waktu. Seiring bertambahnya ukuran, kumpulan data membutuhkan waktu lebih lama untuk dihapus dan ukuran transaksi meningkat. Sebaliknya, waktu untuk menyelesaikan reset grafik kira-kira konstan, dan tindakan menyediakan opsi untuk membuat snapshot sebelum Anda menjalankannya.

Grafik ukuran kanan

Kinerja keseluruhan tergantung pada kapasitas yang disediakan dari grafik Neptunus Analytics. Kapasitas diukur dalam satuan yang disebut Unit Kapasitas Neptunus yang dioptimalkan untuk memori (m -). NCUs Pastikan grafik Anda berukuran cukup untuk mendukung ukuran grafik dan kueri Anda. Perhatikan bahwa peningkatan kapasitas tidak selalu meningkatkan kinerja kueri individu.

Jika memungkinkan, buat grafik dengan mengimpor data dari sumber yang ada seperti Amazon S3 atau cluster atau snapshot Neptunus yang ada. Anda dapat menempatkan batas pada mininum dan kapasitas maksimum. Anda juga dapat mengubah kapasitas yang disediakan pada grafik yang ada.

Pantau CloudWatch metrik seperti`NumQueuedRequestsPerSec`,`NumOpenCypherRequestsPerSec`,`GraphStorageUsagePercent` dan `CPUUtilization` untuk menilai apakah grafik berukuran tepat. Tentukan apakah lebih banyak kapasitas diperlukan untuk mendukung ukuran dan beban grafik Anda. Untuk informasi selengkapnya tentang cara menafsirkan beberapa metrik ini, lihat bagian Pilar keunggulan operasional.

Pilar optimasi biaya

Pilar optimasi biaya dari AWS Well-Architected Framework berfokus pada menghindari biaya yang tidak perlu. Rekomendasi berikut dapat membantu Anda memenuhi prinsip desain pengoptimalan biaya dan praktik terbaik arsitektur untuk Neptunus Analytics.

Pilar optimasi biaya berfokus pada bidang-bidang utama berikut:

- Memahami pengeluaran dari waktu ke waktu dan mengendalikan alokasi dana
- Memilih sumber daya dari jenis dan kuantitas yang tepat
- Penskalaan untuk memenuhi kebutuhan bisnis tanpa pengeluaran berlebihan

Memahami pola penggunaan dan layanan yang dibutuhkan

Sebelum Anda mengadopsi Neptunus Analytics, periksa apakah kasus penggunaan Anda cocok untuk analitik grafik.

- Database grafik: Database grafik seperti Neptunus sangat cocok untuk beban kerja Anda jika model data Anda memiliki struktur grafik yang dapat dilihat dan kueri Anda perlu mengeksplorasi hubungan dan melintasi beberapa lompatan. Database grafik tidak cocok untuk pola berikut:
  - Terutama kueri single-hop. Dalam kasus penggunaan ini, pertimbangkan apakah data Anda mungkin lebih baik direpresentasikan sebagai atribut objek.
  - JSON atau data objek besar biner (gumpalan) disimpan sebagai properti.

- Analisis grafik: Neptunus Analytics adalah mesin database analisis grafik yang dapat dengan cepat menganalisis sejumlah besar data grafik dalam memori untuk mendapatkan wawasan dan menemukan tren. Anda dapat menyimpan dan menanyakan data grafik di database Neptunus dan grafik Neptunus Analytics. Database Neptunus paling cocok untuk kebutuhan pemrosesan transaksional online (OLTP) yang dapat diskalakan. Neptune Analytics adalah yang terbaik untuk beban kerja analitik sementara. Anda dapat menggunakan keduanya dalam kombinasi dengan memuat data dari database Neptunus yang berorientasi transaksi ke grafik Neptunus Analytics untuk menjalankan analitik data tersebut. Saat analisis selesai, Anda dapat menghapus grafik Neptunus Analytics. Untuk perbandingan yang lebih rinci, lihat Kapan menggunakan Neptunus Analytics dan kapan menggunakan Database Neptunus dalam dokumentasi Neptunus Analytics.

Tentukan, dengan memperhatikan biaya, cara terbaik untuk mengisi grafik Neptunus Analytics Anda.

- Data grafik impor massal yang dipentaskan dalam bucket S3. Kami merekomendasikan opsi ini jika data Anda sebelumnya dipentaskan untuk pemuatan massal ke database Neptunus, atau jika Anda sudah memiliki, atau dapat dengan mudah menghasilkan, data yang akan dianalisis dalam CSV atau format lain yang didukung yang diperlukan impor massal. Anda dapat menjalankan impor massal sebagai bagian dari prosedur pembuatan grafik. Anda dapat menempatkan batas pada kapasitas minimum dan maksimum. Anda juga dapat menjalankan impor pada grafik kosong yang dibuat sebelumnya dan memantau tugas impor saat berjalan.
- Anda dapat membuat grafik kosong dan kemudian mengisinya melalui kueri OpenCypher dengan menggunakan pemuatan batch. Opsi ini sangat ideal jika data yang akan dimuat dipentaskan di Amazon S3 dan lebih kecil dari 50 GB.
- Anda dapat mengisi grafik dari data di cluster database Neptunus Anda (didukung dalam Database Neptunus versi 1.3.0 atau yang lebih baru). Maksud dari pola ini adalah untuk menjalankan analitik pada data yang saat ini ada di database grafik Anda. Bahkan jika database awalnya diisi melalui beban massal, itu mungkin telah berubah secara signifikan sejak saat itu. Untuk mengimpor dari database, Neptune Analytics mengkloning database Anda dan mengekspor data dari klon ke bucket S3. Prosedur ini menimbulkan biaya: terutama biaya database Neptunus untuk menjalankan klon dan biaya Amazon S3 untuk menyimpan dan mengkonsumsi data yang diekspor. Klon dihapus ketika ekspor selesai. Anda dapat menghapus data yang diekspor di Amazon S3.
- Anda dapat mengisi grafik dari snapshot cluster database Neptunus. Ini mirip dengan opsi sebelumnya, kecuali bahwa sumbernya adalah snapshot database. Untuk mengimpor dari snapshot, Neptune Analytics pertama-tama mengembalikan snapshot ke cluster database baru, dan kemudian mengekspor data ke bucket S3. Prosedur ini menimbulkan biaya: terutama biaya database Neptunus untuk menjalankan cluster yang dipulihkan dan biaya Amazon S3 untuk menyimpan dan mengkonsumsi data yang diekspor.
- Anda juga dapat melakukan kueri OpenCypher untuk membuat, memperbarui, atau menghapus data dengan menggunakan transaksi yang sesuai dengan atomisitas, konsistensi, isolasi, daya tahan (ACID) pada grafik. Kami merekomendasikan pendekatan ini sebagai cara untuk membuat pembaruan kecil tetapi bukan sebagai cara penyemaian grafik.

Jika data yang diperlukan untuk analitik sudah dipentaskan di Amazon S3, kami merekomendasikan impor massal atau pemuatan batch. Ini lebih hemat biaya daripada mengisi grafik dari cluster atau snapshot database Neptunus.

Pilih sumber daya dengan memperhatikan biaya

Harga Neptunus Analytics menggunakan unit yang dikenal sebagai Unit Kapasitas Neptunus yang dioptimalkan untuk memori (m-NCU). Ada biaya per jam tetap untuk menjalankan grafik dengan m-NCU yang diberikan. Grafik mungkin memiliki replika untuk failover, dan replika ini juga dikenakan biaya m-NCU per jam.

Kami merekomendasikan praktik terbaik berikut untuk memperkirakan kapasitas, membatasi biaya, dan memantau biaya terhadap kinerja:

- Jika memungkinkan, buat grafik dengan mengimpor data dari sumber yang ada: data yang dipentaskan di Amazon S3 atau cluster atau snapshot Neptunus yang ada. Ini menghemat usaha Anda karena Neptunus Analytics melakukan pengangkatan berat penyemaian grafik, dan Anda dapat menentukan kapasitas maksimum terikat.
- Anda dapat mengubah kapasitas yang disediakan pada grafik yang ada.
- Ketika grafik tidak lagi diperlukan, Anda dapat membuat snapshot dan menghapus grafik. Jika Anda perlu menggunakannya lagi, Anda dapat mengembalikan grafik dari snapshot.
- Anda dapat memilih jumlah replika saat membuat grafik. Tetapkan nilai sesuai dengan persyaratan ketersediaan analitik Anda. Hemat biaya dengan meminimalkan pengaturan ini. Nilai maksimum 2 memungkinkan dua contoh replika di Availability Zone terpisah. Nilai minimum 0 berarti bahwa Neptunus Analytics tidak akan menjalankan replika. Namun, pemulihan lebih cepat ketika replika tersedia. Untuk penjelasan tentang kegagalan dan pemulihan grafik, lihat bagian pilar Keandalan.
- Pantau pengeluaran Neptunus Analytics untuk periode penagihan saat ini dan masa lalu dengan menggunakan. AWS Manajemen Penagihan dan Biaya
- Pantau metrik Neptunus Analytics CloudWatch untuk, `NumQueuedRequestsPerSec` terutama`NumOpenCypherRequestsPerSec`,,`GraphStorageUsagePercent`,`GraphSizeBytes`,, `CPUUtlization` dan, untuk menilai apakah kapasitas yang disediakan berukuran tepat untuk grafik. Tentukan apakah kapasitas yang lebih kecil dapat mengakomodasi tingkat permintaan yang diamati, penggunaan CPU, dan ukuran grafik.
- Jika Anda memerlukan titik akhir pribadi untuk grafik Anda, perhatikan biaya untuk titik akhir virtual private cloud (VPC) elastis IPs, gateway NAT, atau biaya terkait VPC lainnya. Untuk selengkapnya, lihat harga Amazon VPC dan harga Amazon EC2.
- Anda mungkin ingin menjalankan satu atau beberapa instance notebook Neptunus untuk menyediakan antarmuka klien guna membantu pengembang dan analis menanyakan dan memvisualisasikan grafik (lihat harga meja kerja Neptunus). Untuk meminimalkan biaya, bagikan

instance di antara pengguna dan buat folder notebook terpisah untuk setiap pengguna. Matikan instance saat tidak digunakan. Untuk pendekatan untuk mengotomatiskan shutdown, lihat posting AWS blog Mengotomatiskan penghentian dan awal sumber daya lingkungan Amazon Neptunus menggunakan tag sumber daya.

Pilar keberlanjutan

Pilar keberlanjutan dari AWS Well-Architected Framework berfokus pada meminimalkan dampak lingkungan dari menjalankan beban kerja cloud. Topik utama mencakup model tanggung jawab bersama untuk keberlanjutan, memahami dampak, dan memaksimalkan penggunaan untuk meminimalkan sumber daya yang dibutuhkan dan mengurangi dampak hilir.

Pilar keberlanjutan berisi area fokus utama berikut:

- Dampak Anda
- Tujuan keberlanjutan
- Penggunaan yang dimaksimalkan
- Mengantisipasi dan mengadopsi penawaran perangkat lunak baru yang lebih efisien
- Penggunaan layanan terkelola
- Pengurangan dampak hilir

Panduan ini berfokus pada pemahaman dampak Anda. Untuk informasi lebih lanjut tentang prinsip desain keberlanjutan lainnya, lihat Kerangka Kerja AWS Well-Architected.

Pilihan dan persyaratan Anda berdampak pada lingkungan. Jika Anda dapat memilih Wilayah AWS yang memiliki intensitas karbon lebih rendah, dan jika kebutuhan Anda mencerminkan kebutuhan beban kerja yang sebenarnya, bukan hanya memaksimalkan waktu kerja dan daya tahan, keberlanjutan beban kerja meningkat. Bagian selanjutnya membahas praktik dan pertimbangan terbaik yang akan memiliki dampak lingkungan yang positif jika diadopsi dalam desain beban kerja Anda dan operasi yang sedang berlangsung

Pertimbangkan Wilayah AWS pilihan Anda

Beberapa Wilayah AWS berada di dekat proyek energi terbarukan Amazon atau terletak di mana jaringan memiliki intensitas karbon yang diterbitkan yang lebih rendah daripada yang lain. Pertimbangkan dampak keberlanjutan untuk Wilayah yang mungkin layak untuk beban kerja Anda, dan rujuk silang daftar Anda dengan Wilayah tempat Analitik Neptunus tersedia.

Optimalkan konsumsi

Minimalkan konsumsi Neptunus Analytics dengan mempraktikkan hal-hal berikut:

- Analytics sering bersifat fana. Grafik diperlukan hanya untuk waktu untuk menjalankan algoritma dan mencatat hasilnya. Jika ini masalahnya, snapshot dan hapus grafik ketika tidak lagi diperlukan. Anda dapat mengembalikannya dari snapshot nanti jika perlu.
- Jika beban kerja bersifat sementara dan Anda memiliki fleksibilitas untuk memutuskan kapan menjalankan analitik, pertimbangkan day-to-day tren konsumsi daya. Permintaan listrik lebih tinggi selama waktu-waktu tertentu. Jika Anda berada di Amerika Serikat, lihat metrik konsumsi listrik harian di situs web Administrasi Informasi Energi AS (EIA). Jalankan beban kerja selama periode off-peak untuk Wilayah Anda jika memungkinkan.
- Jika beban kerja tidak bersifat sementara tetapi hanya perlu tersedia untuk periode terbatas, hapus grafik dan pulihkan dari snapshot saat diperlukan. Jika ketersediaannya mengikuti jadwal, otomatiskan proses restorasi melalui skrip sehingga grafik siap pada waktu yang dijadwalkan.
- Jika data hanya-baca atau tidak berubah sejak snapshot terakhir, jangan snapshot lagi sebelum dihapus.
- Hentikan notebook Neptunus saat tidak digunakan.
- Pantau CloudWatch metrik sepertiNumQueuedRequestsPerSec,NumOpenCypherRequestsPerSec,GraphStorageUsagePercer dan CPUUtilization untuk menilai apakah grafik terlalu besar. Tentukan apakah kapasitas instans yang lebih kecil dapat mengakomodasi tingkat permintaan yang diamati, penggunaan CPU, dan ukuran grafik.

Optimalkan pengembangan perangkat lunak dan pola arsitektur

Untuk mencegah pemborosan, optimalkan model dan kueri, dan bagikan sumber daya komputasi sehingga Anda menggunakan semua sumber daya yang tersedia di instans dan cluster Neptunus. Praktik terbaik khusus meliputi:

- Optimalkan kueri dan pemanggilan algoritma grafik. Gunakan kueri berparameter dan gunakan cache paket kueri, yang diaktifkan secara default. Untuk kueri yang lambat, jalankan rencana penjelasan untuk melakukan perbaikan. Jika Anda menggunakan pencarian kesamaan vektor, putuskan apakah penyematan yang lebih kecil menghasilkan hasil kesamaan yang akurat, karena penyematan yang lebih kecil dapat dibuat, disimpan, dan dicari dengan lebih efisien. Sebelum Anda memanggil algoritma grafik, gunakan MATCH klausa untuk meminimalkan set node input. Filter pada label simpul dan tepi jika memungkinkan.

- Carilah cara paling efisien untuk memuat data ke dalam grafik. Jika Anda memuat dari data di Amazon S3, gunakan impor massal jika data berukuran lebih besar dari 50 GB. Gunakan beban batch untuk data yang lebih kecil.
- Minta pengembang untuk membagikan instance notebook Neptunus alih-alih masing-masing membuat instance mereka sendiri. Buat folder notebook terpisah untuk setiap pengembang pada satu instance Jupyter. Matikan instance saat tidak digunakan.

Sumber daya

Referensi

- AWS Well-Architected
- AWS Dokumentasi Kerangka Well-Architected
- Menerapkan Kerangka Kerja AWS Well-Architected untuk Amazon Neptunus
- Praktik terbaik Neptunus Analytics

Posting blog dan video

- Posting blog Neptunus (Blog AWS Database)
- Otomatiskan penghentian dan dimulainya sumber daya lingkungan Amazon Neptunus menggunakan tag sumber daya (Blog Database)AWS
- Camilan Amazon Neptunus (video pendek) YouTube

Pelatihan

- Memulai dengan Amazon Neptunus
- Membangun dengan Amazon Neptunus
- Pemodelan Data untuk Amazon Neptunus
- Lokakarya Analisis Amazon Neptunus

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Publikasi awal	—	Desember 20, 2024

AWS Glosarium Panduan Preskriptif

Berikut ini adalah istilah yang umum digunakan dalam strategi, panduan, dan pola yang disediakan oleh Panduan AWS Preskriptif. Untuk menyarankan entri, silakan gunakan tautan Berikan umpan balik di akhir glosarium.

Nomor

7 Rs

Tujuh strategi migrasi umum untuk memindahkan aplikasi ke cloud. Strategi ini dibangun di atas 5 Rs yang diidentifikasi Gartner pada tahun 2011 dan terdiri dari yang berikut:

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di. AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instance EC2 di. AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Contoh: Migrasikan Microsoft Hyper-V aplikasi ke AWS.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

- Pensiun — Menonaktifkan atau menghapus aplikasi yang tidak lagi diperlukan di lingkungan sumber Anda.

A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

Metode migrasi database di mana database sumber dan target tetap sinkron (dengan menggunakan alat replikasi dua arah atau operasi penulisan ganda), dan kedua database menangani transaksi dari menghubungkan aplikasi selama migrasi. Metode ini mendukung migrasi dalam batch kecil yang terkontrol alih-alih memerlukan pemotongan satu kali. Ini lebih fleksibel tetapi membutuhkan lebih banyak pekerjaan daripada migrasi aktif-pasif.

migrasi aktif-pasif

Metode migrasi database di mana database sumber dan target disimpan dalam sinkron, tetapi hanya database sumber yang menangani transaksi dari menghubungkan aplikasi sementara data direplikasi ke database target. Basis data target tidak menerima transaksi apa pun selama migrasi.

fungsi agregat

Fungsi SQL yang beroperasi pada sekelompok baris dan menghitung nilai pengembalian tunggal untuk grup. Contoh fungsi agregat meliputi `SUM` dan`MAX`.

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

Proses menghapus informasi pribadi secara permanen dalam kumpulan data. Anonimisasi dapat membantu melindungi privasi pribadi. Data anonim tidak lagi dianggap sebagai data pribadi.

anti-pola

Solusi yang sering digunakan untuk masalah berulang di mana solusinya kontra-produktif, tidak efektif, atau kurang efektif daripada alternatif.

kontrol aplikasi

Pendekatan keamanan yang memungkinkan penggunaan hanya aplikasi yang disetujui untuk membantu melindungi sistem dari malware.

portofolio aplikasi

Kumpulan informasi rinci tentang setiap aplikasi yang digunakan oleh organisasi, termasuk biaya untuk membangun dan memelihara aplikasi, dan nilai bisnisnya. Informasi ini adalah kunci untuk penemuan portofolio dan proses analisis dan membantu mengidentifikasi dan memprioritaskan aplikasi yang akan dimigrasi, dimodernisasi, dan dioptimalkan.

kecerdasan buatan (AI)

Bidang ilmu komputer yang didedikasikan untuk menggunakan teknologi komputasi untuk melakukan fungsi kognitif yang biasanya terkait dengan manusia, seperti belajar, memecahkan masalah, dan mengenali pola. Untuk informasi lebih lanjut, lihat Apa itu Kecerdasan Buatan?

operasi kecerdasan buatan (AIOps)

Proses menggunakan teknik pembelajaran mesin untuk memecahkan masalah operasional, mengurangi insiden operasional dan intervensi manusia, dan meningkatkan kualitas layanan. Untuk informasi selengkapnya tentang cara AIOps digunakan dalam strategi AWS migrasi, lihat panduan integrasi operasi.

enkripsi asimetris

Algoritma enkripsi yang menggunakan sepasang kunci, kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Anda dapat berbagi kunci publik karena tidak digunakan untuk dekripsi, tetapi akses ke kunci pribadi harus sangat dibatasi.

atomisitas, konsistensi, isolasi, daya tahan (ACID)

Satu set properti perangkat lunak yang menjamin validitas data dan keandalan operasional database, bahkan dalam kasus kesalahan, kegagalan daya, atau masalah lainnya.

kontrol akses berbasis atribut (ABAC)

Praktik membuat izin berbutir halus berdasarkan atribut pengguna, seperti departemen, peran pekerjaan, dan nama tim. Untuk informasi selengkapnya, lihat ABAC untuk AWS dokumentasi AWS Identity and Access Management (IAM).

sumber data otoritatif

Lokasi di mana Anda menyimpan versi utama data, yang dianggap sebagai sumber informasi yang paling dapat diandalkan. Anda dapat menyalin data dari sumber data otoritatif ke lokasi lain untuk tujuan memproses atau memodifikasi data, seperti menganonimkan, menyunting, atau membuat nama samaran.

Zona Ketersediaan

Lokasi berbeda di dalam Wilayah AWS yang terisolasi dari kegagalan di Availability Zone lainnya dan menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama.

AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat situs web AWS CAF dan whitepaper AWS CAF.

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

Alat yang mengevaluasi beban kerja migrasi database, merekomendasikan strategi migrasi, dan memberikan perkiraan kerja. AWS WQF disertakan dengan AWS Schema Conversion Tool ()AWS SCT. Ini menganalisis skema database dan objek kode, kode aplikasi, dependensi, dan karakteristik kinerja, dan memberikan laporan penilaian.

B

bot buruk

Bot yang dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

Pandangan interaktif yang terpadu tentang perilaku dan interaksi sumber daya dari waktu ke waktu. Anda dapat menggunakan grafik perilaku dengan Amazon Detective untuk memeriksa upaya logon yang gagal, panggilan API yang mencurigakan, dan tindakan serupa. Untuk informasi selengkapnya, lihat Data dalam grafik perilaku di dokumentasi Detektif.

sistem big-endian

Sistem yang menyimpan byte paling signifikan terlebih dahulu. Lihat juga endianness.

klasifikasi biner

Sebuah proses yang memprediksi hasil biner (salah satu dari dua kelas yang mungkin). Misalnya, model ML Anda mungkin perlu memprediksi masalah seperti “Apakah email ini spam atau bukan spam?” atau “Apakah produk ini buku atau mobil?”

filter mekar

Struktur data probabilistik dan efisien memori yang digunakan untuk menguji apakah suatu elemen adalah anggota dari suatu himpunan.

deployment biru/hijau

Strategi penyebaran tempat Anda membuat dua lingkungan yang terpisah namun identik. Anda menjalankan versi aplikasi saat ini di satu lingkungan (biru) dan versi aplikasi baru di lingkungan lain (hijau). Strategi ini membantu Anda dengan cepat memutar kembali dengan dampak minimal.

bot

Aplikasi perangkat lunak yang menjalankan tugas otomatis melalui internet dan mensimulasikan aktivitas atau interaksi manusia. Beberapa bot berguna atau bermanfaat, seperti perayap web yang mengindeks informasi di internet. Beberapa bot lain, yang dikenal sebagai bot buruk, dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

botnet

Jaringan bot yang terinfeksi oleh malware dan berada di bawah kendali satu pihak, yang dikenal sebagai bot herder atau operator bot. Botnet adalah mekanisme paling terkenal untuk skala bot dan dampaknya.

cabang

Area berisi repositori kode. Cabang pertama yang dibuat dalam repositori adalah cabang utama. Anda dapat membuat cabang baru dari cabang yang ada, dan Anda kemudian dapat mengembangkan fitur atau memperbaiki bug di cabang baru. Cabang yang Anda buat untuk membangun fitur biasanya disebut sebagai cabang fitur. Saat fitur siap dirilis, Anda menggabungkan cabang fitur kembali ke cabang utama. Untuk informasi selengkapnya, lihat Tentang cabang (GitHub dokumentasi).

akses break-glass

Dalam keadaan luar biasa dan melalui proses yang disetujui, cara cepat bagi pengguna untuk mendapatkan akses ke Akun AWS yang biasanya tidak memiliki izin untuk mengaksesnya. Untuk informasi lebih lanjut, lihat indikator Implementasikan prosedur break-glass dalam panduan Well-Architected AWS .

strategi brownfield

Infrastruktur yang ada di lingkungan Anda. Saat mengadopsi strategi brownfield untuk arsitektur sistem, Anda merancang arsitektur di sekitar kendala sistem dan infrastruktur saat ini. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

Apa yang dilakukan bisnis untuk menghasilkan nilai (misalnya, penjualan, layanan pelanggan, atau pemasaran). Arsitektur layanan mikro dan keputusan pengembangan dapat didorong oleh kemampuan bisnis. Untuk informasi selengkapnya, lihat bagian Terorganisir di sekitar kemampuan bisnis dari Menjalankan layanan mikro kontainer di whitepaper. AWS

perencanaan kelangsungan bisnis (BCP)

Rencana yang membahas dampak potensial dari peristiwa yang mengganggu, seperti migrasi skala besar, pada operasi dan memungkinkan bisnis untuk melanjutkan operasi dengan cepat.

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

Rilis versi yang lambat dan bertahap untuk pengguna akhir. Ketika Anda yakin, Anda menyebarkan versi baru dan mengganti versi saat ini secara keseluruhan.

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

Proses melacak perubahan ke sumber data, seperti tabel database, dan merekam metadata tentang perubahan tersebut. Anda dapat menggunakan CDC untuk berbagai tujuan, seperti mengaudit atau mereplikasi perubahan dalam sistem target untuk mempertahankan sinkronisasi.

rekayasa kekacauan

Dengan sengaja memperkenalkan kegagalan atau peristiwa yang mengganggu untuk menguji ketahanan sistem. Anda dapat menggunakan AWS Fault Injection Service (AWS FIS) untuk melakukan eksperimen yang menekankan AWS beban kerja Anda dan mengevaluasi responsnya.

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

Proses kategorisasi yang membantu menghasilkan prediksi. Model ML untuk masalah klasifikasi memprediksi nilai diskrit. Nilai diskrit selalu berbeda satu sama lain. Misalnya, model mungkin perlu mengevaluasi apakah ada mobil dalam gambar atau tidak.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

Tim multi-disiplin yang mendorong upaya adopsi cloud di seluruh organisasi, termasuk mengembangkan praktik terbaik cloud, memobilisasi sumber daya, menetapkan jadwal migrasi, dan memimpin organisasi melalui transformasi skala besar. Untuk informasi selengkapnya, lihat posting CCo E di Blog Strategi AWS Cloud Perusahaan.

komputasi cloud

Teknologi cloud yang biasanya digunakan untuk penyimpanan data jarak jauh dan manajemen perangkat IoT. Cloud computing umumnya terhubung ke teknologi edge computing.

model operasi cloud

Dalam organisasi TI, model operasi yang digunakan untuk membangun, mematangkan, dan mengoptimalkan satu atau lebih lingkungan cloud. Untuk informasi selengkapnya, lihat Membangun Model Operasi Cloud Anda.

tahap adopsi cloud

Empat fase yang biasanya dilalui organisasi ketika mereka bermigrasi ke AWS Cloud:

- Proyek — Menjalankan beberapa proyek terkait cloud untuk bukti konsep dan tujuan pembelajaran
- Foundation — Melakukan investasi dasar untuk meningkatkan adopsi cloud Anda (misalnya, membuat landing zone, mendefinisikan CCo E, membuat model operasi)
- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog The Journey Toward Cloud-First & the Stages of Adoption di blog Strategi Perusahaan. AWS Cloud Untuk informasi tentang bagaimana kaitannya dengan strategi AWS migrasi, lihat panduan kesiapan migrasi.

CMDB

Lihat database manajemen konfigurasi.

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau. Bitbucket Cloud Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori

dikhususkan untuk satu bagian fungsionalitas. Pipa CI/CD tunggal dapat menggunakan beberapa repositori.

cache dingin

Cache buffer yang kosong, tidak terisi dengan baik, atau berisi data basi atau tidak relevan. Ini mempengaruhi kinerja karena instance database harus membaca dari memori utama atau disk, yang lebih lambat daripada membaca dari cache buffer.

data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat atau kelas penyimpanan yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer (CV)

Bidang AI yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, Amazon SageMaker AI menyediakan algoritma pemrosesan gambar untuk CV.

konfigurasi drift

Untuk beban kerja, konfigurasi berubah dari status yang diharapkan. Ini dapat menyebabkan beban kerja menjadi tidak patuh, dan biasanya bertahap dan tidak disengaja.

database manajemen konfigurasi (CMDB)

Repositori yang menyimpan dan mengelola informasi tentang database dan lingkungan TI, termasuk komponen perangkat keras dan perangkat lunak dan konfigurasinya. Anda biasanya menggunakan data dari CMDB dalam penemuan portofolio dan tahap analisis migrasi.

paket kesesuaian

Kumpulan AWS Config aturan dan tindakan remediasi yang dapat Anda kumpulkan untuk menyesuaikan kepatuhan dan pemeriksaan keamanan Anda. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal di Akun AWS dan Region, atau di seluruh organisasi, dengan menggunakan templat YAMM. Untuk informasi selengkapnya, lihat Paket kesesuaian dalam dokumentasi. AWS Config

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD is commonly described as a pipeline. CI/CDdapat membantu

Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat Manfaat pengiriman berkelanjutan. CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat Continuous Delivery vs Continuous Deployment.

CV

Lihat visi komputer.

D

data saat istirahat

Data yang stasioner di jaringan Anda, seperti data yang ada di penyimpanan.

klasifikasi data

Proses untuk mengidentifikasi dan mengkategorikan data dalam jaringan Anda berdasarkan kekritisan dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat Klasifikasi data.

penyimpangan data

Variasi yang berarti antara data produksi dan data yang digunakan untuk melatih model ML, atau perubahan yang berarti dalam data input dari waktu ke waktu. Penyimpangan data dapat mengurangi kualitas, akurasi, dan keadilan keseluruhan dalam prediksi model ML.

data dalam transit

Data yang aktif bergerak melalui jaringan Anda, seperti antara sumber daya jaringan.

jala data

Kerangka arsitektur yang menyediakan kepemilikan data terdistribusi dan terdesentralisasi dengan manajemen dan tata kelola terpusat.

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data di dalamnya AWS Cloud dapat mengurangi risiko privasi, biaya, dan jejak karbon analitik Anda.

perimeter data

Satu set pagar pembatas pencegahan di AWS lingkungan Anda yang membantu memastikan bahwa hanya identitas tepercaya yang mengakses sumber daya tepercaya dari jaringan yang diharapkan. Untuk informasi selengkapnya, lihat Membangun perimeter data pada AWS.

prapemrosesan data

Untuk mengubah data mentah menjadi format yang mudah diuraikan oleh model ML Anda. Preprocessing data dapat berarti menghapus kolom atau baris tertentu dan menangani nilai yang hilang, tidak konsisten, atau duplikat.

asal data

Proses melacak asal dan riwayat data sepanjang siklus hidupnya, seperti bagaimana data dihasilkan, ditransmisikan, dan disimpan.

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

Sistem manajemen data yang mendukung intelijen bisnis, seperti analitik. Gudang data biasanya berisi sejumlah besar data historis, dan biasanya digunakan untuk kueri dan analisis.

bahasa definisi database (DDL)

Pernyataan atau perintah untuk membuat atau memodifikasi struktur tabel dan objek dalam database.

bahasa manipulasi basis data (DHTML)

Pernyataan atau perintah untuk memodifikasi (memasukkan, memperbarui, dan menghapus) informasi dalam database.

DDL

Lihat bahasa definisi database.

ansambel yang dalam

Untuk menggabungkan beberapa model pembelajaran mendalam untuk prediksi. Anda dapat menggunakan ansambel dalam untuk mendapatkan prediksi yang lebih akurat atau untuk memperkirakan ketidakpastian dalam prediksi.

pembelajaran mendalam

Subbidang ML yang menggunakan beberapa lapisan jaringan saraf tiruan untuk mengidentifikasi pemetaan antara data input dan variabel target yang diinginkan.

defense-in-depth

Pendekatan keamanan informasi di mana serangkaian mekanisme dan kontrol keamanan dilapisi dengan cermat di seluruh jaringan komputer untuk melindungi kerahasiaan, integritas, dan ketersediaan jaringan dan data di dalamnya. Saat Anda mengadopsi strategi ini AWS, Anda menambahkan beberapa kontrol pada lapisan AWS Organizations struktur yang berbeda untuk membantu mengamankan sumber daya. Misalnya, defense-in-depth pendekatan mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

administrator yang didelegasikan

Di AWS Organizations, layanan yang kompatibel dapat mendaftarkan akun AWS anggota untuk mengelola akun organisasi dan mengelola izin untuk layanan tersebut. Akun ini disebut administrator yang didelegasikan untuk layanan itu. Untuk informasi selengkapnya dan daftar layanan yang kompatibel, lihat Layanan yang berfungsi dengan AWS Organizations AWS Organizations dokumentasi.

deployment

Proses pembuatan aplikasi, fitur baru, atau perbaikan kode tersedia di lingkungan target. Deployment melibatkan penerapan perubahan dalam basis kode dan kemudian membangun dan menjalankan basis kode itu di lingkungan aplikasi.

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

Kontrol keamanan yang dirancang untuk mendeteksi, mencatat, dan memperingatkan setelah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan kedua, memperingatkan Anda tentang peristiwa keamanan yang melewati kontrol pencegahan yang ada. Untuk informasi selengkapnya, lihat Kontrol Detektif dalam Menerapkan kontrol keamanan pada. AWS

pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik

manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

kembar digital

Representasi virtual dari sistem dunia nyata, seperti bangunan, pabrik, peralatan industri, atau jalur produksi. Kembar digital mendukung pemeliharaan prediktif, pemantauan jarak jauh, dan optimalisasi produksi.

tabel dimensi

Dalam skema bintang, tabel yang lebih kecil yang berisi atribut data tentang data kuantitatif dalam tabel fakta. Atribut tabel dimensi biasanya bidang teks atau angka diskrit yang berperilaku seperti teks. Atribut ini biasanya digunakan untuk pembatasan kueri, pemfilteran, dan pelabelan set hasil.

musibah

Peristiwa yang mencegah beban kerja atau sistem memenuhi tujuan bisnisnya di lokasi utama yang digunakan. Peristiwa ini dapat berupa bencana alam, kegagalan teknis, atau akibat dari tindakan manusia, seperti kesalahan konfigurasi yang tidak disengaja atau serangan malware.

pemulihan bencana (DR)

Strategi dan proses yang Anda gunakan untuk meminimalkan downtime dan kehilangan data yang disebabkan oleh bencana. Untuk informasi selengkapnya, lihat Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML~

Lihat bahasa manipulasi basis data.

desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani oleh setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

DR

Lihat pemulihan bencana.

deteksi drift

Melacak penyimpangan dari konfigurasi dasar. Misalnya, Anda dapat menggunakan AWS CloudFormation untuk mendeteksi penyimpangan dalam sumber daya sistem, atau Anda dapat menggunakannya AWS Control Tower untuk mendeteksi perubahan di landing zone yang mungkin memengaruhi kepatuhan terhadap persyaratan tata kelola.

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

Teknologi yang meningkatkan daya komputasi untuk perangkat pintar di tepi jaringan IoT. Jika dibandingkan dengan komputasi awan, komputasi tepi dapat mengurangi latensi komunikasi dan meningkatkan waktu respons.

pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat Apa itu Pertukaran Data Elektronik.

enkripsi

Proses komputasi yang mengubah data plaintext, yang dapat dibaca manusia, menjadi ciphertext.

kunci enkripsi

String kriptografi dari bit acak yang dihasilkan oleh algoritma enkripsi. Panjang kunci dapat bervariasi, dan setiap kunci dirancang agar tidak dapat diprediksi dan unik.

endianness

Urutan byte disimpan dalam memori komputer. Sistem big-endian menyimpan byte paling signifikan terlebih dahulu. Sistem little-endian menyimpan byte paling tidak signifikan terlebih dahulu.

titik akhir

Lihat titik akhir layanan.

layanan endpoint

Layanan yang dapat Anda host di cloud pribadi virtual (VPC) untuk dibagikan dengan pengguna lain. Anda dapat membuat layanan endpoint dengan AWS PrivateLink dan memberikan izin kepada prinsipal lain Akun AWS atau ke AWS Identity and Access Management (IAM). Akun atau prinsipal ini dapat terhubung ke layanan endpoint Anda secara pribadi dengan membuat titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat Membuat layanan titik akhir di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).

perencanaan sumber daya perusahaan (ERP)

Sistem yang mengotomatiskan dan mengelola proses bisnis utama (seperti akuntansi, MES, dan manajemen proyek) untuk suatu perusahaan.

enkripsi amplop

Proses mengenkripsi kunci enkripsi dengan kunci enkripsi lain. Untuk informasi selengkapnya, lihat Enkripsi amplop dalam dokumentasi AWS Key Management Service (AWS KMS).

lingkungan

Sebuah contoh dari aplikasi yang sedang berjalan. Berikut ini adalah jenis lingkungan yang umum dalam komputasi awan:

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang pengguna akhir dapat mengakses. Dalam pipa CI/CD, lingkungan produksi adalah lingkungan penyebaran terakhir.

- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

epik

Dalam metodologi tangkas, kategori fungsional yang membantu mengatur dan memprioritaskan pekerjaan Anda. Epik memberikan deskripsi tingkat tinggi tentang persyaratan dan tugas implementasi. Misalnya, epos keamanan AWS CAF mencakup manajemen identitas dan akses, kontrol detektif, keamanan infrastruktur, perlindungan data, dan respons insiden. Untuk informasi selengkapnya tentang epos dalam strategi AWS migrasi, lihat panduan implementasi program.

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

Proses menganalisis dataset untuk memahami karakteristik utamanya. Anda mengumpulkan atau mengumpulkan data dan kemudian melakukan penyelidikan awal untuk menemukan pola, mendeteksi anomali, dan memeriksa asumsi. EDA dilakukan dengan menghitung statistik ringkasan dan membuat visualisasi data.

F

tabel fakta

Tabel tengah dalam skema bintang. Ini menyimpan data kuantitatif tentang operasi bisnis. Biasanya, tabel fakta berisi dua jenis kolom: kolom yang berisi ukuran dan yang berisi kunci asing ke tabel dimensi.

gagal cepat

Filosofi yang menggunakan pengujian yang sering dan bertahap untuk mengurangi siklus hidup pengembangan. Ini adalah bagian penting dari pendekatan tangkas.

batas isolasi kesalahan

Dalam AWS Cloud, batas seperti Availability Zone, Wilayah AWS, control plane, atau data plane yang membatasi efek kegagalan dan membantu meningkatkan ketahanan beban kerja. Untuk informasi selengkapnya, lihat Batas Isolasi AWS Kesalahan.

cabang fitur

Lihat cabang.

fitur

Data input yang Anda gunakan untuk membuat prediksi. Misalnya, dalam konteks manufaktur, fitur bisa berupa gambar yang diambil secara berkala dari lini manufaktur.

pentingnya fitur

Seberapa signifikan fitur untuk prediksi model. Ini biasanya dinyatakan sebagai skor numerik yang dapat dihitung melalui berbagai teknik, seperti Shapley Additive Explanations (SHAP) dan gradien terintegrasi. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

transformasi fitur

Untuk mengoptimalkan data untuk proses ML, termasuk memperkaya data dengan sumber tambahan, menskalakan nilai, atau mengekstrak beberapa set informasi dari satu bidang data. Hal ini memungkinkan model ML untuk mendapatkan keuntungan dari data. Misalnya, jika Anda memecah tanggal “2021-05-27 00:15:37" menjadi “2021", “Mei”, “Kamis”, dan “15", Anda dapat membantu algoritme pembelajaran mempelajari pola bernuansa yang terkait dengan komponen data yang berbeda.

beberapa tembakan mendorong

Menyediakan LLM dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Teknik ini adalah aplikasi pembelajaran dalam konteks, di mana model belajar dari contoh (bidikan) yang tertanam dalam petunjuk. Beberapa bidikan dapat efektif untuk tugas-tugas yang memerlukan pemformatan, penalaran, atau pengetahuan domain tertentu. Lihat juga bidikan nol.

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui pengambilan data perubahan untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih

menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

FM

Lihat model pondasi.

model pondasi (FM)

Jaringan saraf pembelajaran mendalam yang besar yang telah melatih kumpulan data besar-besaran data umum dan tidak berlabel. FMs mampu melakukan berbagai tugas umum, seperti memahami bahasa, menghasilkan teks dan gambar, dan berbicara dalam bahasa alami. Untuk informasi selengkapnya, lihat Apa itu Model Foundation.

G

AI generatif

Subset model AI yang telah dilatih pada sejumlah besar data dan yang dapat menggunakan prompt teks sederhana untuk membuat konten dan artefak baru, seperti gambar, video, teks, dan audio. Untuk informasi lebih lanjut, lihat Apa itu AI Generatif.

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

Di Amazon CloudFront, opsi untuk mencegah pengguna di negara tertentu mengakses distribusi konten. Anda dapat menggunakan daftar izinkan atau daftar blokir untuk menentukan negara yang disetujui dan dilarang. Untuk informasi selengkapnya, lihat Membatasi distribusi geografis konten Anda dalam dokumentasi. CloudFront

Alur kerja Gitflow

Pendekatan di mana lingkungan bawah dan atas menggunakan cabang yang berbeda dalam repositori kode sumber. Alur kerja Gitflow dianggap warisan, dan alur kerja berbasis batang adalah pendekatan modern yang lebih disukai.

gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur,

gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

strategi greenfield

Tidak adanya infrastruktur yang ada di lingkungan baru. Saat mengadopsi strategi greenfield untuk arsitektur sistem, Anda dapat memilih semua teknologi baru tanpa batasan kompatibilitas dengan infrastruktur yang ada, juga dikenal sebagai brownfield. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi ()OUs. Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

Memigrasi database sumber Anda ke database target yang menggunakan mesin database yang berbeda (misalnya, Oracle ke Amazon Aurora). Migrasi heterogen biasanya merupakan bagian dari upaya arsitektur ulang, dan mengubah skema dapat menjadi tugas yang kompleks. AWS menyediakan AWS SCT yang membantu dengan konversi skema.

ketersediaan tinggi (HA)

Kemampuan beban kerja untuk beroperasi terus menerus, tanpa intervensi, jika terjadi tantangan atau bencana. Sistem HA dirancang untuk gagal secara otomatis, secara konsisten memberikan kinerja berkualitas tinggi, dan menangani beban dan kegagalan yang berbeda dengan dampak kinerja minimal.

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

data penahanan

Sebagian dari data historis berlabel yang ditahan dari kumpulan data yang digunakan untuk melatih model pembelajaran mesin. Anda dapat menggunakan data penahanan untuk mengevaluasi kinerja model dengan membandingkan prediksi model dengan data penahanan.

migrasi database homogen

Memigrasi database sumber Anda ke database target yang berbagi mesin database yang sama (misalnya, Microsoft SQL Server ke Amazon RDS for SQL Server). Migrasi homogen biasanya merupakan bagian dari upaya rehosting atau replatforming. Anda dapat menggunakan utilitas database asli untuk memigrasi skema.

data panas

Data yang sering diakses, seperti data real-time atau data translasi terbaru. Data ini biasanya memerlukan tingkat atau kelas penyimpanan berkinerja tinggi untuk memberikan respons kueri yang cepat.

perbaikan terbaru

Perbaikan mendesak untuk masalah kritis dalam lingkungan produksi. Karena urgensinya, perbaikan terbaru biasanya dibuat di luar alur kerja DevOps rilis biasa.

periode hypercare

Segera setelah cutover, periode waktu ketika tim migrasi mengelola dan memantau aplikasi yang dimigrasi di cloud untuk mengatasi masalah apa pun. Biasanya, periode ini panjangnya 1-4 hari. Pada akhir periode hypercare, tim migrasi biasanya mentransfer tanggung jawab untuk aplikasi ke tim operasi cloud.

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

Kebijakan yang dilampirkan pada satu atau beberapa prinsip IAM yang mendefinisikan izin mereka dalam lingkungan. AWS Cloud

aplikasi idle

Aplikasi yang memiliki penggunaan CPU dan memori rata-rata antara 5 dan 20 persen selama periode 90 hari. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini atau mempertahankannya di tempat.

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

Model yang menyebarkan infrastruktur baru untuk beban kerja produksi alih-alih memperbarui, menambal, atau memodifikasi infrastruktur yang ada. Infrastruktur yang tidak dapat diubah secara inheren lebih konsisten, andal, dan dapat diprediksi daripada infrastruktur yang dapat berubah. Untuk informasi selengkapnya, lihat praktik terbaik Deploy using immutable infrastructure di AWS Well-Architected Framework.

masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

migrasi inkremental

Strategi cutover di mana Anda memigrasikan aplikasi Anda dalam bagian-bagian kecil alih-alih melakukan satu cutover penuh. Misalnya, Anda mungkin hanya memindahkan beberapa layanan mikro atau pengguna ke sistem baru pada awalnya. Setelah Anda memverifikasi bahwa semuanya berfungsi dengan baik, Anda dapat secara bertahap memindahkan layanan mikro atau pengguna tambahan hingga Anda dapat menonaktifkan sistem lama Anda. Strategi ini mengurangi risiko yang terkait dengan migrasi besar.

Industri 4.0

Sebuah istilah yang diperkenalkan oleh Klaus Schwab pada tahun 2016 untuk merujuk pada modernisasi proses manufaktur melalui kemajuan dalam konektivitas, data real-time, otomatisasi, analitik, dan AI/ML.

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

Proses penyediaan dan pengelolaan infrastruktur aplikasi melalui satu set file konfigurasi. IAc dirancang untuk membantu Anda memusatkan manajemen infrastruktur, menstandarisasi sumber daya, dan menskalakan dengan cepat sehingga lingkungan baru dapat diulang, andal, dan konsisten.

Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi lebih lanjut, lihat Membangun strategi transformasi digital Internet of Things (IIoT) industri.

inspeksi VPC

Dalam arsitektur AWS multi-akun, VPC terpusat yang mengelola inspeksi lalu lintas jaringan antara VPCs (dalam yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

Internet of Things (IoT)

Jaringan objek fisik yang terhubung dengan sensor atau prosesor tertanam yang berkomunikasi dengan perangkat dan sistem lain melalui internet atau melalui jaringan komunikasi lokal. Untuk informasi selengkapnya, lihat Apa itu IoT?

interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

Serangkaian praktik terbaik untuk memberikan layanan TI dan menyelaraskan layanan ini dengan persyaratan bisnis. ITIL menyediakan dasar untuk ITSM.

Manajemen layanan TI (ITSM)

Kegiatan yang terkait dengan merancang, menerapkan, mengelola, dan mendukung layanan TI untuk suatu organisasi. Untuk informasi tentang mengintegrasikan operasi cloud dengan alat ITSM, lihat panduan integrasi operasi.

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

L

kontrol akses berbasis label (LBAC)

Implementasi kontrol akses wajib (MAC) di mana pengguna dan data itu sendiri masing-masing secara eksplisit diberi nilai label keamanan. Persimpangan antara label keamanan pengguna dan label keamanan data menentukan baris dan kolom mana yang dapat dilihat oleh pengguna.

landing zone

Landing zone adalah AWS lingkungan multi-akun yang dirancang dengan baik yang dapat diskalakan dan aman. Ini adalah titik awal dari mana organisasi Anda dapat dengan cepat meluncurkan dan menyebarkan beban kerja dan aplikasi dengan percaya diri dalam lingkungan keamanan dan infrastruktur mereka. Untuk informasi selengkapnya tentang zona pendaratan, lihat Menyiapkan lingkungan multi-akun AWS yang aman dan dapat diskalakan.

model bahasa besar (LLM)

Model AI pembelajaran mendalam yang dilatih sebelumnya pada sejumlah besar data. LLM dapat melakukan beberapa tugas, seperti menjawab pertanyaan, meringkas dokumen, menerjemahkan teks ke dalam bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat Apa itu LLMs.

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

Praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM.

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga endianness.

LLM

Lihat model bahasa besar.

lingkungan yang lebih rendah

Lihat lingkungan.

M

pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat Machine Learning.

cabang utama

Lihat cabang.

malware

Perangkat lunak yang dirancang untuk membahayakan keamanan atau privasi komputer. Malware dapat mengganggu sistem komputer, membocorkan informasi sensitif, atau mendapatkan akses yang tidak sah. Contoh malware termasuk virus, worm, ransomware, Trojan horse, spyware, dan keyloggers.

layanan terkelola

Layanan AWS yang AWS mengoperasikan lapisan infrastruktur, sistem operasi, dan platform, dan Anda mengakses titik akhir untuk menyimpan dan mengambil data. Amazon Simple Storage Service (Amazon S3) dan Amazon DynamoDB adalah contoh layanan terkelola. Ini juga dikenal sebagai layanan abstrak.

sistem eksekusi manufaktur (MES)

Sistem perangkat lunak untuk melacak, memantau, mendokumentasikan, dan mengendalikan proses produksi yang mengubah bahan baku menjadi produk jadi di lantai toko.

PETA

Lihat Program Percepatan Migrasi.

mekanisme

Proses lengkap di mana Anda membuat alat, mendorong adopsi alat, dan kemudian memeriksa hasilnya untuk melakukan penyesuaian. Mekanisme adalah siklus yang memperkuat dan meningkatkan dirinya sendiri saat beroperasi. Untuk informasi lebih lanjut, lihat Membangun mekanisme di AWS Well-Architected Framework.

akun anggota

Semua Akun AWS selain akun manajemen yang merupakan bagian dari organisasi di AWS Organizations. Akun dapat menjadi anggota dari hanya satu organisasi pada suatu waktu.

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

Protokol komunikasi ringan machine-to-machine (M2M), berdasarkan pola terbitkan/berlangganan, untuk perangkat IoT yang dibatasi sumber daya.

layanan mikro

Layanan kecil dan independen yang berkomunikasi dengan jelas APIs dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali, dan ketahanan. Untuk informasi selengkapnya, lihat Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server.

arsitektur microservices

Pendekatan untuk membangun aplikasi dengan komponen independen yang menjalankan setiap proses aplikasi sebagai layanan mikro. Layanan mikro ini berkomunikasi melalui antarmuka yang terdefinisi dengan baik dengan menggunakan ringan. APIs Setiap layanan mikro dalam arsitektur ini dapat diperbarui, digunakan, dan diskalakan untuk memenuhi permintaan fungsi tertentu dari suatu aplikasi. Untuk informasi selengkapnya, lihat Menerapkan layanan mikro di AWS.

Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

migrasi dalam skala

Proses memindahkan sebagian besar portofolio aplikasi ke cloud dalam gelombang, dengan lebih banyak aplikasi bergerak pada tingkat yang lebih cepat di setiap gelombang. Fase ini menggunakan praktik dan pelajaran terbaik dari fase sebelumnya untuk mengimplementasikan pabrik migrasi tim, alat, dan proses untuk merampingkan migrasi beban kerja melalui otomatisasi dan pengiriman tangkas. Ini adalah fase ketiga dari strategi AWS migrasi.

pabrik migrasi

Tim lintas fungsi yang merampingkan migrasi beban kerja melalui pendekatan otomatis dan gesit. Tim pabrik migrasi biasanya mencakup operasi, analis dan pemilik bisnis, insinyur migrasi, pengembang, dan DevOps profesional yang bekerja di sprint. Antara 20 dan 50 persen portofolio aplikasi perusahaan terdiri dari pola berulang yang dapat dioptimalkan dengan pendekatan pabrik. Untuk informasi selengkapnya, lihat diskusi tentang pabrik migrasi dan panduan Pabrik Migrasi Cloud di kumpulan konten ini.

metadata migrasi

Informasi tentang aplikasi dan server yang diperlukan untuk menyelesaikan migrasi. Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda. Contoh metadata migrasi termasuk subnet target, grup keamanan, dan akun. AWS

pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 dengan Layanan Migrasi AWS Aplikasi.

Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke. AWS Cloud MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). Alat MPA (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

Penilaian Kesiapan Migrasi (MRA)

Proses mendapatkan wawasan tentang status kesiapan cloud organisasi, mengidentifikasi kekuatan dan kelemahan, dan membangun rencana aksi untuk menutup kesenjangan yang diidentifikasi, menggunakan CAF. AWS Untuk informasi selengkapnya, lihat panduan kesiapan migrasi. MRA adalah tahap pertama dari strategi AWS migrasi.

strategi migrasi

Pendekatan yang digunakan untuk memigrasikan beban kerja ke file. AWS Cloud Untuk informasi lebih lanjut, lihat entri 7 Rs di glosarium ini dan lihat Memobilisasi organisasi Anda untuk mempercepat migrasi skala besar.

ML

Lihat pembelajaran mesin.

modernisasi

Mengubah aplikasi usang (warisan atau monolitik) dan infrastrukturnya menjadi sistem yang gesit, elastis, dan sangat tersedia di cloud untuk mengurangi biaya, mendapatkan efisiensi, dan memanfaatkan inovasi. Untuk informasi selengkapnya, lihat Strategi untuk memodernisasi aplikasi di. AWS Cloud

penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat Mengevaluasi kesiapan modernisasi untuk aplikasi di. AWS Cloud

aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat Menguraikan monolit menjadi layanan mikro.

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

Sebuah proses yang membantu menghasilkan prediksi untuk beberapa kelas (memprediksi satu dari lebih dari dua hasil). Misalnya, model ML mungkin bertanya “Apakah produk ini buku, mobil, atau telepon?” atau “Kategori produk mana yang paling menarik bagi pelanggan ini?”

infrastruktur yang bisa berubah

Model yang memperbarui dan memodifikasi infrastruktur yang ada untuk beban kerja produksi. Untuk meningkatkan konsistensi, keandalan, dan prediktabilitas, AWS Well-Architected Framework merekomendasikan penggunaan infrastruktur yang tidak dapat diubah sebagai praktik terbaik.

O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

Metode migrasi di mana beban kerja sumber diturunkan selama proses migrasi. Metode ini melibatkan waktu henti yang diperpanjang dan biasanya digunakan untuk beban kerja kecil dan tidak kritis.

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

Metode migrasi di mana beban kerja sumber disalin ke sistem target tanpa diambil offline. Aplikasi yang terhubung ke beban kerja dapat terus berfungsi selama migrasi. Metode ini melibatkan waktu henti nol hingga minimal dan biasanya digunakan untuk beban kerja produksi yang kritis.

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

Protokol komunikasi machine-to-machine (M2M) untuk otomasi industri. OPC-UA menyediakan standar interoperabilitas dengan enkripsi data, otentikasi, dan skema otorisasi.

perjanjian tingkat operasional (OLA)

Perjanjian yang menjelaskan apa yang dijanjikan kelompok TI fungsional untuk diberikan satu sama lain, untuk mendukung perjanjian tingkat layanan (SLA).

Tinjauan Kesiapan Operasional (ORR)

Daftar pertanyaan dan praktik terbaik terkait yang membantu Anda memahami, mengevaluasi, mencegah, atau mengurangi ruang lingkup insiden dan kemungkinan kegagalan. Untuk informasi lebih lanjut, lihat Ulasan Kesiapan Operasional (ORR) dalam Kerangka Kerja Well-Architected AWS .

teknologi operasional (OT)

Sistem perangkat keras dan perangkat lunak yang bekerja dengan lingkungan fisik untuk mengendalikan operasi industri, peralatan, dan infrastruktur. Di bidang manufaktur, integrasi sistem OT dan teknologi informasi (TI) adalah fokus utama untuk transformasi Industri 4.0.

integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat panduan integrasi operasi.

jejak organisasi

Jejak yang dibuat oleh AWS CloudTrail itu mencatat semua peristiwa untuk semua Akun AWS dalam organisasi di AWS Organizations. Jejak ini dibuat di setiap Akun AWS bagian organisasi dan melacak aktivitas di setiap akun. Untuk informasi selengkapnya, lihat Membuat jejak untuk organisasi dalam CloudTrail dokumentasi.

manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi, dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat panduan OCM.

kontrol akses asal (OAC)

Di CloudFront, opsi yang disempurnakan untuk membatasi akses untuk mengamankan konten Amazon Simple Storage Service (Amazon S3) Anda. OAC mendukung semua bucket S3 di semua Wilayah AWS, enkripsi sisi server dengan AWS KMS (SSE-KMS), dan dinamis dan permintaan ke bucket S3. `PUT` `DELETE`

identitas akses asal (OAI)

Di CloudFront, opsi untuk membatasi akses untuk mengamankan konten Amazon S3 Anda. Saat Anda menggunakan OAI, CloudFront buat prinsipal yang dapat diautentikasi oleh Amazon S3. Prinsipal yang diautentikasi dapat mengakses konten dalam bucket S3 hanya melalui distribusi tertentu. CloudFront Lihat juga OAC, yang menyediakan kontrol akses yang lebih terperinci dan ditingkatkan.

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

P

batas izin

Kebijakan manajemen IAM yang dilampirkan pada prinsipal IAM untuk menetapkan izin maksimum yang dapat dimiliki pengguna atau peran. Untuk informasi selengkapnya, lihat Batas izin dalam dokumentasi IAM.

Informasi Identifikasi Pribadi (PII)

Informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar. Contoh PII termasuk nama, alamat, dan informasi kontak.

PII

Lihat informasi yang dapat diidentifikasi secara pribadi.

buku pedoman

Serangkaian langkah yang telah ditentukan sebelumnya yang menangkap pekerjaan yang terkait dengan migrasi, seperti mengirimkan fungsi operasi inti di cloud. Buku pedoman dapat berupa skrip, runbook otomatis, atau ringkasan proses atau langkah-langkah yang diperlukan untuk mengoperasikan lingkungan modern Anda.

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

Objek yang dapat menentukan izin (lihat kebijakan berbasis identitas), menentukan kondisi akses (lihat kebijakan berbasis sumber daya), atau menentukan izin maksimum untuk semua akun di organisasi (lihat kebijakan kontrol layanan). AWS Organizations

ketekunan poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka. Untuk informasi selengkapnya, lihat Mengaktifkan persistensi data di layanan mikro.

penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat Mengevaluasi kesiapan migrasi.

predikat

Kondisi kueri yang mengembalikan `true` atau`false`, biasanya terletak di `WHERE` klausa.

predikat pushdown

Teknik optimasi kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

kontrol preventif

Kontrol keamanan yang dirancang untuk mencegah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan pertama untuk membantu mencegah akses tidak sah atau perubahan yang tidak diinginkan ke jaringan Anda. Untuk informasi selengkapnya, lihat Kontrol pencegahan dalam Menerapkan kontrol keamanan pada. AWS

principal

Entitas AWS yang dapat melakukan tindakan dan mengakses sumber daya. Entitas ini biasanya merupakan pengguna root untuk Akun AWS, peran IAM, atau pengguna. Untuk informasi selengkapnya, lihat Prinsip dalam istilah dan konsep Peran dalam dokumentasi IAM.

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona yang dihosting pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau lebih. VPCs Untuk informasi selengkapnya, lihat Bekerja dengan zona yang dihosting pribadi di dokumentasi Route 53.

kontrol proaktif

Kontrol keamanan yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat panduan referensi Kontrol dalam AWS Control Tower dokumentasi dan lihat Kontrol proaktif dalam Menerapkan kontrol keamanan pada AWS.

manajemen siklus hidup produk (PLM)

Manajemen data dan proses untuk suatu produk di seluruh siklus hidupnya, mulai dari desain, pengembangan, dan peluncuran, melalui pertumbuhan dan kematangan, hingga penurunan dan penghapusan.

lingkungan produksi

Lihat lingkungan.

pengontrol logika yang dapat diprogram (PLC)

Di bidang manufaktur, komputer yang sangat andal dan mudah beradaptasi yang memantau mesin dan mengotomatiskan proses manufaktur.

rantai cepat

Menggunakan output dari satu prompt LLM sebagai input untuk prompt berikutnya untuk menghasilkan respons yang lebih baik. Teknik ini digunakan untuk memecah tugas yang kompleks menjadi subtugas, atau untuk secara iteratif memperbaiki atau memperluas respons awal. Ini membantu meningkatkan akurasi dan relevansi respons model dan memungkinkan hasil yang lebih terperinci dan dipersonalisasi.

pseudonimisasi

Proses penggantian pengenal pribadi dalam kumpulan data dengan nilai placeholder. Pseudonimisasi dapat membantu melindungi privasi pribadi. Data pseudonim masih dianggap sebagai data pribadi.

publish/subscribe (pub/sub)

Pola yang memungkinkan komunikasi asinkron antara layanan mikro untuk meningkatkan skalabilitas dan daya tanggap. Misalnya, dalam MES berbasis layanan mikro, layanan mikro dapat mempublikasikan pesan peristiwa ke saluran yang dapat berlangganan layanan mikro lainnya. Sistem dapat menambahkan layanan mikro baru tanpa mengubah layanan penerbitan.

Q

rencana kueri

Serangkaian langkah, seperti instruksi, yang digunakan untuk mengakses data dalam sistem database relasional SQL.

regresi rencana kueri

Ketika pengoptimal layanan database memilih rencana yang kurang optimal daripada sebelum perubahan yang diberikan ke lingkungan database. Hal ini dapat disebabkan oleh perubahan statistik, kendala, pengaturan lingkungan, pengikatan parameter kueri, dan pembaruan ke mesin database.

R

Matriks RACI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

LAP

Lihat Retrieval Augmented Generation.

ransomware

Perangkat lunak berbahaya yang dirancang untuk memblokir akses ke sistem komputer atau data sampai pembayaran dilakukan.

Matriks RASCI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

Salinan database yang digunakan untuk tujuan read-only. Anda dapat merutekan kueri ke replika baca untuk mengurangi beban pada database utama Anda.

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

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.

tujuan waktu pemulihan (RTO)

Penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan.

refactor

Lihat 7 Rs.

Wilayah

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat Menentukan Wilayah AWS akun yang dapat digunakan.

regresi

Teknik ML yang memprediksi nilai numerik. Misalnya, untuk memecahkan masalah “Berapa harga rumah ini akan dijual?” Model ML dapat menggunakan model regresi linier untuk memprediksi harga jual rumah berdasarkan fakta yang diketahui tentang rumah (misalnya, luas persegi).

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

Kemampuan aplikasi untuk melawan atau pulih dari gangguan. Ketersediaan tinggi dan pemulihan bencana adalah pertimbangan umum ketika merencanakan ketahanan di. AWS Cloud Untuk informasi lebih lanjut, lihat AWS Cloud Ketahanan.

kebijakan berbasis sumber daya

Kebijakan yang dilampirkan ke sumber daya, seperti bucket Amazon S3, titik akhir, atau kunci enkripsi. Jenis kebijakan ini menentukan prinsipal mana yang diizinkan mengakses, tindakan yang didukung, dan kondisi lain yang harus dipenuhi.

matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI)

Matriks yang mendefinisikan 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), dikonsultasikan (C), dan diinformasikan (I). Tipe dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

kontrol responsif

Kontrol keamanan yang dirancang untuk mendorong remediasi efek samping atau penyimpangan dari garis dasar keamanan Anda. Untuk informasi selengkapnya, lihat Kontrol responsif dalam Menerapkan kontrol keamanan pada AWS.

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

Retrieval Augmented Generation (RAG)

Teknologi AI generatif di mana LLM merujuk sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin melakukan pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat Apa itu RAG.

rotasi

Proses memperbarui rahasia secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensil.

kontrol akses baris dan kolom (RCAC)

Penggunaan ekspresi SQL dasar dan fleksibel yang telah menetapkan aturan akses. RCAC terdiri dari izin baris dan topeng kolom.

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

Satu set prosedur manual atau otomatis yang diperlukan untuk melakukan tugas tertentu. Ini biasanya dibangun untuk merampingkan operasi berulang atau prosedur dengan tingkat kesalahan yang tinggi.

D

SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke AWS Management Console atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat Tentang federasi berbasis SAMP 2.0 dalam dokumentasi IAM.

SCADA

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensil pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadata-nya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat Apa yang ada di rahasia Secrets Manager? dalam dokumentasi Secrets Manager.

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

Pagar pembatas teknis atau administratif yang mencegah, mendeteksi, atau mengurangi kemampuan pelaku ancaman untuk mengeksploitasi kerentanan keamanan. Ada empat jenis kontrol keamanan utama: preventif, detektif, responsif, dan proaktif.

pengerasan keamanan

Proses mengurangi permukaan serangan untuk membuatnya lebih tahan terhadap serangan. Ini dapat mencakup tindakan seperti menghapus sumber daya yang tidak lagi diperlukan, menerapkan praktik keamanan terbaik untuk memberikan hak istimewa paling sedikit, atau menonaktifkan fitur yang tidak perlu dalam file konfigurasi.

sistem informasi keamanan dan manajemen acara (SIEM)

Alat dan layanan yang menggabungkan sistem manajemen informasi keamanan (SIM) dan manajemen acara keamanan (SEM). Sistem SIEM mengumpulkan, memantau, dan menganalisis data dari server, jaringan, perangkat, dan sumber lain untuk mendeteksi ancaman dan pelanggaran keamanan, dan untuk menghasilkan peringatan.

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan detektif atau responsif yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans EC2 Amazon, atau memutar kredensil.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCPs menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCPs daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam AWS Organizations dokumentasi.

titik akhir layanan

URL titik masuk untuk file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat Layanan AWS titik akhir di Referensi Umum AWS.

perjanjian tingkat layanan (SLA)

Perjanjian yang menjelaskan apa yang dijanjikan tim TI untuk diberikan kepada pelanggan mereka, seperti uptime dan kinerja layanan.

indikator tingkat layanan (SLI)

Pengukuran aspek kinerja layanan, seperti tingkat kesalahan, ketersediaan, atau throughputnya.

tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator tingkat layanan.

model tanggung jawab bersama

Model yang menjelaskan tanggung jawab yang Anda bagikan AWS untuk keamanan dan kepatuhan cloud. AWS bertanggung jawab atas keamanan cloud, sedangkan Anda bertanggung jawab atas keamanan di cloud. Untuk informasi selengkapnya, lihat Model tanggung jawab bersama.

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

Kegagalan dalam satu komponen penting dari aplikasi yang dapat mengganggu sistem.

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

Pola untuk menskalakan dan mempercepat proyek modernisasi. Ketika fitur baru dan rilis produk didefinisikan, tim inti berpisah untuk membuat tim produk baru. Ini membantu meningkatkan kemampuan dan layanan organisasi Anda, meningkatkan produktivitas pengembang, dan

mendukung inovasi yang cepat. Untuk informasi lebih lanjut, lihat Pendekatan bertahap untuk memodernisasi aplikasi di. AWS Cloud

SPOF

Lihat satu titik kegagalan.

skema bintang

Struktur organisasi database yang menggunakan satu tabel fakta besar untuk menyimpan data transaksional atau terukur dan menggunakan satu atau lebih tabel dimensi yang lebih kecil untuk menyimpan atribut data. Struktur ini dirancang untuk digunakan dalam gudang data atau untuk tujuan intelijen bisnis.

pola ara pencekik

Pendekatan untuk memodernisasi sistem monolitik dengan menulis ulang secara bertahap dan mengganti fungsionalitas sistem sampai sistem warisan dapat dinonaktifkan. Pola ini menggunakan analogi pohon ara yang tumbuh menjadi pohon yang sudah mapan dan akhirnya mengatasi dan menggantikan inangnya. Pola ini diperkenalkan oleh Martin Fowler sebagai cara untuk mengelola risiko saat menulis ulang sistem monolitik. Untuk contoh cara menerapkan pola ini, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

subnet

Rentang alamat IP dalam VPC Anda. Subnet harus berada di Availability Zone tunggal.

kontrol pengawasan dan akuisisi data (SCADA)

Di bidang manufaktur, sistem yang menggunakan perangkat keras dan perangkat lunak untuk memantau aset fisik dan operasi produksi.

enkripsi simetris

Algoritma enkripsi yang menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi data.

pengujian sintetis

Menguji sistem dengan cara yang mensimulasikan interaksi pengguna untuk mendeteksi potensi masalah atau untuk memantau kinerja. Anda dapat menggunakan Amazon CloudWatch Synthetics untuk membuat tes ini.

sistem prompt

Teknik untuk memberikan konteks, instruksi, atau pedoman ke LLM untuk mengarahkan perilakunya. Permintaan sistem membantu mengatur konteks dan menetapkan aturan untuk interaksi dengan pengguna.

T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda dapat membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai AWS sumber daya Anda.

variabel target

Nilai yang Anda coba prediksi dalam ML yang diawasi. Ini juga disebut sebagai variabel hasil. Misalnya, dalam pengaturan manufaktur, variabel target bisa menjadi cacat produk.

daftar tugas

Alat yang digunakan untuk melacak kemajuan melalui runbook. Daftar tugas berisi ikhtisar runbook dan daftar tugas umum yang harus diselesaikan. Untuk setiap tugas umum, itu termasuk perkiraan jumlah waktu yang dibutuhkan, pemilik, dan kemajuan.

lingkungan uji

Lihat lingkungan.

pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. Untuk informasi selengkapnya, lihat Apa itu gateway transit dalam AWS Transit Gateway dokumentasi.

alur kerja berbasis batang

Pendekatan di mana pengembang membangun dan menguji fitur secara lokal di cabang fitur dan kemudian menggabungkan perubahan tersebut ke cabang utama. Cabang utama kemudian dibangun untuk pengembangan, praproduksi, dan lingkungan produksi, secara berurutan.

akses tepercaya

Memberikan izin ke layanan yang Anda tentukan untuk melakukan tugas di organisasi Anda di dalam AWS Organizations dan di akunnya atas nama Anda. Layanan tepercaya menciptakan peran terkait layanan di setiap akun, ketika peran itu diperlukan, untuk melakukan tugas manajemen untuk Anda. Untuk informasi selengkapnya, lihat Menggunakan AWS Organizations dengan AWS layanan lain dalam AWS Organizations dokumentasi.

penyetelan

Untuk mengubah aspek proses pelatihan Anda untuk meningkatkan akurasi model ML. Misalnya, Anda dapat melatih model ML dengan membuat set pelabelan, menambahkan label, dan kemudian mengulangi langkah-langkah ini beberapa kali di bawah pengaturan yang berbeda untuk mengoptimalkan model.

tim dua pizza

Sebuah DevOps tim kecil yang bisa Anda beri makan dengan dua pizza. Ukuran tim dua pizza memastikan peluang terbaik untuk berkolaborasi dalam pengembangan perangkat lunak.

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian: ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data. Untuk informasi lebih lanjut, lihat panduan Mengukur ketidakpastian dalam sistem pembelajaran mendalam.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau

memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat lingkungan.

V

menyedot debu

Operasi pemeliharaan database yang melibatkan pembersihan setelah pembaruan tambahan untuk merebut kembali penyimpanan dan meningkatkan kinerja.

kendali versi

Proses dan alat yang melacak perubahan, seperti perubahan kode sumber dalam repositori.

Peering VPC

Koneksi antara dua VPCs yang memungkinkan Anda untuk merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat Apa itu peering VPC di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

Cache buffer yang berisi data saat ini dan relevan yang sering diakses. Instance database dapat membaca dari cache buffer, yang lebih cepat daripada membaca dari memori utama atau disk.

data hangat

Data yang jarang diakses. Saat menanyakan jenis data ini, kueri yang cukup lambat biasanya dapat diterima.

fungsi jendela

Fungsi SQL yang melakukan perhitungan pada sekelompok baris yang berhubungan dengan catatan saat ini. Fungsi jendela berguna untuk memproses tugas, seperti menghitung rata-rata bergerak atau mengakses nilai baris berdasarkan posisi relatif dari baris saat ini.

beban kerja

Kumpulan sumber daya dan kode yang memberikan nilai bisnis, seperti aplikasi yang dihadapi pelanggan atau proses backend.

aliran kerja

Grup fungsional dalam proyek migrasi yang bertanggung jawab atas serangkaian tugas tertentu. Setiap alur kerja independen tetapi mendukung alur kerja lain dalam proyek. Misalnya, alur kerja portofolio bertanggung jawab untuk memprioritaskan aplikasi, perencanaan gelombang, dan mengumpulkan metadata migrasi. Alur kerja portofolio mengirimkan aset ini ke alur kerja migrasi, yang kemudian memigrasikan server dan aplikasi.

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

Model penyimpanan yang menulis data satu kali dan mencegah data dihapus atau dimodifikasi. Pengguna yang berwenang dapat membaca data sebanyak yang diperlukan, tetapi mereka tidak dapat mengubahnya. Infrastruktur penyimpanan data ini dianggap tidak dapat diubah.

Z

eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan zero-day.

kerentanan zero-day

Cacat atau kerentanan yang tak tanggung-tanggung dalam sistem produksi. Aktor ancaman dapat menggunakan jenis kerentanan ini untuk menyerang sistem. Pengembang sering menyadari kerentanan sebagai akibat dari serangan tersebut.

bisikan zero-shot

Memberikan LLM dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembakan) yang dapat membantu membimbingnya. LLM harus menggunakan pengetahuan pra-terlatih untuk menangani tugas. Efektivitas bidikan nol tergantung pada kompleksitas tugas dan kualitas prompt. Lihat juga beberapa bidikan yang diminta.

aplikasi zombie

Aplikasi yang memiliki CPU rata-rata dan penggunaan memori di bawah 5 persen. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.